Title
The design of Bayes consistent loss functions for classification

Permalink
https://escholarship.org/uc/item/1cv1947c

Author
Masnadi-Shirazi, Hamed

Publication Date
2011

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

The design of Bayes consistent loss functions for classification

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering (Signal and Image Processing)

by

Hamed Masnadi-Shirazi

Committee in charge:

Professor Nuno Vasconcelos, Chair
Professor Serge Belongie
Professor Charles Elkan
Professor Kenneth Kreutz-Delgado
Professor Bhaskar D. Rao
Professor Lawrence Saul

2011

The dissertation of Hamed Masnadi-Shirazi is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2011

DEDICATION

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

In the Name of Allāh, the Most Gracious, the Most Merciful

I dedicate this thesis to my mother, Mehri Daneshvar, and my father, Dr. Mohammad Ali Masnadi-Shirazi, as I owe them everything after God.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

It has been said that if you want to thank God you should thank his subjects. And so, I would like to acknowledge many people for helping me during my education and doctoral work. Their help and support was instrumental in the completion of this work.

First of all, I would like to express my deepest gratitude to my supervisor, Professor Nuno Vasconcelos. Apart from providing his exceptional scientific guidance and insight, which is without a doubt, reflected in this thesis, I should also thank him for giving me a chance to explore new ideas while being patient towards my mistakes and also demonstrating to me what it means to be a true scientific professional. I am also grateful for having an exceptional doctoral committee, and wish to thank Professor Serge Belongie, Professor Charles Elkan, Professor Kenneth Kreutz-Delgado, Professor Bhaskar D. Rao and Professor Lawrence Saul for their valuable input, accessibility and informative courses.

My research was supported in part by the National Science Foundation (NSF) (PI: Nuno Vasconcelos) the NSF Integrative Graduate and Research Traineeship (NSF-IGERT) (PIs: Virginia de Sa, Garrison Cottrell) the Swartz Center for Computational Neuroscience (director: Scott Makeig) and the UCSD ECE department. I would like to thank everyone involved for providing me with the opportunity to work in this area.

I would like to thank all my colleagues at SVCL, Dr. Dashan Gao, Dr. Antoni Bert Chan, Sunhyoung Han, Nikhil Rasiwasia, Vijay Mahadevan, Mohammad Saberian, Jose Maria Costa Pereira, Kritika Muralidharan and Weixin Li for their support, friendship, assistance throughout the years and their collaboration on some of the experiments.

I would like to thank my friends who have helped me in so many ways both in the good times and the bad : Dr. Mahmoud and Ency Tarokh and their family, Dr. Charles and Sedigheh Kunkel and their family, Dr. Khosrow Behbehani and his family, Dr. Mohammad H. Taghavi and his family, Zeinab Taghavi and

her family, Dr. Koohyar Minoo and his family, Ali Afsahi and his family, Kamaal Martin and his family and Dr. Ebraheem I. Fontain and his family.

I would like to thank Mr. Moghaddam and Mr. Jafary, my middle school principle and Farsi teacher, for believing in me when no one else did.

I would like to thank my extended family, grandparents, my many aunts, uncles and cousins for being my support and foundation throughout the years and during my most difficult times. I reserve a special thanks for my brother, Alireza Masnadi-Shiraz, for being my brother-in-arms and my sister, Maryam Masnadi-Shirazi, for her endless kindness and grace. Last but not least, I would like to thank my mother, Mehri Daneshvar, and my father, Dr. Mohammad Ali Masnadi-Shirazi for simply everything. I will not even make an attempt at listing the many things you have done for me, I know I can never repay you. I just hope that you are pleased with your son and may God reward you.

The text of Chapter II, in part, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “A Statistical View of Margin Based Risk Minimization” , to be submitted to *Annals of Statistics*, 2011. The dissertation author was a primary researcher and an author of the cited material. The author would like to thank Mohammad Saberian and Alireza Masnadi-Shirazi for helpful discussions and comments.

The text of Chapter III, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “Variable Margin Bayes Consistent Losses for Classifier Design” , submitted for publication to *Journal of Machine Learning Research*, 2010. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter IV, in part, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “On the Design of Loss Functions for Classification: theory, robustness to outliers, and SavageBoost” , in *Neural Information Processing Systems (NIPS)*, 2008 and Hamed Masnadi-Shirazi, Vijay Mahadevan and Nuno Vasconcelos, “On the Design of Robust Classifiers for Computer Vision" , in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. The dissertation author was a primary researcher and an author of the cited material. The author would like to thank Vijay Mahadevan for conducting the tracking experiments in section IV.H.3 and Nikhil Rasiwasia for providing the scene classification data in section IV.H.2.

The text of Chapter V, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, "Risk minimization, probability elicitation, and cost-sensitive SVMs " , in *International Conference on Machine Learning (ICML)*, 2010. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter VI, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, "Cost-Sensitive Boosting" , in *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2010. The dissertation author was a primary researcher and an author of the cited material.

VITA

1998-2003	Bachelor of Science, *Summa Cum Laude* Electrical Engineering, Shiraz University, Iran and University of Texas at Arlington, USA
2003–2007	Master of Science Electrical Engineering (Signal and Image Processing), University of California at San Diego
2005–2011	Research Assistant Statistical and Visual Computing Laboratory Department of Electrical and Computer Engineering University of California at San Diego
2011	Doctor of Philosophy Electrical Engineering (Signal and Image Processing), University of California at San Diego

PUBLICATIONS

Hamed Masnadi-Shirazi and Nuno Vasconcelos. A Statistical View of Margin Based Risk Minimization. to be submitted, *Annals of Statistics*.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Variable Margin Bayes Consistent Losses for Classifier Design. submitted, *Journal of Machine Learning Research*.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Cost-Sensitive Boosting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33(2), pp. 294, March 2010.

Mohammad Saberian, Hamed Masnadi-Shirazi and Nuno Vasconcelos. TaylorBoost: First and Second-order Boosting Algorithms with Explicit Margin Control. *To appear in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Variable margin losses for classifier design. *Proceedings of Neural Information Processing Systems (NIPS)*, 2010.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive SVMs. *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

Hamed Masnadi-Shirazi, Nuno Vasconcelos and Vijay Mahadevan. On the Design of Robust Classifiers for Computer Vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. On the Design of Loss Functions for Classification: theory, robustness to outliers, and SavageBoost. *Proceedings of Neural Information Processing Systems (NIPS)*, 2008.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. High Detection-rate Cascades for Real-Time Object Detection. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Asymmetric Boosting. *Proceedings of International Conference on Machine Learning (ICML)*, 2007.

ABSTRACT OF THE DISSERTATION

The design of Bayes consistent loss functions for classification

by

Hamed Masnadi-Shirazi
Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)
University of California, San Diego, 2011
Professor Nuno Vasconcelos, Chair

The combination of using loss functions that are both Bayes consistent and margin enforcing has lead to powerful classification algorithms such as AdaBoost that uses the exponential loss and logistic regression and LogitBoost that use the logistic loss. The use of Bayes consistent margin enforcing losses along with efficient optimization techniques has also lead to other successful classification algorithms such as SVM classifiers that use the hinge loss function. The success of boosting and SVM classifiers is not surprising when looked at from the standpoint of Bayes consistency. Such algorithms are all based on Bayes consistent loss functions and so are guaranteed to converge to the Bayes optimal decision rule as the number of training samples increases. Despite the importance and success of Bayes consistent loss functions, the number of such known loss functions has remained small in the literature. This is in part due to the fact that a generative method for deriving such loss functions did not exist. Not having a generative method for deriving Bayes consistent loss functions not only prevents one from effectively designing loss functions with certain shapes, but also prevents a full analysis and taxonomy of the possible shapes and properties that such loss function can have. In this thesis we solve these problems by providing a generative method for deriving Bayes consistent loss functions. We also fully analyze such loss functions and explore the design of loss functions with certain shapes and properties. This is achieved by studying and relating the two fields of risk minimization in machine learning and probability elicitation in statistics. Specifically, the class of Bayes consistent loss functions is partitioned into different varieties based on their convexity properties. The convexity properties of the loss and associated risk of Bayes consistent loss functions are also studied in detail which, for the first time, enable the derivation of non convex Bayes consistent loss functions. We also develop a fully constructive method for the derivation of novel canonical loss functions. This is due to a simple connection between the associated minimum conditional risk and optimal link functions. The added insight allows us to derive variable margin losses with explicit margin control. We then establish a common boosting framework, canonical gradientBoost, for building boosting classifiers from all canonical losses. Next, we extend the probability elicitation view of loss function design to the problem of designing robust loss functions for classification. The robust Savage loss and corresponding SavageBoost algorithm are derived and shown to outperform other boosting algorithms on a set of experiments designed to test the robustness of the algorithms to outliers in the training data. We also argue that a robust loss should penalizes both large positive and large negative margins. The Tangent loss and the associated TangentBoost classifier are derived with the desired robust properties. We also develop a general framework for the derivation of Bayes consistent cost sensitive loss functions. This is then used to derive a novel cost sensitive hinge loss function. A cost-sensitive SVM learning algorithm is then derived. Unlike previous SVM algorithms, the one now proposed is shown to enforce cost sensitivity for both separable and non-separable training data, independent of the choice of slack penalty. Finally, we present a novel framework for the design of cost-sensitive boosting algorithms. The proposed framework is used to derive cost-sensitive extensions of AdaBoost, RealBoost and LogitBoost. Experimental evidence, over different machine learning and computer vision problems is presented in support of the new algorithms.

Chapter I

Introduction

I.A Bayes consistent loss functions for classification

When dealing with classification problems, the Bayes decision rule is considered optimal in the sense that it minimizes the average probability of error [28]. Implementing the Bayes decision rule requires estimating the posterior probability distribution for the classes. Estimating a probability distribution can be difficult and unreliable especially when the true form of the distribution is not know, many parameters need to be estimated or when only limited amounts of high dimensional training data is available. Under such circumstances, an arguably better approach could be to learn the classifier decision function directly. In this method a decision function is learned directly so as to minimize the average error over the training set. Obviously a decision function that minimizes the probability of error would be directly implementing the Bayes decision rule by definition.

In the more general case, the average loss over the training data can be minimized, where the loss function can be chosen depending on the problem. The average loss is customarily called the risk. Under this setting, the average error is a special case of minimizing the average loss when the zero one loss function is used where a unit value of loss is assigned to any misclassified data point and zero loss is assigned to any correctly classified data point. Using a loss function other than the zero one loss function is desirable because the zero one loss function is non-differentiable and can lead to a difficult optimization problem. On the other hand not just any differentiable function can be used as the loss function to overcome this problem. We still require that the loss function be such that we arrive at the optimal Bayes decision rule function after minimizing the associated risk. When the resulting classifier converges asymptotically to the Bayes decision rule, as training samples increase, the loss is said to be Bayes consistent [35, 17, 119, 57]. For example the zero one loss is a Bayes consistent loss function by definition.

Other known examples of Bayes consistent loss functions include the hinge loss, the exponential loss, and the logistic loss. Apart from being differentiable, these losses assign a penalty to examples classified correctly but close to the boundary. This guarantees a classification margin, and improved generalization when learning from finite datasets [98]. Such loss functions are called margin enforcing. This is unlike the zero one loss which is neither differentiable nor margin enforcing.

The combination of using loss functions that are both Bayes consistent and margin enforcing has lead to powerful classification algorithms such as AdaBoost that uses the exponential loss [33, 35] and logistic regression and LogitBoost that use the logistic loss [35]. The use of Bayes consistent margin enforcing losses along with efficient optimization techniques has also lead to other successful classification algorithms such as SVM classifiers [22] that use the hinge loss function [119, 59].

Boosting and SVM classifiers, based on Bayes consistent loss functions, have been successfully applied to countless classification problems. For example, major advances have been achieved in computer vision tasks that can be formulated as classification problems. The AdaBoost algorithm alone has found multiple applications in vision, e.g. real-time object detection [105, 102, 103, 38, 58], tracking [7], and segmentation [109]. Traditional machine learning problems such as fraud detection [37, 59, 100], text filtering [87], natural language processing [21, 40] and medical diagnosis [65] have also made use of classification algorithms such as boosting [85] and SVMs. Even seemingly unrelated fields such as biotechnology have successfully used SVM classifiers in their work [73, 72]. The success of boosting and SVM classifiers is not surprising when looked at from the standpoint of Bayes consistency. Such algorithms are all based on Bayes consistent loss functions and so are guaranteed to converge to the Bayes optimal decision rule as the number of training samples increases.

Despite the importance and success of Bayes consistent loss functions, the number of such known loss functions has remained small in the literature. This is in part due to the fact that a generative method for deriving such loss functions did not exist. One would ideally prefer to custom tailor the loss function based on the intended application of the classifier. For example in a cost sensitive classification problem such as fraud detection where the cost of misclassifying one class is much higher than the other, one would like to have a cost sensitive loss function that could enforce cost sensitive results. When dealing with noisy and outlier ridden data as in the case of many computer vision datasets, one would like to have a robust loss function with tapered growth and non convex shape that could disregard outliers and lead to a robust classifier. When dealing with classification problems with limited training data, one would also like to know the relationship between Bayes consistency and the margin enforcing properties of the loss function, and if variable margin Bayes consistent loss function are available that could possibly improve the classification performance.

I.B Contributions of the thesis

Not having a generative method for deriving Bayes consistent loss functions not only prevents one from effectively designing loss functions with certain shapes, but also prevents a full analysis and taxonomy of the possible shapes and properties that such loss function can have. In this thesis we provide a generative method for deriving Bayes consistent loss function. We also fully analyze such loss functions and explore the design of loss functions with certain shapes and properties. finally we demonstrate the application of custom tailored loss functions to specific classification problems such as cost sensitive classification or robust classification and show improved performance on a variety of training datasets from the machine learning and computer vision literature. The main contributions of the thesis are as follows.

I.B.1 Generative formula and analysis of Bayes consistent loss functions

We present a new framework for the design and analysis of Bayes consistent loss functions. This is achieved by studying and relating the two fields of risk minimization in machine learning and probability elicitation in statistics. The added insight provided by the probability elicitation view allows us to obtain a generative formula for deriving novel Bayes consistent loss functions. In particular, the class of margin enforcing composite losses are considered and it is shown that the classical progression from loss to risk is overly restrictive: once a loss is specified, both the optimal link, and the functional form of the minimum risk are immediately pined down. This is, however, not the case in our progression: it is shown that any functional form of the minimum conditional risk, which satisfies some mild constraints, supports many link and loss function pairs. Hence, once the risk is selected, one degree of freedom remains: by selecting a class of link functions, it is possible to tailor the loss, so as to guarantee classifiers with desirable traits.

Next, a special class of canonical loss functions are studied under this setting where a simple relationship exists between the link and minimal conditional risk. These canonical loss functions are fully considered and analyzed. We then move on to study the general case of non canonical loss functions. The class of Bayes consistent loss functions is partitioned into four varieties based on their convexity properties. The convexity properties of the loss and associated risk of Bayes consistent loss functions are also studied in detail which, for the first time, enable the derivation of non convex Bayes consistent loss functions . Notable results of our analysis are that all loss functions found from our generative formula are margin enforcing and that the loss and associated conditional risk are quasiconvex. We also show that the risk and empirical risk have a unique minimum that can be found in practice with functional gradient descent algorithms. The margin enforcing property makes such loss functions well suited for classification problems and the properties of the risk greatly simplifies the optimization problem associated with minimizing the risk by ensuring that the minimization will not get stuck in local minima. Finally a taxonomy of Bayes consistent loss functions is provided based on their shape and boundedness properties. A large number of novel Bayes consistent losses are derived with different shapes and properties. Also, a series of recipes are provided that can be used as a guide for designing and deriving other novel loss functions that are specially tailored for certain classification problems.

I.B.2 Canonical variable margin Bayes consistent losses

In general, it is difficult to anticipate the properties, and shape, of a loss function that results from combining a certain minimal risk with a certain link function. We address this problem for the class of canonical risks. We derive a complete characterization of the relationships between loss, optimal link, and minimum risk, and also characterize the properties of the loss whenever the optimal link is in the family of inverse sigmoid functions. We then present a general method for deriving canonical loss functions with explicit control of the classification margin. This is applied to deriving variable margin loss functions from existing minimum risks and novel loss functions derived from cumulative distribution functions. This result shows how the set of canonical loss functions is at least as large as the set of zero mean symmetrical pdf functions and how each can be derived from the other.

The practical importance of these results are studied by establishing a common boosting framework, canonical gradientBoost, for all canonical losses, which enables a direct comparison of the impact of the loss on classifier performance. This in turn allows us to directly study the relationship between the margin properties of the loss function and its Bayes consistent properties on a series of classification problems. A number of novel variable margin Bayes consistent loss functions are derived and shown to have higher classification accuracy when compared to their fixed margin counterparts.

I.B.3 Bayes consistent losses for robust classification

We present a new framework for the design of robust Bayes consistent loss functions. These are loss functions that are well suited for classification problems that involve training sets with noise, outliers, ambiguity or lack of labels. In this context, an issue of particular concern is a well known limitation of most current margin-enforcing losses: their unbounded growth with negative margins which leads to poor performance on training sets with noise and outliers. We address this problem by deriving a novel robust Bayes consistent loss, denoted as Savage loss and an associated SavageBoost algorithm . Unlike all previous Bayes consistent loss functions, the Savage loss is bounded for strongly negative values. This is akin to robust loss functions proposed in the statistics literature to reduce the impact of outliers. We demonstrate the robust properties of the Savage loss on a series of experiments involving outliers and show that the SavageBoost boosting algorithm is indeed more outlier resistant than classical methods, such as AdaBoost, RealBoost, and LogitBoost.

Next, we argue that robustness requires a more subtle constraint on the loss than simply bounding its growth for large negative margins: in addition to this, robustness also requires penalizing large positive margins. We present a simple classification problem that demonstrates this point, and show how all existing methods (including SavageBoost) fail in this case. We then derive a set of necessary conditions that any Bayes consistent loss function must satisfy, in order to guarantee a bounded penalty for both large negative and positive margins. These conditions are used to derive a novel robust loss, which we denote by Tangent loss, and an associated boosting algorithm, denoted TangentBoost . Experiments involving various computer vision problems, including scene classification, object tracking, recognition, and MIL show that the proposed algorithm consistently outperforms previous methods.

I.B.4 Bayes consistent losses for cost sensitive classification

We lay the theoretical foundation for cost sensitive loss function design. The derivation of the new cost-sensitive loss functions draw on the connections between risk minimization and probability elicitation and such connections are generalized to the case of cost-sensitive classification. This theory is then used to develop cost-sensitive extensions of state-of-the-art machine learning techniques.

Specifically, we extend the SVM hinge loss, and derive the optimal cost-sensitive learning algorithm as the minimizer of the associated risk. The new hinge loss is minimized by an SVM that 1) implements the cost-sensitive Bayes decision rule, and 2) approximates the cost-sensitive Bayes risk. The resulting SVM algorithm avoids the shortcomings of previous methods, producing cost-sensitive decision rules for both cases of separable and inseparable training data.

We also present a general framework for the cost-sensitive extension of boosting algorithms and consider the problem of how to extend loss functions used in boosting algorithms, based on the theory of cost sensitive loss function design so as to achieve optimal cost-sensitive decision rules. We introduce cost-sensitive versions of the exponential and logistic losses, which underlie AdaBoost, RealBoost, and LogitBoost. Cost-sensitive extensions of the algorithms are derived, and shown to satisfy the necessary conditions for cost-sensitive optimality.

Finally, the performance of the proposed cost-sensitive algorithms is also evaluated through experiments on a variety of cost sensitive machine learning and computer vision problems such as fraud detection, medical diagnosis, business decision making, face detection and car detection.

I.C Organization of the thesis

The rest of the thesis is organized as follows. In Chapter II, we present a new framework for the design and analysis of Bayes consistent loss functions. This is achieved by studying and relating the two fields of risk minimization in machine learning and probability elicitation in statistics. This chapter is the theoretical back bone of the thesis and all other chapters use the fundamental theorems developed in this chapter. In Chapter III we further expand the theory of Bayes consistent loss function design for the special case of canonical loss functions with explicit control of the classification margin. This is applied to deriving variable margin loss functions from existing minimum risks and novel loss functions derived from cumulative distribution functions. In Chapter IV we present a new framework for the design of robust Bayes consistent loss functions. We address classification problems that involve training sets with noise and outliers by deriving and analyzing the novel robust Bayes consistent Savage and Tangent loss functions and their associated SavageBoost and TangentBoost algorithms. In Chapter V we present a new framework for the design of cost sensitive Bayes consistent loss functions. This general theory is then used to derive a novel cost sensitive SVM classifier. In Chapter VI we present a general framework for the cost sensitive extension of loss functions used in boosting algorithms. We introduce cost-sensitive versions of the exponential and logistic losses, and derive cost sensitive extensions of the AdaBoost, RealBoost and LogitBoost algorithms. Finally, conclusions are provided in Chapter VII.

Chapter II

The design of Bayes consistent loss functions

II.A Classification and risk minimization

We start by briefly reviewing the principles of classification, risk minimization, and large-margin classifier design.

II.A.1 Risk minimization

A classifier is a mapping $g : \mathcal{X} \rightarrow \{-1, 1\}$ that assigns a class label $y \in \{-1, 1\}$ to a feature vector $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is some feature space. This mapping is of the form

$$g(\mathbf{x}) = sign[p(\mathbf{x})], \tag{II.1}$$

for some *predictor* $p : \mathcal{X} \rightarrow \mathbb{R}$. If feature vectors are drawn with probability density $P_{\mathbf{X}}(\mathbf{x})$, $P_Y(y)$ is the probability distribution of the labels $y \in \{-1, 1\}$, and $L(\mathbf{x}, y)$ a loss function, the classification risk is

$$R = E_{\mathbf{X},Y}[L(p(\mathbf{x}), y)]. \tag{II.2}$$

For any non-negative loss, the risk is minimized by minimizing the conditional risk $E_{Y|\mathbf{X}}[L(p(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}]$ for every $\mathbf{x} \in \mathcal{X}$. Denoting by $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$ this can be written as

$$C(\eta, p) = \eta L(p, 1) + (1 - \eta) L(p, -1), \tag{II.3}$$

where we have omitted the dependence of η and g on $\mathbf{x}$ for notational convenience. For simplicity of the presentation, we will 1) use this omission in the remainder of this work, and 2) denote the conditional risk as simply the *risk*. It is useful to express p as a composition of two functions

$$p(\mathbf{x}) = f(\eta(\mathbf{x})), \tag{II.4}$$

where $f : [0, 1] \rightarrow \mathbb{R}$ is a *link function*. η maps feature vectors into posterior class probabilities, and f maps these probabilities into predictions on the real line. Note, however, that $p(\mathbf{x})$ is usually not linear in $\mathbf{x}$. The use of a link function is referred to as *probability calibration* in the machine learning literature [75, 50, 112].

For example, a well known method for learning f^{-1} given p, using a sigmoidal nonlinearity, is presented in [75].

II.A.2 The 0-1 loss

A popular loss for classification is the zero-one loss which can be written as

$$\begin{aligned} L_{0/1}(f,y) &= \frac{1-sign(yp)}{2} = \frac{1-sign(yf)}{2} \\ &= \begin{cases} 0, & \text{if } y = sign(f); \\ 1, & \text{if } y \neq sign(f), \end{cases} \end{aligned} \tag{II.5}$$

leading to

$$\begin{aligned} C_{0/1}(\eta,f) &= \eta\frac{1-sign(f)}{2} + (1-\eta)\frac{1+sign(f)}{2} \\ &= \begin{cases} 1-\eta, & \text{if } f \geq 0; \\ \eta, & \text{if } f < 0, \end{cases} \end{aligned} \tag{II.6}$$

It follows that the risk $C(\eta, f)$ is minimized by f^* if $f^* \geq 0$ when $\eta > 1/2$ and $f^* < 0$ when $\eta < 1/2$.

There are usually many $f^*(\eta)$ that satisfy this condition, e.g., $f^*(\eta) = 2\eta - 1$, $f^*(\eta) = \log\frac{\eta}{1-\eta}$ or any other f^* such that

$$sign[f^*(\eta)] = sign[\eta - 1/2]. \tag{II.7}$$

Any of these f^* will produce a classifier equivalent to the optimal Bayes decision rule

$$g^* = sign[f^*(\eta)] \quad \text{with} \quad f^*(\eta) = 2\eta - 1. \tag{II.8}$$

The minimum risk

$$\begin{aligned} C^*_{0/1}(\eta) &= \eta\left(\frac{1}{2} - \frac{1}{2}sign(2\eta-1)\right) + (1-\eta)\left(\frac{1}{2} + \frac{1}{2}sign(2\eta-1)\right) \\ &= E_{Y|\mathbf{X}}\left[\left.\frac{1-yg^*(\mathbf{x})}{2}\right| \mathbf{X} = \mathbf{x}\right] \\ &= P_{Y|\mathbf{X}}[y \neq g^*(\mathbf{x})|\mathbf{X} = \mathbf{x}] \end{aligned}$$

is the probability of classification error for $\mathbf{x}$, also known as the *Bayes error rate* for $\mathbf{x}$.

II.A.3 Large-margin classification

Since $L_{0/1}(\eta, f)$ does not penalize small positive values of yf, it does not encourage the creation of a classification margin [98]. This is known to compromise the generalization ability of the decision rule when learning is based on finite training samples [98]. A number of alternative margin-enforcing losses have been proposed in the machine learning literature [35]. Like the zero-one loss, they have the form

$$L_\phi(p, y) = \phi(yp), \tag{II.9}$$

but rely on functions $\phi(v)$ which are convex upper-bounds of that - $\phi(v) = (1 - sign(v))/2$ - used in (II.5). We refer to these ϕ functions as the *losses* used for classifier design by the different large-margin methods. They are listed in Table II.1. Each loss defines a *risk*

$$C_\phi(\eta, f) = \eta\phi(f) + (1-\eta)\phi(-f), \tag{II.10}$$

which is minimized by any functions f^* such that

$$f^*_\phi(\eta) = \arg\min_f C_\phi(\eta, f). \tag{II.11}$$

These functions are denoted as *optimal links* for the loss ϕ. The *minimum risk* is

$$C^*_\phi(\eta) = \min_f C_\phi(\eta, f) = C_\phi(\eta, f^*_\phi(\eta)). \tag{II.12}$$

The loss ϕ and link f are denoted the *components of the risk* $C_\phi(\eta, f)$. The loss ϕ and optimal link f^*_ϕ are denoted the *components of the minimum risk* $C^*_\phi(\eta)$.

In all cases of Table II.1, $f^*_\phi(\eta)$ satisfies (II.7), and the associated decision rule is asymptotically equivalent to the Bayes decision rule. Whenever this holds, the risk is said to be *Bayes consistent*. As shown in Figure II.1 a) all losses of Table II.1 are also margin enforcing. This leads to classifiers that generalize better

Table II.1 Machine learning algorithms progress from loss ϕ, to inverse link function $f_\phi^*(\eta)$, and minimum conditional risk $C_\phi^*(\eta)$.

Algorithm	$\phi(v)$	$f_\phi^*(\eta)$	$C_\phi^*(\eta)$
Least squares	$(1-v)^2$	$2\eta - 1$	$4\eta(1-\eta)$
Modified LS	$\max(1-v,0)^2$	$2\eta - 1$	$4\eta(1-\eta)$
SVM	$\max(1-v,0)$	$sign(2\eta-1)$	$1-\|2\eta-1\|$
Boosting	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$2\sqrt{\eta(1-\eta)}$
Logistic Regression	$\log(1+e^{-v})$	$\log\frac{\eta}{1-\eta}$	-$\eta\log\eta - (1-\eta)\log(1-\eta)$

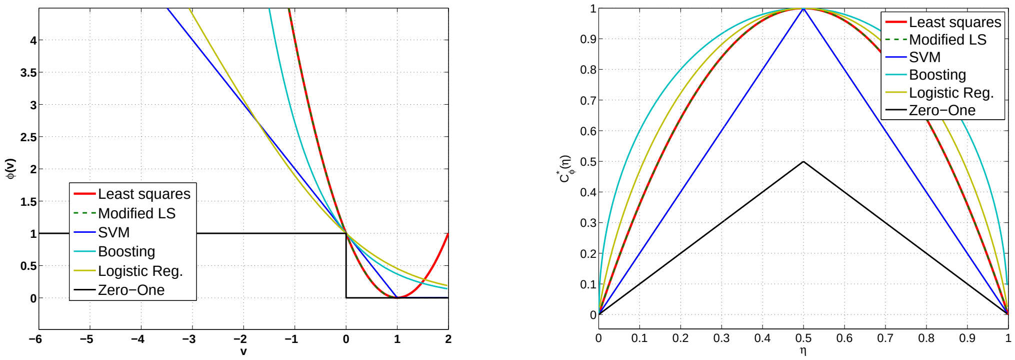


Figure II.1 Loss function $\phi(v)$ (left) and minimum conditional risk $C_\phi^*(\eta)$ (right) associated with the different methods discussed in the text.

than those designed with the 0-1 loss, for finite training samples. The minimum risks $C_\phi(\eta)$ are plotted in Figure II.1 b), along with the Bayes error rate.

II.A.4 Properties of risk minimization

It has been shown that any convex loss $\phi(v)$, differentiable at the origin, such that $\phi'(0) = 0$ is Bayes consistent [11]. The following lemma lists some general properties of the minimum risk and its components. These properties do not require convexity of $\phi(\cdot)$.

Lemma 1. *Let $f^*_\phi(\eta)$ be defined as in (II.11) and $C^*_\phi(\eta)$ as in (II.12). Then,*

$$\begin{aligned} f^*_\phi(\eta) &= -f^*_\phi(1-\eta) && \text{(II.13)} \\ \eta\phi'(f^*_\phi) &= (1-\eta)\phi'(-f^*_\phi) && \text{(II.14)} \\ C^*_\phi(\eta) &= C^*_\phi(1-\eta) && \text{(II.15)} \\ [C^*_\phi]'(\eta) &= \phi(f^*_\phi(\eta)) - \phi(-f^*_\phi(\eta)). && \text{(II.16)} \\ [C^*_\phi]''(\eta) &= \frac{1}{1-\eta}[f^*_\phi]'(\eta)\phi'(f^*_\phi(\eta)). && \text{(II.17)} \end{aligned}$$

Furthermore, if f^*_ϕ is invertible, then

$$\begin{aligned} [f^*_\phi]^{-1}(v) &= 1 - [f^*_\phi]^{-1}(-v) && \text{(II.18)} \\ \phi(v) &= C^*_\phi\{[f^*_\phi]^{-1}(v)\} + (1-[f^*_\phi]^{-1}(v))[C^*_\phi]'\{[f^*_\phi]^{-1}(v)\}. && \text{(II.19)} \end{aligned}$$

Proof. By definition, $f^*_\phi(\eta)$ is the function which minimizes $C_\phi(\eta, f)$. Since the conditional risk has the symmetry

$$C_\phi(\eta, f) = \eta\phi(f) + (1-\eta)\phi(-f) = C_\phi(1-\eta, -f) \qquad \text{(II.20)}$$

it follows that if $f^*_\phi(\eta)$ minimizes $C(\eta, f)$ then $-f^*_\phi(1-\eta)$ minimizes $C(1-\eta, -f)$. Hence, f^*_ϕ has the symmetry of (II.13). Setting derivatives of (II.20) to zero,

$$\eta\phi'(f^*_\phi) - (1-\eta)\phi'(-f^*_\phi) = 0.$$

and (II.14) follows. From the definition of $C^*_\phi(\eta)$,

$$\begin{aligned} C^*_\phi(\eta) &= C_\phi(\eta, f^*_\phi(\eta)) && \text{(II.21)} \\ &= \eta\phi(f^*_\phi(\eta)) + (1-\eta)\phi(-f^*_\phi(\eta)) && \text{(II.22)} \\ &= C_\phi(1-\eta, -f^*_\phi(\eta)) \\ &= C_\phi(1-\eta, f^*_\phi(1-\eta)) \\ &= C^*_\phi(1-\eta) \end{aligned}$$

where we have used the symmetry of (II.13). From (II.22),

$$\begin{aligned} [C_\phi^*]'(\eta) &= \phi(f_\phi^*(\eta)) - \phi(-f_\phi^*(\eta)) + \eta\phi'(f_\phi^*(\eta))[f_\phi^*]'(\eta) \\ &\quad -(1-\eta)\phi'(-f_\phi^*(\eta))[f_\phi^*]'(\eta) \\ &= \phi(f_\phi^*(\eta)) - \phi(-f_\phi^*(\eta)) + [f_\phi^*]'(\eta)\{\eta\phi'(f_\phi^*(\eta)) - (1-\eta)\phi'(-f_\phi^*(\eta))\} \\ &= \phi(f_\phi^*(\eta)) - \phi(-f_\phi^*(\eta)), \end{aligned}$$

where we have used (II.14). Furthermore,

$$\begin{aligned} [C_\phi^*]''(\eta) &= \phi'(f_\phi^*(\eta))[f_\phi^*]'(\eta) + \phi'(-f_\phi^*(\eta))[f_\phi^*]'(\eta), \\ &= [f_\phi^*]'(\eta)\{\phi'(f_\phi^*(\eta)) + \phi'(-f_\phi^*(\eta))\} \\ &= [f_\phi^*]'(\eta)\phi'(f_\phi^*(\eta))\{1 + \frac{\eta}{1-\eta}\} \\ &= \frac{1}{1-\eta}[f_\phi^*]'(\eta)\phi'(f_\phi^*(\eta)) \end{aligned}$$

If f_ϕ^* is invertible, let $\eta = [f_\phi^*]^{-1}(v)$. It follows from (II.13) that

$$\begin{aligned} -v &= f_\phi^*(1-\eta) \\ 1-\eta &= [f_\phi^*]^{-1}(-v) \\ \eta &= 1 - [f_\phi^*]^{-1}(-v) \\ [f_\phi^*]^{-1}(v) &= 1 - [f_\phi^*]^{-1}(-v). \end{aligned}$$

From (II.12),

$$\begin{aligned} C_\phi^*\{[f_\phi^*]^{-1}(v)\} &= [f_\phi^*]^{-1}(v)\phi(v) + (1 - [f_\phi^*]^{-1}(v))\phi(-v) \\ &= \phi(-v) + [f_\phi^*]^{-1}(v)(\phi(v) - \phi(-v)) \end{aligned}$$

and, from (II.16),

$$\begin{aligned} (1 - [f_\phi^*]^{-1}(v))[C_\phi^*]'\{[f_\phi^*]^{-1}(v)\} &= (1 - [f_\phi^*]^{-1}(v))(\phi(v) - \phi(-v)) \\ &= \phi(v) - \phi(-v) - [f_\phi^*]^{-1}(v)(\phi(v) - \phi(-v)). \end{aligned}$$

Adding the two equations leads to (II.19). ■

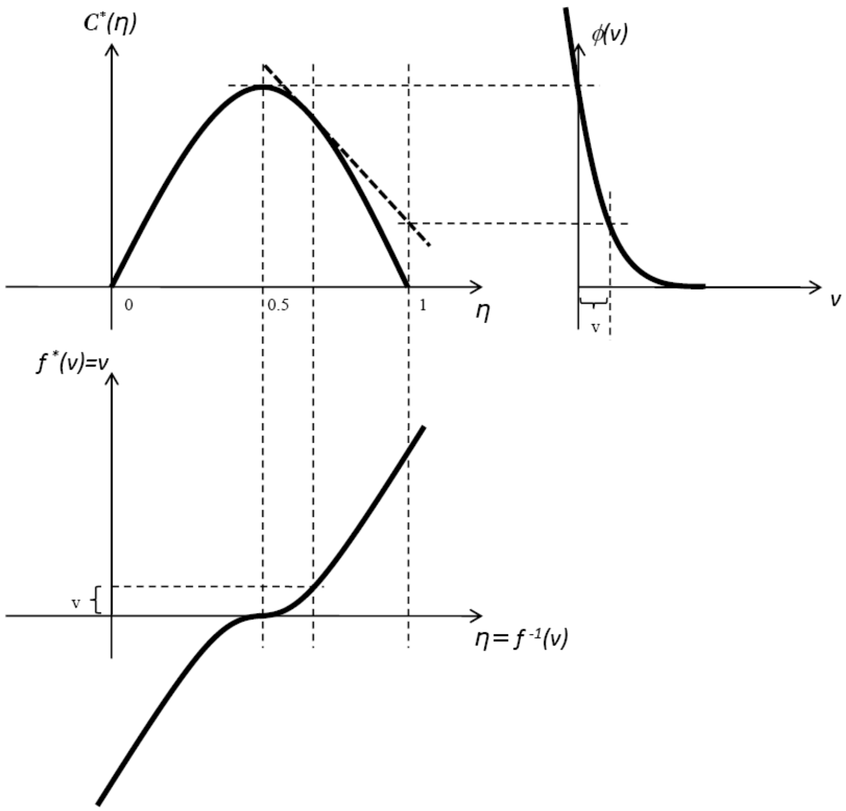


Figure II.2 Relationship between the risk, link and loss function.

Property (II.19) has an interesting geometrical interpretation. Let

$$\nabla C^*_\phi(\eta, \eta_0) = C^*_\phi(\eta_0) + (\eta - \eta_0)[C^*_\phi]'(\eta_0) \tag{II.23}$$

be the linearization of $C^*_\phi(\eta)$ around $\eta = \eta_0$. It follows from (II.19) that

$$\phi(v) = \nabla C^*_\phi(1, [f^*_\phi]^{-1}(v)). \tag{II.24}$$

This defines a geometric relationship between the minimum risk $C^*_\phi(\eta)$, the optimal link $f^*_\phi(v)$, and the loss $\phi(v)$, which is illustrated in Figure II.2.

II.B Classifier design and probability elicitation

Table II.1 shows that the difficulty of learning the predictor $p(\mathbf{x})$ follows from the need to estimate the posterior probability $\eta(\mathbf{x})$. Given the latter, $p(\mathbf{x})$

can be computed with the given optimal links $f^*_\phi(\eta)$. It must be the case, then, that the learning algorithms are estimating $\eta(\mathbf{x})$.

II.B.1 Estimation of posterior probabilities

This has indeed been shown to be the case by Zhang [119], who proved the following results.

Theorem 2. *(Zhang) The minimum conditional risk of (II.12) has the following properties.*

1. *$C^*_\phi(\eta)$ is a a concave function $\eta \in [0, 1]$.*
2. *If f^*_ϕ is differentiable, then $C^*_\phi(\eta)$ is differentiable and, for any $\hat{\eta}$,*

$$C_\phi(\eta, f^*_\phi(\hat{\eta})) - C^*_\phi(\eta) = B_{-C^*_\phi}(\eta, \hat{\eta}), \tag{II.25}$$

where

$$B_F(\eta, \hat{\eta}) = F(\eta) - F(\hat{\eta}) - (\eta - \hat{\eta})F'(\hat{\eta}). \tag{II.26}$$

is the Bregman divergence of the convex function F.

Proof. See [119]. ■

The second property implies that the search for the $\hat{f}(\eta(\mathbf{x}))$ which minimizes (II.10) is equivalent to the search for the probability estimate $\hat{\eta}(\mathbf{x})$ of minimum Bregman divergence with $\eta(\mathbf{x})$, for the Bregman divergence defined by $-C^*_\phi(\eta)$. This view of classifier design places a lot less emphasis on the loss function $\phi(v)$ than the traditional machine learning view of the problem. For example, two losses ϕ_1, ϕ_2 of equal minimum risk $C^*_{\phi_1}(\eta) = C^*_{\phi_2}(\eta)$ lead to the minimization of the same Bregman divergence. They should thus be identically good for classifier design. What matters for the quality of the probability estimation is the form of the minimum risk, and associated Bregman divergence. This raises the question of whether minimizing a cost of the form of (II.10) is the best way to elicit the posterior probability $\eta(\mathbf{x})$.

II.B.2 Probability elicitation

The problem of probability elicitation has been extensively studied in statistics. In particular, Savage studied the design of reward functions that encourage probability forecasters to make accurate predictions [82]. He formalized this problem as the study of *calibrated rewards*.

Definition 1. *Consider a binary problem with two events $y \in \{-1, 1\}$. Let*

- *$I_1(\hat{\eta})$ be the reward for the prediction $\hat{\eta}$ when the event $y = 1$ holds,*
- *$I_{-1}(\hat{\eta})$ be the reward for the prediction $\hat{\eta}$ when the event $y = -1$ holds.*

The expected reward

$$I(\eta, \hat{\eta}) = \eta I_1(\hat{\eta}) + (1 - \eta) I_{-1}(\hat{\eta}). \tag{II.27}$$

is calibrated if it achieves its maximal value when $\hat{\eta} = \eta, \forall \eta$, i.e.

$$I(\eta, \hat{\eta}) \leq I(\eta, \eta) = J(\eta), \;\; \forall \eta \tag{II.28}$$

with equality if and only if $\hat{\eta} = \eta$. $J(\eta)$ is the maximal expected reward.

For simplicity, we will refer to $I(\eta, \hat{\eta})$ as the *reward*, to $J(\eta)$ as the *maximal reward*, and to $I_1(\eta), I_{-1}(\eta)$ as the *conditional rewards*. The definition implies that calibrated rewards are maximized when there is no probability estimation error. Savage asked the question of which conditional rewards lead to a calibrated reward, and showed that the following holds.

Theorem 3. *(Savage) The reward $I(\eta, \hat{\eta})$ of (II.27) is calibrated, with maximal reward $J(\eta) = I(\eta, \eta)$, if and only if*

1. *$J(\eta)$ is strictly convex,*
2. *$I_1(\eta)$ and $I_{-1}(\eta)$ satisfy*

$$I_1(\eta) \;=\; J(\eta) + (1 - \eta) J'(\eta) \tag{II.29}$$

$$I_{-1}(\eta) \;=\; J(\eta) - \eta J'(\eta). \tag{II.30}$$

In this case, for any pair $(\eta, \hat{\eta})$ there is a reward loss of

$$I(\eta, \eta) - I(\eta, \hat{\eta}) = B_J(\eta, \hat{\eta}). \tag{II.31}$$

Proof. See [82]. ■

When (II.29) and (II.30) hold, we refer to $I(\eta, \hat{\eta})$ as the reward *derived* from $J(\eta)$. The theorem shows that every calibrated reward is derived from some strictly convex maximal reward $J(\eta)$. For any η, the calibrated prediction $\hat{\eta}$ is the one of minimum Bregman divergence $B_J(\eta, \hat{\eta})$ with η. The similarities between Theorems 2 and 3 are quite striking. They suggest that the negative of the risk could be a calibrated reward.

This is investigated in the next section. For now, we note that Savage investigated a related problem, by studying the set of convex functions $J(\eta)$ that, when used in (II.29)-(II.30), lead to Bregman divergences of certain forms. In particular, he showed that for divergences of the form $B_J(\eta, \hat{\eta}) = H(h(\eta) - h(\hat{\eta}))$, with $H(0) = 0$ and $H(v) > 0, v \neq 0$, and $h(v)$ any function, only two cases are possible. In the first $h(v) = v$, i.e. the loss only depends on the difference $\eta - \hat{\eta}$, and the admissible J are

$$J_1(\eta) = k\eta^2 + l\eta + m, \tag{II.32}$$

for some integers (k, l, m). In the second $h(v) = \log(v)$, i.e. the loss only depends on the ratio $\eta/\hat{\eta}$, and the admissible J are of the form

$$J_2(\eta) = m + l\eta - k \log \eta. \tag{II.33}$$

II.B.3 Risk minimization as probability elicitation

In this section, we consider the more general question of the equivalence between (negative) risks defined by losses of the form of (II.9) and calibrated rewards. We start by denoting the set of calibrated rewards by $\mathcal{C}$ and the set of negative risks by $\mathcal{R}$. The following result shows that $\mathcal{R} \subset \mathcal{C}$.

Theorem 4. *Let $C_\phi(\eta, f)$ be the risk defined by a loss $\phi(v)$ as in (II.9). Then*

$$I(\eta, \hat{\eta}) = -C_\phi(\eta, f^*_\phi(\hat{\eta}))$$

is a calibrated reward, of maximum $J(\eta) = -C^*_\phi(\eta)$. The estimate $\hat{\eta} = \eta$ simultaneously minimizes the risk and maximizes the reward. The maximal reward has the symmetry

$$J(\eta) = J(1-\eta). \tag{II.34}$$

Proof. By definition

$$C_\phi(\eta, f^*_\phi(\hat{\eta})) = \eta\phi(f^*_\phi(\hat{\eta})) + (1-\eta)\phi(-f^*_\phi(\hat{\eta})),$$

and

$$I(\eta, \hat{\eta}) \;\; = \;\; -\eta\phi(f^*_\phi(\hat{\eta})) - (1-\eta)\phi(-f^*_\phi(\hat{\eta})).$$

Defining

$$\begin{aligned} I_1(\eta) &= -\phi(f^*_\phi(\eta)) && \text{(II.35)} \\ I_{-1}(\eta) &= -\phi(-f^*_\phi(\eta)) && \text{(II.36)} \end{aligned}$$

it follows that

$$I(\eta, \hat{\eta}) \;\; = \;\; \eta I_1(\hat{\eta}) + (1-\eta)I_{-1}(\hat{\eta})$$

is a reward of the form of (II.27), with maximum $J(\eta) = -C^*_\phi(\eta)$. To show that this reward is calibrated, we need to verify that the conditions of Theorem 3 hold. The convexity of $J(\eta)$ follows from the concavity of $C^*_\phi(\eta)$. Using (II.16) and the fact that $C^*_\phi(\eta) = C_\phi(\eta, f^*_\phi(\eta))$,

$$\begin{aligned} & J(\eta) + (1-\eta)J'(\eta) = \\ & \quad = \;\; -C^*_\phi(\eta) - (1-\eta)[C^*]'_\phi(\eta) \\ & \quad = \;\; -\eta\phi(f^*_\phi(\eta)) - (1-\eta)\phi(-f^*_\phi(\eta)) - (1-\eta)\{\phi(f^*_\phi(\eta)) - \phi(-f^*_\phi(\eta))\} \\ & \quad = \;\; -\phi(f^*_\phi(\eta)) = I_1(\eta), \end{aligned}$$

and

$$
\begin{aligned}
&J(\eta) - \eta J'(\eta) = \\
&= \quad -C_\phi^*(\eta) + \eta [C^*]'_\phi(\eta) \\
&= \quad -\eta\phi(f_\phi^*(\eta)) - (1-\eta)\phi(-f_\phi^*(\eta)) + \eta\{\phi(f_\phi^*(\eta)) - \phi(-f_\phi^*(\eta))\} \\
&= \quad -\phi(-f_\phi^*(\eta)) = I_{-1}(\eta).
\end{aligned}
$$

Hence, (II.29)-(II.30) also hold and $-C_\phi(\eta, f_\phi^*(\hat{\eta}))$ is a calibrated reward. Finally,the joint optimality of $\eta = \hat{\eta}$ follows from (II.28), (II.31), and the symmetry of (II.34) from (II.15). ■

II.B.4 Probability elicitation as risk minimization

We have so far shown that the minimization of the risk of (II.10) is a calibrated form of probability elicitation. The converse question is whether all calibrated probability elicitation procedures can be expressed as risk minimization. Or, more formally, whether $\mathcal{C} \subset \mathcal{R}$. For this, we seek conditions on the set of $J(\eta)$ that, when used in (II.29)-(II.30), results in a calibrated reward such that

$$
\begin{aligned}
I_1(\eta) &= -\phi(f_\phi^*(\eta)) \qquad \text{(II.37)} \\
I_{-1}(\eta) &= -\phi(-f_\phi^*(\eta)) \qquad \text{(II.38)}
\end{aligned}
$$

for some loss $\phi(\cdot)$, and optimal link $f_\phi^*(\eta)$. We start by showing that this set does not include all strictly convex $J(\eta)$.

Lemma 5. *Consider the strictly convex function*

$$J(\eta) = -\log \eta \qquad \text{(II.39)}$$

associated with the Bregman divergence

$$B_J(\eta, \hat{\eta}) = \frac{\eta}{\hat{\eta}} - \log \frac{\eta}{\hat{\eta}} - 1$$

commonly referred to as the Itakura-Saito distortion [66]. Consider the calibrated reward $I(\eta, \hat{\eta})$ with conditional rewards $I_1(\eta), I_{-1}(\eta)$ derived from $J(\eta)$ with (II.29)-(II.30). There is no pair of $(\phi(\cdot), f_\phi^*(\eta))$ such that (II.37)-(II.38) hold.

Proof. Assume that there is a pair $(\phi(\cdot), f^*_\phi(\eta))$ such that (II.37)-(II.38) hold. Then

$$\begin{aligned} C(\eta, f^*_\phi(\hat{\eta})) &= -I(\eta, \hat{\eta}) \\ &= \eta\phi(f^*_\phi(\hat{\eta})) + (1-\eta)\phi(-f^*_\phi(\hat{\eta})) \end{aligned}$$

is a risk defined by a loss of the form of (II.9). By Theorem 4, it follows that $C(\eta, f^*_\phi(\hat{\eta}))$ has minimum $C^*_\phi(\eta) = -J(\eta)$, and

$$C^*_\phi(\eta) \neq C^*_\phi(1-\eta).$$

This contradicts (II.15). ■

The lemma shows that not all Bregman divergences $B_J(\eta, \hat{\eta})$ can be minimized by minimizing a risk of the form of (II.10). In fact, any $C_\phi(\eta, f)$ for which this is true must satisfy all properties of Lemma 1, with $C^*_\phi(\eta) = -J(\eta)$. It follows that (II.13)- (II.17) (with $C^*_\phi(\eta) = -J(\eta)$) are necessary conditions for the equivalence between probability elicitation and risk minimization. We next show that these conditions are redundant. We start by deriving the set of necessary and sufficient conditions for the conditional rewards to have the form of (II.37)-(II.38).

Theorem 6. *Let $J(\eta)$ be a strictly convex and continuously differentiable function,*

$$\begin{aligned} I_1(\eta) &= J(\eta) + (1-\eta)J'(\eta) && \text{(II.40)} \\ I_{-1}(\eta) &= J(\eta) - \eta J'(\eta), && \text{(II.41)} \end{aligned}$$

and $f(\eta)$ any invertible function with symmetry

$$f^{-1}(-v) = 1 - f^{-1}(v). \tag{II.42}$$

Then there is a function $\phi(v)$ such that

$$\begin{aligned} I_1(\eta) &= -\phi(f(\eta)) && \text{(II.43)} \\ I_{-1}(\eta) &= -\phi(-f(\eta)) && \text{(II.44)} \end{aligned}$$

if and only if

$$J(\eta) = J(1-\eta). \tag{II.45}$$

Proof. Assume that (II.45) holds, let $v = f(\eta)$ and define $\phi(v)$ as

$$\phi(v) \;=\; -I_1(f^{-1}(v)) = -J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v)).$$

From the symmetry of f and J, and the fact that $J'(\eta) = -J'(1-\eta)$, it follows that

$$\begin{aligned}\phi(-v) &= -J(f^{-1}(-v)) - (1-f^{-1}(-v))J'(f^{-1}(-v))\\ &= -J(1-f^{-1}(v)) - f^{-1}(v)J'(1-f^{-1}(v))\\ &= -J(f^{-1}(v)) + f^{-1}(v)J'(f^{-1}(v))\\ &= -I_{-1}(f^{-1}(v)).\end{aligned}$$

Hence, there is a $\phi(v)$ such that (II.43) and (II.44) hold. To prove the converse, assume that (II.43), and (II.44) hold and let $v = f(\eta)$. Then, from (II.43) and (II.44)

$$\begin{aligned}I_1[f^{-1}(v)] &= -\phi(v) \qquad \text{(II.46)}\\ I_{-1}[f^{-1}(v)] &= -\phi(-v),\end{aligned}$$

and

$$I_{-1}[f^{-1}(v)] = I_1[f^{-1}(-v)].$$

Using (II.42)

$$I_{-1}[f^{-1}(v)] = I_1[1-f^{-1}(v)],$$

and

$$I_{-1}(\eta) = I_1(1-\eta).$$

From (II.40) and (II.41), this implies that

$$J(\eta) - \eta J'(\eta) = J(1-\eta) + \eta J'(1-\eta)$$

or

$$J(\eta) - J(1-\eta) = \eta[J'(\eta) + J'(1-\eta)]. \tag{II.47}$$

This implies that $J(0) = J(1)$. For $\eta \notin \{0,1\}$, take derivatives on both sides of (II.47). Then

$$J'(\eta) + J'(1-\eta) = J'(\eta) + J'(1-\eta) + \eta[J''(\eta) - J''(1-\eta)],$$

from which it follows that

$$J''(\eta) = J''(1-\eta).$$

This implies that

$$J'(\eta) = -J'(1-\eta) + k$$

for some constant k. Since, from (II.47), $J'(1/2) = 0$ it follows that $k = 0$. This implies that

$$J(\eta) = J(1-\eta) + k$$

for some constant k. From $J(0) = J(1)$ it follows that $k = 0$, showing that (II.45) holds. ■

Note that, from Theorem 3, the reward with components $I_1(\eta)$ and $I_{-1}(\eta)$ is calibrated. Hence, Theorem 6 implies that any calibrated reward derived from a $J(\eta)$ with the symmetry of (II.45) is a negative risk. This proves the following theorem.

Theorem 7. *Let $I(\eta, \hat{\eta})$ be a calibrated reward derived from a maximal reward of symmetry*

$$J(\eta) = J(1-\eta).$$

Then for any invertible link $f^*_\phi(\eta)$ with symmetry

$$[f^*_\phi]^{-1}(-v) = 1 - [f^*_\phi]^{-1}(v)$$

there is a loss $\phi(v)$ of the form of (II.9) such that

$$I(\eta, \hat{\eta}) = -C_\phi(\eta, f^*_\phi(\hat{\eta})).$$

The risk has minimum $C^*_\phi(\eta) = -J(\eta)$, and the estimate $\hat{\eta} = \eta$ simultaneously minimizes the risk and maximizes the reward. The link and loss are related by

$$\phi(v) \;=\; -J\{[f^*_\phi]^{-1}(v)\} - (1 - [f^*_\phi]^{-1}(v))J'\{[f^*_\phi]^{-1}(v)\}. \tag{II.48}$$

Note that (II.48) follows from $C^*_\phi(\eta) = -J(\eta)$ and (II.19).

II.B.5 Discussion

Theorems 4 and 7 establish an equivalence relationship between risks derived from losses of the form of (II.9) and calibrated rewards derived from convex maximal rewards with the symmetry of (II.45). In particular, all such risks are (negative) calibrated rewards and vice-versa. From Theorems 2 and 3 it follows that the optimization carried out by all machine learning algorithms is equivalent to Savage's procedure for probability elicitation. Both procedures reduce to the minimization of the Bregman divergence

$$\hat{\eta}^* = \arg\min_{\hat{\eta}} B_J(\eta, \hat{\eta}), \tag{II.49}$$

where $J(\eta) = -C^*_\phi(\eta)$ is a convex function such that $J(\eta) = J(1 - \eta)$. In both cases, the predictions $\hat{\eta}^*$ are calibrated.

Here it is necessary to make an important caveat. While the previous theorems are suitable for generating Bayes consistent losses, they do not necessarily lead to so called proper losses and calibrated reward functions [82, 17] unless an additional condition is satisfied. The required additional condition is that the domain of $f^*_\phi(\eta)$ be restricted to $[0 \;\; 1]$. Under this condition $(f^*_\phi)^{-1}(v)$ will have a range of $[0 \;\; 1]$ and can be used to recover the estimates η which can now be interpreted as true probabilities given that their values are confined to $[0 \;\; 1]$. Thus, it should be understood that whenever the terms proper loss and calibrated score function are used, the additional condition on $f^*_\phi(\eta)$ is implied.

For example, the logistic loss is a Bayes consistent and proper loss function because the domain of $f^*_\phi(\eta) = \frac{\eta}{1-\eta}$ is $[0 \;\; 1]$. On the other hand, the least squares loss is Bayes consistent but not a proper loss function since $f^*_\phi(\eta) = 8\eta - 4$ does not have a domain of $[0 \;\; 1]$. Yet, the least squares loss can be made to be partially proper by restricting the domain of its link function to $[0 \;\; 1]$, which would in turn alter the loss and restrict its domain to $[-4 \;\; 4]$.

Given that we are mainly concerned with classification problems, we will not restrict ourselves to proper losses. This will be of particular use when dealing with outliers which do not necessarily have a probabilistic interpretation (see Chapter IV), or when dealing with the SVM classifier which uses the hinge loss function which is not a proper loss function (see Chapter V).

While explicit minimization of (II.49) requires access to the probability estimates $\hat{\eta}$, these are not directly observable in machine learning problems. Hence, learning algorithms attack the problem indirectly. They operate in the space of observations $\mathbf{x}$, and start from the loss $\phi(yf(\mathbf{x}))$. This defines the risk $C_\phi(\eta(\mathbf{x}), f(\mathbf{x}))$ which is minimized with respect to $f(\mathbf{x})$ to obtain an estimate $p(\mathbf{x})$ of $f^*_\phi(\eta(\mathbf{x}))$ and the associated minimum risk $C^*_\phi(\eta(\mathbf{x}))$. Upon convergence of $p(\mathbf{x})$ to $f^*_\phi(\eta(\mathbf{x}))$, the probability $\eta(\mathbf{x})$ can be recovered by simple application of

$$\eta(\mathbf{x}) = ([f^*_\phi]^{-1} \circ p)(\mathbf{x}), \tag{II.50}$$

whenever f^*_ϕ is invertible. On the other hand, Savage's procedure operates directly on the space of probability estimates, maximizing (with respect to $\hat{\eta}$) the reward $I(\eta, \hat{\eta})$, derived from $J(\eta)$. The optimal solutions of the two procedures are nevertheless identical if $J(\eta) = -C^*_\phi(\eta)$. In fact, this relation makes it possible to express the learning algorithms in "Savage form", i.e. as procedures for the maximization of (II.27), by deriving the reward functions associated with each of the $C^*_\phi(\eta)$ in Table II.1. This is done by using (II.29) and (II.30) with $J(\eta) = -C^*_\phi(\eta)$, and the results are shown in Table II.2.

A second fundamental difference has to do with the degrees of freedom of risk minimization vs. probability elicitation. On one hand, the minimization

Table II.2 Probability elicitation form for various machine learning algorithms, and Savage's procedure. In Savage 1 and 2 $m' = m + k$.

Algorithm	$I_1(\eta)$	$I_{-1}(\eta)$	$J(\eta)$
Least squares	$-4(1-\eta)^2$	$-4\eta^2$	$-4\eta(1-\eta)$
Modified LS	$-4(1-\eta)^2$	$-4\eta^2$	$-4\eta(1-\eta)$
SVM	$sign[2\eta-1]-1$	$-sign[2\eta-1]-1$	$\lvert 2\eta-1\rvert - 1$
Boosting	$-\sqrt{\frac{1-\eta}{\eta}}$	$-\sqrt{\frac{\eta}{1-\eta}}$	$-2\sqrt{\eta(1-\eta)}$
Log. Regression	$\log\eta$	$\log(1-\eta)$	$\eta\log\eta + (1-\eta)\log(1-\eta)$
Savage 1	$-k(1-\eta)^2 + m' + l$	$-k\eta^2 + m$	$k\eta^2 + l\eta + m$
Savage 2	$-k(1/\eta + \log\eta) + m' + l$	$-k\log\eta + m'$	$m + l\eta - k\log\eta$

Table II.3 Probability elicitation form progresses from minimum conditional risk, and link function $(f^*_\phi)^{-1}(\eta)$, to loss ϕ. $f^*_\phi(\eta)$ is not invertible for the SVM and modified LS methods.

Algorithm	$J(\eta)$	$(f^*_\phi)^{-1}(v)$	$\phi(v)$
Least squares	$-4\eta(1-\eta)$	$\frac{1}{2}(v+1)$	$(1-v)^2$
Modified LS	$-4\eta(1-\eta)$	NA	$\max(1-v,0)^2$
SVM	$\lvert 2\eta-1\rvert - 1$	N/A	$\max(1-v,0)$
Boosting	$-2\sqrt{\eta(1-\eta)}$	$\frac{e^{2v}}{1+e^{2v}}$	$\exp(-v)$
Logistic Regression	$\eta\log\eta + (1-\eta)\log(1-\eta)$	$\frac{e^{v}}{1+e^{v}}$	$\log(1+e^{-v})$

of the risk $C_\phi(f,\eta)$ leaves no degree of freedom for the selection of either the link function $f^*_\phi(\eta)$ or the minimum risk (maximum reward) $C^*_\phi(\eta)$. They are simply the ones that result from the optimization. On the other, Theorem 7 shows that the specification of a maximum reward (minimum risk) does not uniquely define either the loss $\phi(v)$ or link $f^*(\eta)$. Given a $C^*_\phi(\eta)$, there could be multiple pairs (ϕ, f^*_ϕ) for which (II.48) holds. Some intuition for this can be obtained from Figure II.2. The main fact to note is that $f^*_\phi(\eta)$ is determined by ϕ, according to (II.11). Then, given $\phi(v)$ and $f^*_\phi(\eta)$, $C^*_\phi(\eta)$ is uniquely defined. However, a given $C^*_\phi(\eta)$ can be consistent with multiple pairs of $(\phi(v), f^*_\phi(\eta))$. Selecting a particular ϕ only "pins down" f^*_ϕ, from the set of all f^*_ϕ that are compatible with Bayes decision rule. Similarly, choosing a f^*_ϕ "pins down" ϕ. This is the case of the algorithms in Table II.1, for which the associated inverse link functions are presented in Table II.3. From these, and (II.48) it is possible to recover $\phi(v)$, also shown in the table.

II.C Canonical risk minimization

In general, due to the multiplicity of ϕ and f^*_ϕ that satisfy (II.48) for a given $J(\eta) = -C^*_\phi(\eta)$, it is impossible to completely characterize the loss function responsible for a given minimum risk. In this section, we study an exception to this rule.

II.C.1 Canonical risks

We start with a lemma that relates the symmetry conditions, on $J(\eta)$ and $f^*_\phi(\eta)$, of Theorem 7.

Lemma 8. *Let $J(\eta)$ be a strictly convex and differentiable function such that $J(\eta) = J(1-\eta)$. Then $J'(\eta)$ is invertible and*

$$[J']^{-1}(-v) = 1 - [J']^{-1}(v). \tag{II.51}$$

Proof. From the strict convexity of $J(\eta)$ it follows that $J'(\eta)$ has positive derivative for all η . Hence, $J'(\eta)$ is invertible. From the symmetry of $J(\eta)$,

$$J'(\eta) \quad = \quad -J'(1-\eta)$$

and, for any v such that $\eta = [J']^{-1}(v)$,

$$\begin{aligned} v &= -J'(1-[J']^{-1}(v)) \\ [J']^{-1}(-v) &= 1-[J']^{-1}(v). \end{aligned}$$

■

The lemma shows that the equivalence between risks and calibrated rewards requires the derivative of $J(\eta)$ to have the *same* symmetry as the optimal link $f^*_\phi(\eta)$. This suggests that the former can be used as the latter. When this is the case, the conditional risk is in canonical form, and (f^*, J) are said to be a canonical pair [17] .

Definition 2. *Let $J(\eta)$ be a maximal reward, and $C^*_\phi(\eta) = -J(\eta)$ a minimum risk. If the optimal link associated with $C^*_\phi(\eta)$ is*

$$f^*_\phi(\eta) = J'(\eta) \tag{II.52}$$

the risk $C_\phi(\eta, f)$ is said to be in canonical form. $f^*_\phi(\eta)$ is denoted a canonical link and $\phi(v)$, the loss given by (II.19), a canonical loss.

For example, as shown in Table II.3, the risk of logistic regression is derived from the convex and symmetric $J(\eta) = \eta \log(\eta) + (1 - \eta) \log(1 - \eta)$. This has derivative $J'(\eta) = \log(\frac{\eta}{1-\eta})$ and, from Table II.1, $J'(\eta) = f^*_\phi(\eta)$. It is also possible to show that (II.19) holds. Hence, the risk of logistic regression is in canonical form.

II.C.2 Constructing canonical risks from $J(\eta)$

By introducing a one-to-one relationship between $J(\eta)$ and $f^*(\eta)$ (up to an additive constant), (II.52) removes the ambiguity about the $f^*(\eta)$ and $\phi(v)$ associated with a particular minimum risk/maximal reward. In particular, using (II.52) in Theorem 7 leads to the following result.

Theorem 9. *Let $I(\eta, \hat{\eta})$ be a calibrated reward derived from a convex maximal reward of symmetry*

$$J(\eta) = J(1 - \eta).$$

Then

$$I(\eta, \hat{\eta}) = -C_\phi(\eta, -[C^*_\phi]'(\hat{\eta})).$$

where $C_\phi(\eta, f)$ is the risk derived from the loss

$$\phi(v) \quad = \quad -J\{[J']^{-1}(v)\} - (1 - [J']^{-1}(v))v. \tag{II.53}$$

The risk is in canonical form and has minimum $C^*_\phi(\eta) = -J(\eta)$. The estimate $\hat{\eta} = \eta$ simultaneously minimizes the risk and maximizes the reward.

Hence, for canonical risks, there is a one-to-one relationship between loss, minimum risk, and optimal link (up to an additive constant). Note that this is not necessarily true for all risks in common use, which are not necessarily in canonical form. For example, the risk of boosting is derived from the convex, differentiable, and symmetric $J(\eta) = -2\sqrt{\eta(1-\eta)}$. Since this has derivative

$$J'(\eta) = \frac{2\eta - 1}{\sqrt{\eta(1-\eta)}} \neq \frac{1}{2}\log\frac{\eta}{1-\eta} = f^*_\phi(\eta), \tag{II.54}$$

the risk is not in canonical form. What the theorem shows is that *it is possible* to derive a canonical risk for each maximal reward $J(\eta)$. For example, it is possible to derive a canonical form for the $J(\eta)$ of boosting, $J(\eta) = -2\sqrt{\eta(1-\eta)}$. From (II.52)

$$f^*_\phi(\eta) = \frac{2\eta - 1}{\sqrt{\eta(1-\eta)}} \tag{II.55}$$

and, from (II.52) and (II.53)

$$\begin{aligned}\phi(v) &= -J\{[f^*_\phi]^{-1}(v)\} - (1 - [f^*_\phi]^{-1}(v))v \\ &= 2\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]} - [1 - [f^*_\phi]^{-1}(v)]v\end{aligned}$$

Using $\eta = [f^*_\phi]^{-1}(v)$ in both sides of (II.55),

$$v = \frac{2[f^*_\phi]^{-1}(v) - 1}{\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \tag{II.56}$$

and

$$\begin{aligned}\phi(v) &= 2\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]} - \frac{[1 - [f^*_\phi]^{-1}(v)][2[f^*_\phi]^{-1}(v) - 1]}{\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \\ &= \frac{2[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)] - [1 - [f^*_\phi]^{-1}(v)][2[f^*_\phi]^{-1}(v) - 1]}{\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \\ &= \sqrt{\frac{1 - [f^*_\phi]^{-1}(v)}{[f^*_\phi]^{-1}(v)}} \qquad \text{(II.57)}\end{aligned}$$

Finally, solving (II.56) for $[f^*_\phi]^{-1}(v)$,

$$[f^*_\phi]^{-1}(v) = \frac{1}{2} \pm \frac{1}{2}\frac{v}{\sqrt{4 + v^2}}.$$

Of the two solutions, one is monotonically increasing (+ in between the two terms) with v, and the other decreasing ($-$). Enforcing the constraint of an increasing link function leads to

$$[f^*_\phi]^{-1}(v) = \frac{1}{2} + \frac{1}{2}\frac{v}{\sqrt{4+v^2}},$$

and

$$\phi(v) = \sqrt{\frac{\sqrt{4+v^2}-v}{\sqrt{4+v^2}+v}}.$$

Figure II.3 presents a comparison of these loss and link functions and those associated with logistic regression and boosting. Note that the canonical version of boosting is much closer to logistic regression than to boosting itself. One would thus expect a boosting algorithm derived from the canonical boosting loss to behave very similarly to logitBoost [35]. This procedure can be used to derive canonical versions of the algorithms associated with any $J(\eta)$ in Table II.3, since all of these satisfy the symmetry condition of Theorem 9. The top of Table II.4 presents the canonical loss and inverse link for each of such $J(\eta)$.

Also, Table II.4 includes novel $J(\eta)$ that can be used to derive canonical loss functions. For example the novel $J(\eta;a) = \cosh\left[a\left(\frac{1}{2}-\eta\right)\right] - \cosh(\frac{-a}{2})$ can be used to derive the canonical hyperbolic cosine loss as

$$\begin{aligned}
J(\eta;a) &= \cosh\left[a\left(\frac{1}{2}-\eta\right)\right] - \cosh(\frac{-a}{2}) && \text{(II.58)}\\
[f^*_\phi]^{-1}(v;a) &= \frac{1}{2} - \frac{1}{a}\sinh^{-1}(\frac{-v}{a}) && \text{(II.59)}\\
\phi(v;a) &= -\cosh\left(\sinh^{-1}\left(\frac{-v}{a}\right)\right) - \cosh(\frac{-a}{2}) && \text{(II.60)}\\
&\quad -\left(\frac{1}{2} + \frac{1}{a}\sinh^{-1}\left(\frac{-v}{a}\right)\right)v.
\end{aligned}$$

where $a \in \{-\infty \;\; \infty\}$ is a shape parameter.

II.C.3 Constructing canonical risks from $f^*_\phi(\eta)$

It is also possible to design a canonical risk from any invertible optimal link $f^*_\phi(\eta)$ with the symmetry of (II.42). The first step is to derive the corresponding maximal reward $J(\eta)$, using (II.52). This guarantees that (II.45) holds and

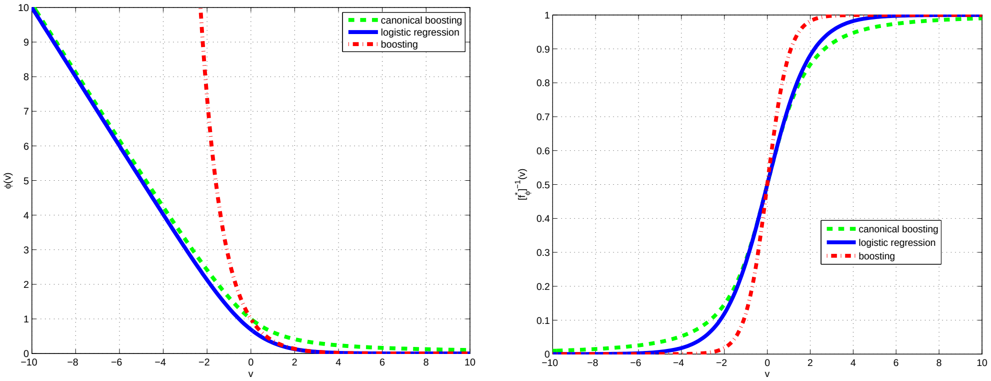


Figure II.3 Loss (left) and link functions (right) of canonical boosting, boosting, and logistic regression.

allows the derivation of the loss $\phi(v)$ using Theorem 9, as was done in the the previous section. For example, the hyperbolic tangent $\tanh v = \frac{e^{2v}-1}{e^{2v}+1}$ is a commonly used non-linearity in the neural network literature [12]. It can be mapped, by converting its range from $[-1, 1]$ to $[0, 1]$, into the link function

$$[f_\phi^*]^{-1}(v) = \frac{1}{2} + \frac{1}{2} \tanh v, \tag{II.61}$$

which is identical to that of boosting $(e^{2v}/(e^{2v}+1))$. The inverse link is

$$f_\phi^*(\eta) = \frac{1}{2} \log \frac{\eta}{1-\eta},$$

which is the canonical inverse link for the maximal expected reward

$$J(\eta) = \frac{1}{2}[\eta \log \eta + (1-\eta)\log(1-\eta)].$$

The associated canonical loss is

$$\phi(v) = \frac{1}{2} \log(1 + e^{-2v}).$$

Note that, once again, the components $J(\eta)$ and $\phi(v)$ of the canonical risk are much closer to those of logistic regression than to those of boosting.

II.C.4 Properties of the canonical form

While canonical risks can be easily designed from either $J(\eta)$ or $f_\phi^*(\eta)$, it is much less clear how to design a loss $\phi(v)$ that guarantees a canonical risk. The

Table II.4 Canonical groupings of maximal reward $J(\eta)$, link function $(f_\phi^*)^{-1}(\eta)$, and loss ϕ. The top of the table shows the components of the canonical risk derived from the maximal reward shown in the second column. The bottom of the table shows those derived from the link function in the third column.

Algorithm	$J(\eta)$	$(f_\phi^*)^{-1}(v)$	$\phi(v)$
Canonical Least Squares	$-4\eta(1-\eta)$	$\frac{1}{8}(v+4)$	$\frac{1}{16}(4-v)^2$
Canonical Boosting	$-2\sqrt{\eta(1-\eta)}$	$\frac{1}{2}+\frac{1}{2}\frac{v}{\sqrt{4+v^2}}$	$\sqrt{\frac{\sqrt{4+v^2}-v}{\sqrt{4+v^2}+v}}$
Logistic Regression	$\eta\log\eta+(1-\eta)\log(1-\eta)$	$\frac{e^v}{1+e^v}$	$\log(1+e^{-v})$
Canonical Hyperbolic Cosine	$\cosh(a(\frac{1}{2}-\eta))-\cosh(\frac{-a}{2})$	$\frac{1}{2}-\frac{1}{a}\sinh^{-1}(\frac{-v}{a})$	(II.60)
Canonical Secant	$\sec(a(\frac{1}{2}-\eta))-\sec(\frac{-a}{2})$	$\frac{1}{2}-\frac{1}{a}\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v})$	(II.75)
Canonical tanh link	$\frac{1}{2}\left[\eta\log\eta+(1-\eta)\log(1-\eta)\right]$	$\frac{e^{2v}}{1+e^{2v}}$	$\log\sqrt{1+e^{-2v}}$
Canonical arctan link	$-\frac{1}{a}\log\left(\frac{\cos[(\eta-\frac{1}{2})a]}{\cos\frac{a}{2}}\right)$	$\frac{1}{2}+\frac{1}{a}\arctan v$	(II.71)

following lemma solves this problem by relating $\phi(v)$ to $J(\eta)$ and $f_\phi^*(\eta)$.

Lemma 10. *Let $C_\phi(\eta, f)$ be the canonical risk derived from a convex and symmetric reward $J(\eta)$, as in Theorem 9. Then*

$$\phi'(v) = -[J']^{-1}(-v) = [f_\phi^*]^{-1}(v) - 1. \tag{II.62}$$

Proof. The lemma follows from taking derivatives on both sides of (II.53),

$$\begin{aligned}
\phi'(v) &= -J'\{[J']^{-1}(v)\}\{[J']^{-1}\}'(v) - (1-[J']^{-1}(v)) + \{[J']^{-1}\}'(v)v \\
&= -v\{[J']^{-1}\}'(v) - (1-[J']^{-1}(v)) + \{[J']^{-1}\}'(v)v \\
&= -(1-[J']^{-1}(v)) \\
&= -[J']^{-1}(-v),
\end{aligned}$$

where we have also used (II.51). The equality with $[f_\phi^*]^{-1}(v) - 1$ follows from (II.52). ■

The lemma has various interesting consequences. First, it specifies the one-to-one mapping (up to additive constants) between the three components (loss, link, and maximal reward/minimum risk) of a canonical risk. This is illustrated in Figure II.4, the equivalent of Figure II.2 for a canonical risk. Note that,

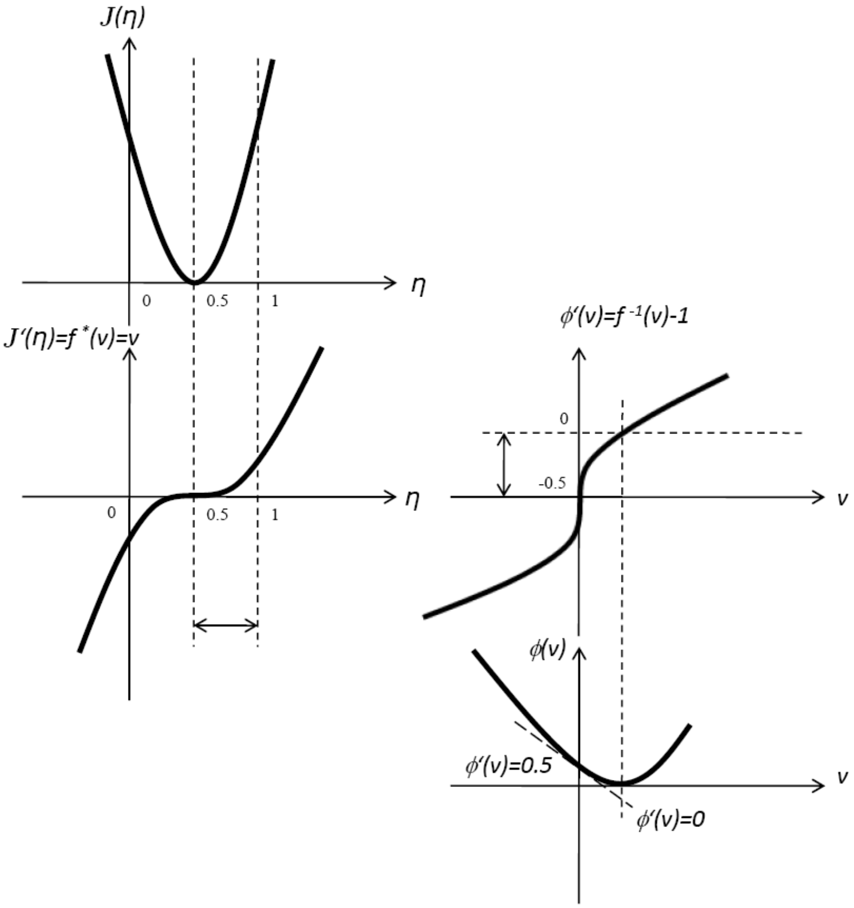


Figure II.4 Relationship between the canonical optimal risk, link and loss function.

in Figure II.4, the specification of $\phi(v)$ no longer leaves $[f^*_\phi]^{-1}(v)$ unconstrained. Instead, it makes $[f^*_\phi]^{-1}(v)$ equal to $1 + \phi'(v)$. $J([f^*_\phi]^{-1}(v))$ is then constrained by (II.24). Second, the lemma establishes a necessary condition for the canonical form that is easy to verify. For example, logistic regression has $-[J']^{-1}(-v) = [f^*_\phi]^{-1}(v) = \frac{1}{1+e^{-v}}$ and $\phi'(v) = -\frac{e^{-v}}{1+e^{-v}} = [f^*_\phi]^{-1} - 1$, while boosting has $[f^*_\phi]^{-1} = \frac{1}{1+e^{-2v}}$ and $\phi'(v) = -e^{-v} \neq [f^*_\phi]^{-1} - 1$. This (plus the symmetry of J and f^*_ϕ) shows that the former is in canonical form but the latter is not. Finally, (II.62) leads to the following precise characterization of the set of canonical losses.

Theorem 11. *Let $C_\phi(\eta, f)$ be the conditional risk of the loss ϕ, as defined in (II.10).*

$C_\phi(\eta, f)$ is a canonical risk if and only if

$$\phi_{odd}(v) = -\frac{v}{2}, \tag{II.63}$$

where

$$\phi_{odd}(v) = \frac{\phi(v) - \phi(-v)}{2} \tag{II.64}$$

is the odd component of $\phi(v)$,

Proof. Consider $J(\eta)$ as in Theorem 9. Then, from Lemmas 8 and 10, $C_\phi(\eta, f)$ is a canonical risk if and only if

$$\begin{aligned}\phi'(v) &= -[J']^{-1}(-v)\\ &= [J']^{-1}(v) - 1\\ &= -\phi'(-v) - 1.\end{aligned}$$

In this case, from (II.53),

$$\begin{aligned}\phi(v) &= -J\{[J']^{-1}(v)\} - (1 - [J']^{-1}(v))v\\ &= -J\{-\phi'(-v)\} - (1 + \phi'(-v))v.\end{aligned}$$

It follows that

$$\phi(-v) = -J\{-\phi'(v)\} + (1 + \phi'(v))v$$

and

$$\begin{aligned}\phi(v) - \phi(-v) &= J\{-\phi'(v)\} - J\{-\phi'(-v)\} - (2 + \phi'(v) + \phi'(-v))v\\ &= J\{-\phi'(v)\} - J\{1 + \phi'(v)\} - v.\end{aligned}$$

Using the symmetry of $J(\eta)$, this is equivalent to

$$\phi(v) - \phi(-v) = -v$$

and the theorem follows. ∎

This theorem states that all canonical losses are of the form

$$\phi(v) = -\frac{v}{2} + \Psi(v) \tag{II.65}$$

where $\Psi(v)$ is an even function. Using Lemma 10 it can be easily shown that the associated link functions are of the form

$$[f_\phi^*]^{-1}(v) = \frac{1}{2} + \Psi'(v), \tag{II.66}$$

where $\Psi'(v)$ is an odd function. If $\Psi'(v)$ is invertible it also implies that

$$J'(\eta) = f_\phi^*(\eta) = [\Psi']^{-1}(\eta - 1/2). \tag{II.67}$$

Hence, up to constants, all the degrees of freedom of $\phi(v)$ are consumed by the specification of either $J(\eta)$ or $f_\phi^*(\eta)$. The next theorem shows that this results in a number of properties for the loss, link, and minimum of a canonical risk.

Theorem 12. *Let $C_\phi(\eta, f)$ be a canonical risk of loss $\phi(v)$, with optimal link $f_\phi^*(\eta)$, and maximal reward $J(\eta) = -C_\phi^*(\eta)$. Then, $\phi(v)$ has the following properties*

1. *$\phi(v)$ is 1) convex and 2) strictly convex if $J(\eta)$ is bounded*
2. *$\phi_{odd}(v) = -v/2$*
3. *$\phi'(v) + \phi'(-v) = -1$*
4. *$\phi'(0) = -1/2$*
5. *$\phi''(v) = \phi''(-v)$.*

The following properties hold for $f_\phi^*(\eta)$

1. *$f_\phi^*(\eta)$ is monotonically increasing*
2. *$f_\phi^*(\eta) = -f_\phi^*(1-\eta)$*
3. *$f_\phi^*(1/2) = 0$*
4. *$[f_\phi^*]^{-1}(v)$ exists and is monotonically increasing*

5. $[f^*_\phi]^{-1}(v) = 1 - [f^*_\phi]^{-1}(-v)$

6. $[f^*_\phi]^{-1}(0) = 1/2.$

The following properties hold for $J(\eta)$

1. $J(\eta)$ *is strictly convex*

2. $J(\eta) = J(1-\eta).$

The following properties hold for the relationship between $\phi(v)$, $f^*_\phi(\eta)$, *and* $J(\eta)$.

1. $J'(\eta) = f^*_\phi(\eta)$

2. $J''(\eta) = 1/\phi''[f^*_\phi(\eta)]$

3. $[f^*_\phi]^{-1}(v) = 1 + \phi'(v)$

4. $\phi'[f^*_\phi(\eta)] = \eta - 1$

5. $J'[1+\phi'(v)] = v$

6. $J[1+\phi'(v)] = v\phi'(v) - \phi(v).$

Finally, $C_\phi(\eta, f)$ *is Bayes consistent.*

Proof. We start by proving the relationships between $\phi(v)$, $f^*_\phi(\eta)$, and $J(\eta)$. Property 1 follows from the definition of canonical risk. Property 3 follows from Lemma 10. Property 4 follows from Property 3, using $v = f^*_\phi(\eta)$. Property 2 follows from taking derivatives on both sides of (II.67) (The invertability of Ψ follows from the invertability of $[f^*_\phi]^{-1}$ and (II.66).)

$$J''(\eta) = \frac{1}{\Psi''\{[\Psi']^{-1}(\eta - 1/2)\}},$$

differentiating twice both sides of (II.65)

$$\phi''(v) = \Psi''(v)$$

and using (II.67). Property 5 follows from using $\eta = [f^*_\phi]^{-1}(v)$ and Property 3 in Property 1. Property 6 follows from (II.53) and Property 5 ($[J']^{-1}(v) = 1 + \phi'(v)$).

We next consider the properties of $\phi(v)$. Property 1 follows from $J''(\eta) = 1/\phi''[f^*_\phi(\eta)]$ and the strict convexity of $J(\eta)$. Property 2 follows from Theorem 9. Properties 3 and 5 follow from differentiating once or twice, respectively, the two sides of Property 2. Property 4 follows by setting $v = 0$ in Property 3.

With respect to the properties of f^*_ϕ, Properties 2 and 5 follow from Lemma 1, Property 3 from Property 2 with $\eta = 1/2$, and Property 6 from Property 5 with $v = 0$. From the strict convexity of $J(\eta)$ and $f^*_\phi(\eta) = J'(\eta)$ it follows that $[f^*_\phi]'(\eta) > 0, \forall \eta$. Hence, $f^*_\phi(\eta)$ is monotonically increasing and invertible. Since

$$\left([f^*_\phi]^{-1}\right)'(v) = \frac{1}{[f^*_\phi]'(\eta)\big|_{\eta=[f^*_\phi]^{-1}(v)}},$$

it follows that $\left([f^*_\phi]^{-1}\right)'(v) > 0$ for all $v \in (-\infty, \infty)$, and $[f^*_\phi]^{-1}(v)$ is monotonically increasing. Finally, from the monotonicity of $f^*_\phi(\eta)$ and Property 3 it follows that $f^*_\phi(\eta)$ satisfies (II.7) and the risk is Bayes consistent.

■

We have already seen two canonical risks, derived from either $J(\eta)$ or $f^*_\phi(\eta)$, whose loss functions and link functions are very similar to those of logistic regression. The theorem suggests that this will be the case for many canonical risks. Note that, like the logistic loss, all canonical losses are concave, have a derivative of $-1/2$ at the origin, derivative symmetry of $\phi'(v) + \phi'(-v) = -1$, and odd-symmetric curvature. Also, all optimal canonical links share the symmetry of the sigmoidal link of logistic regression. A natural question is then whether *all* canonical losses and links are really just like those of logistic regression. We next investigate this question.

II.C.5 Canonical loss behavior

We start by considering a family of losses inspired by (II.66), namely the canonical losses derived from inverse link functions of the form

$$[f^*_\phi]^{-1}(v; a) \quad = \quad \frac{1}{2} + \frac{1}{a}\arctan v. \qquad \text{(II.68)}$$

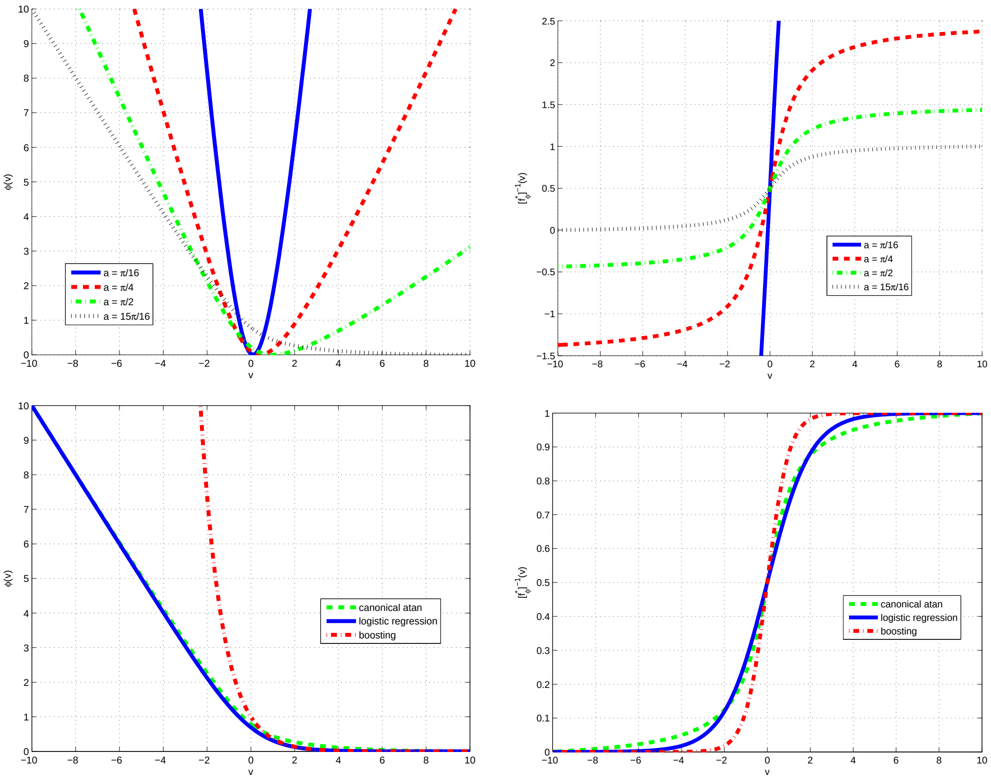


Figure II.5 Loss (left) and link functions (right) of canonical boosting, boosting, and logistic regression.

This family is parametrized by $a \in (0, \pi)$ and can be shown to have canonical risk components

$$f^*_\phi(\eta; a) = \tan\left[\left(\eta - \frac{1}{2}\right) a\right] \tag{II.69}$$

$$J(\eta; a) = -\frac{1}{a}\log\left(\frac{\cos[(\eta - \frac{1}{2})a]}{\cos\frac{a}{2}}\right) \tag{II.70}$$

and loss

$$\phi(v; a) = -\frac{1}{a}\log\left[\cos\left(\frac{a}{2}\right)\sqrt{1+v^2}\right] - \left(\frac{1}{2} - \frac{1}{a}\arctan v\right) v. \tag{II.71}$$

Figure II.5 presents plots of the link and loss functions for various values of the parameter a. The behavior of these functions changes dramatically as a goes from $\pi/16$ to $15\pi/16$. For large a, the loss and link functions are indeed very similar to those of logistic regression. This is made clear in the bottom plots, where the functions obtained when $a = 15\pi/16$ are compared to those of boosting and logistic regression. Note, once again, the similarity with the latter. In particular, as a increases, the loss function exhibits the traits typical of the classification losses of Figure II.1: it is margin enforcing and zero for most positive values of its argument. Similarly, the optimal link function also becomes one typical for classification, exhibiting a sigmoidal shape of value 1/2 at the origin, and saturating at 0 and 1. A very different behavior emerges for small a, where the loss function becomes approximately quadratic and the inverse link function is approximately linear. These characteristics are very similar to those of canonical least squares, and more suitable for regression problems. For intermediate values of the parameter a the loss and link are somewhere between functions suitable for classification and regression.

This behavior is not restricted to the family of losses of (II.71). For example, it also holds for the canonical risks derived from the following family of maximum reward functions

$$J(\eta; a) \;=\; \sec\left[a\left(\frac{1}{2}-\eta\right)\right] - \sec(\frac{-a}{2}) \tag{II.72}$$

parameterized by $a \in (0, \pi)$. The resulting canonical links and losses are

$$\begin{aligned}
J(\eta; a) &= \sec\left[a\left(\frac{1}{2}-\eta\right)\right] - \sec(\frac{-a}{2}) && \text{(II.73)}\\
[f_\phi^*]^{-1}(v; a) &= \frac{1}{2} - \frac{1}{a}\sin^{-1}\left(\frac{a-\sqrt{a^2+4v^2}}{2v}\right) && \text{(II.74)}\\
\phi(v; a) &= -\sec\left(\sin^{-1}\left(\frac{a-\sqrt{a^2+4v^2}}{2v}\right)\right) + \sec(\frac{-a}{2}) && \text{(II.75)}\\
&\quad -\left(\frac{1}{2}+\frac{1}{a}\sin^{-1}\left(\frac{a-\sqrt{a^2+4v^2}}{2v}\right)\right)v
\end{aligned}$$

Figure II.6 presents the plots of the loss and link for this family, as a varies between 0 and π. Note how the loss eventually increases for large positive margins. These maximal rewards, canonical link and loss functions are summarized by Table II.4. The bottom portion of the table reports to risks derived from an inverse link function, and the top portion to risks derived from a maximal reward.

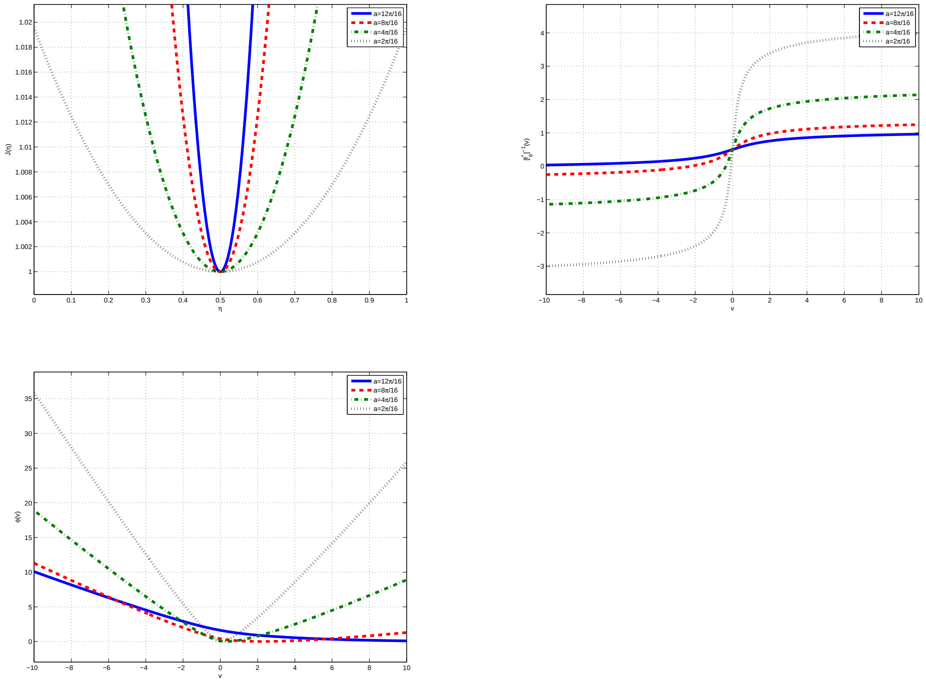


Figure II.6 $J(\eta)$ (left), link functions (right) and canonical secant loss functions for varying values of $a \in (0\ \pi)$.

Regression losses that have this behavior are defined as

Definition 3. *A loss function $\phi(v)$ is denoted as a regression loss if it has the following two properties*

1. $\lim_{v\to\infty} \phi(v) = \infty$

2. $\lim_{v\to\infty} \phi'(v) > 0$

The following theorem specializes Theorem 12 to the case of canonical regression losses.

Theorem 13. *Let $C_\phi(\eta, f)$ be a canonical risk of loss $\phi(v)$, with optimal link $f^*_\phi(\eta)$, and maximal reward $J(\eta) = -C^*_\phi(\eta)$. If $\phi(v)$ is a regression loss, the properties of Theorem 12 are complemented as follows. For the link $f^*_\phi(\eta)$*

1. $\lim_{v\to\infty} [f^*_\phi]^{-1}(v) > 1$

2. $\lim_{v\to-\infty}[f^*_\phi]^{-1}(v) < 0.$

For the loss

1. $\lim_{v\to-\infty}\phi'(v) < -1$

2. $\phi(v)$ *is decreasing up to the point* $v = f^*_\phi(1)$ *and increasing from then onwards.*

3. $\phi(v)$ *is margin enforcing.*

Proof. We start with the properties of $[f^*_\phi]^{-1}$. From Theorem 12

$$[f^*_\phi]^{-1}(v) = 1 + \phi'(v).$$

Hence, Property 1 follows from the second property of regression losses. Combining this with the symmetry

$$[f^*_\phi]^{-1}(v) = 1 - [f^*_\phi]^{-1}(-v)$$

leads to Property 2. We next consider the properties of the loss $\phi(v)$. Property 1 follows from $\phi(v) + \phi(-v) = -1$ and the first property of regression losses. Property 2 follows from the facts that

$$\phi'(v) \quad = \quad [f^*_\phi]^{-1}(v) - 1,$$

$[f^*_\phi]^{-1}(v)$ is a monotonically increasing function, and the limit properties of $[f^*_\phi]^{-1}(v)$ above such that $\phi'(v) = 0$ when $1 = [f^*_\phi]^{-1}(v)$. Property 3 ($\phi(0) > 0$) is proven in the most general case in Theorem 21. ■

Figure II.4 shows the relationship between regression losses and their optimal inverse link functions.

II.C.6 Classification losses

We start by defining classification losses.

Table II.5 Relation between $J(\eta)$, $J'(\eta)$ and $f^*_\phi(\eta)$

Name	$J(\eta)$	$J'(\eta) = f^*_\phi(\eta)$
LS	$-4\eta + 4\eta^2$	$-4 + 8\eta$
Exp	$-2\sqrt{\eta(1-\eta)}$	$\frac{-(1-2\eta)}{\sqrt{\eta(1-\eta)}}$
Log	$\eta\log(\eta) + (1-\eta)\log(1-\eta)$	$\log(\frac{\eta}{1-\eta})$
Sec	$\sec(a(\frac{1}{2}-\eta)) - \sec(\frac{-a}{2})$	$-a\sec(a(\frac{1}{2}-\eta))\tan(a(\frac{1}{2}-\eta))$
Cosh	$\cosh(a(\frac{1}{2}-\eta))$	$-a\sinh(a(\frac{1}{2}-\eta))$
Log-Cos	$-\frac{1}{a}\log\left(\frac{\cos[(\eta-\frac{1}{2})a]}{\cos\frac{a}{2}}\right)$	$\tan(a(\eta-\frac{1}{2}))$

Table II.6 Canonical link functions and their range.

Name	$f^{-1}(v) = \eta$	Range
LS	$\frac{v+4}{8}$	$[-\infty \quad +\infty]$
Exp	$(\frac{1}{2} + \frac{1}{2}\frac{v}{\sqrt{4+v^2}})$	$[0 \quad 1]$
Log	$\frac{e^v}{1+e^v}$	$[0 \quad 1]$
Sec	$\frac{1}{2} - \frac{1}{a}\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v})$	$[\frac{1}{2} - \frac{\pi}{2a} \quad \frac{1}{2} + \frac{\pi}{2a}]$
Cosh	$\frac{1}{2} - \frac{1}{a}\sinh^{-1}(\frac{-v}{a})$	$[-\infty \quad +\infty]$
Log-Cos	$\frac{1}{a}\tan^{-1}(v) + \frac{1}{2}$	$[\frac{1}{2} - \frac{\pi}{2a} \quad \frac{1}{2} + \frac{\pi}{2a}]$

Table II.7 $J(\eta)$ functions and their valid domains.

Name	$J(\eta)$	Domain
LS	$-4\eta + 4\eta^2$	$[-\infty \quad +\infty]$
Exp	$-2\sqrt{(\eta(1-\eta))}$	$[0 \quad 1]$
Log	$\eta\log(\eta) + (1-\eta)\log(1-\eta)$	$(0 \quad 1)$
Sec	$\sec(a(\frac{1}{2}-\eta)) - \sec(\frac{-a}{2})$	$[\frac{1}{2} - \frac{\pi}{2a} \quad \frac{1}{2} + \frac{\pi}{2a}]$
Cosh	$\cosh(a(\frac{1}{2}-\eta))$	$[-\infty \quad +\infty]$
Log-Cos	$-\frac{1}{a}\log\left(\frac{\cos[(\eta-\frac{1}{2})a]}{\cos\frac{a}{2}}\right)$	$[\frac{1}{2} - \frac{\pi}{2a} \quad \frac{1}{2} + \frac{\pi}{2a}]$

Definition 4. *A loss function $\phi(v)$ is denoted as a classification loss if it has the following two properties*

1. *vanishing loss:* $\lim_{v\to\infty}\phi(v) = 0$

2. *vanishing derivative:* $\lim_{v\to\infty}\phi'(v) = 0$

The following theorem specializes Theorem 12 to the case of canonical classification losses.

Theorem 14. *Let $C_\phi(\eta, f)$ be a canonical risk of loss $\phi(v)$, with optimal link $f^*_\phi(\eta)$, and maximal reward $J(\eta) = -C^*_\phi(\eta)$. If $\phi(v)$ is a classification loss, the properties of Theorem 12 are complemented as follows. For the link $f^*_\phi(\eta)$*

1. $\lim_{v\to\infty}[f^*_\phi]^{-1}(v) = 1$

2. $\lim_{v\to-\infty}[f^*_\phi]^{-1}(v) = 0.$

For the loss

1. $\lim_{v\to-\infty}\phi'(v) = -1$

2. *$\phi(v)$ is monotonically decreasing;*

3. *$\phi(v)$ is margin enforcing.*

For the maximal reward

1. $J(1) = J(0) = -\lim_{v\to\infty} v[f^*_\phi]^{-1}(-v).$

Proof. We start with the properties of $[f^*_\phi]^{-1}$. From Theorem 12

$$[f^*_\phi]^{-1}(v) = 1 + \phi'(v).$$

Hence, Property 1 follows from the vanishing derivative property of the classification loss. Combining this with the symmetry

$$[f^*_\phi]^{-1}(v) = 1 - [f^*_\phi]^{-1}(-v)$$

leads to Property 2. We next consider the properties of the loss $\phi(v)$. Property 1 follows from $\phi(v) + \phi(-v) = -1$ and the vanishing derivative property. Property 2 follows from the facts that

$$\phi'(v) \;\; = \;\; [f_\phi^*]^{-1}(v) - 1,$$

$[f_\phi^*]^{-1}(v)$ is a monotonically increasing function, and the limit properties of $[f_\phi^*]^{-1}(v)$ above. Property 3 ($\phi(0) > 0$) is proven in the most general case in Theorem 21. $\phi(v)$ is monotonically decreasing, it has limit 0 as $v \to \infty$, and derivative $\phi'(0) = -\frac{1}{2}$. Finally, taking the limit of

$$\phi(v) = -J\{[f_\phi^*]^{-1}(v)\} - (1 - [f_\phi^*]^{-1}(v))v$$

and using $J(\eta) = J(1 - \eta)$, it follows that

$$\begin{aligned} J(0) = J(1) &= -\lim_{v \to \infty} (1 - [f_\phi^*]^{-1}(v))v \\ &= -\lim_{v \to \infty} v[f_\phi^*]^{-1}(-v) \end{aligned}$$

■

Figure II.7 shows the relationship between classification losses and their optimal inverse link functions.

II.D Non canonical loss functions

In the non canonical case, it is tempting to assume that any link function can be paired with any valid $J(\eta)$ and used to derive a novel Bayes consistent loss function. This is only true if care is taken to make sure that the range of the used link function is compatible with the domain of the used $J(\eta)$. We have presented the range of each link function in Table II.6 and we also include the domain of each $J(\eta)$ in Table II.7. (Note that the domain of $\sec(a(\frac{1}{2} - \eta))$ and $-\frac{1}{a}\log\left(\frac{\cos[(\eta - \frac{1}{2})a]}{\cos\frac{a}{2}}\right)$ are purposely restricted to ensure convexity and symmetry of $J(\eta)$.)

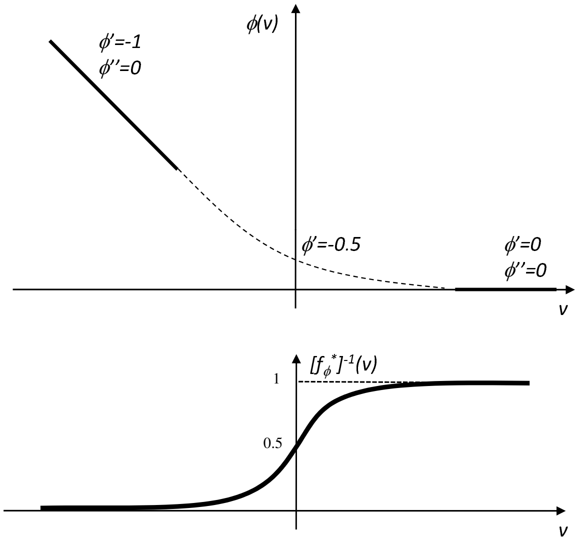


Figure II.7 Classification losses and their optimal inverse link functions.

For example, the pairing of the LS canonical link function $f^{-1}(v) = \frac{v+4}{8}$ and $J(\eta) = \sec(a(\frac{1}{2} - \eta))$ is not correct. The range of $\eta = f^{-1}(v) = \frac{v+4}{8}$ is $\eta \in [-\infty \quad +\infty]$ which is incompatible with the domain of $J(f^{-1}(v)) = J(\eta) = \sec(a(\frac{1}{2} - \eta))$ which is $\eta \in [\frac{1}{2} - \frac{\pi}{2a} \quad \frac{1}{2} + \frac{\pi}{2a}]$.

Theorem 15. *Let $f^{-1}(v)$ be a link function as defined in Theorem-7 with Range $R_{f^{-1}}$, and $J(\eta)$ a convex and symmetric function as defined in Theorem-7 with Domain D_J. Equation-(II.48) can be used to derive a Bayes consistent loss function $\phi(v)$ if $R_{f^{-1}} \subseteq D_J$.*

In summary in order to have a valid pairing of functions the range of the link function must be a subset of the domain of $J(\eta)$ such that the convexity and symmetry of $J(\eta)$ is preserved.

Table II.8 presents a list of 24 valid pairings of $J(\eta)$ (from Table II.7) and link functions (from Table II.6) and the resulting loss functions. Among these loss functions only three were previously known (marked by $*$) and the rest are novel. In the following sections we will further explore in detail some of these novel Bayes consistent loss functions and their properties.

II.E Non convex Bayes consistent loss functions

It is interesting to note that many of the novel Bayes consistent loss functions derived from Theorem 7 in Table II.8 are not convex, violating what has been a hallmark of the ϕ functions used in the literature. The convexity of ϕ is, however, not important. What matters is that the risk of (II.10) have some notion of convexity, given that the risk is being minimized and not the loss. This is made clear by the probability elicitation view and (II.28) which states that the expected reward function must have a global maximum at $\hat{\eta} = \eta$. This in turn does not even require that $I(\eta, \hat{\eta}) = -C(\eta, f^*(\hat{\eta}))$ be a concave function of $\hat{\eta}$. In fact we show that the risk of (II.10) is not necessarily convex on $\hat{\eta}$ but provably quasi convex on $\hat{\eta}$.

Theorem 16. *The conditional risk of (II.10) is quasi convex with respect to $\hat{\eta}$ irrespective of the convexity of the loss ϕ.*

Proof. Noting that $I(\eta, \hat{\eta}) = -C(\eta, f^*(\hat{\eta}))$ we can write

$$C(\eta, f^*(\hat{\eta})) = -I(\eta, \hat{\eta}) = -J(\hat{\eta}) + J'(\hat{\eta})(\hat{\eta} - \eta) \tag{II.76}$$

taking the first derivative with respect to $\hat{\eta}$

$$C'(\eta, f^*(\hat{\eta})) = J''(\hat{\eta})(\hat{\eta} - \eta) \tag{II.77}$$

taking the second derivative with respect to $\hat{\eta}$

$$C''(\eta, f^*(\hat{\eta})) = J'''(\hat{\eta})(\hat{\eta} - \eta) + J''(\hat{\eta}). \tag{II.78}$$

Given the fact that $J''(\hat{\eta}) > 0$, the first derivative (II.77) changes sign and is zero only at $\hat{\eta} = \eta$ thus proving that the risk is quasi convex on $\hat{\eta}$. On the other hand, the second derivative (II.78) is not necessarily positive for all values of η and $\hat{\eta}$ meaning that the risk is not necessarily convex over the probability estimates $\hat{\eta}$. In fact this is made clear by (II.28) which requires that $\hat{\eta} = \eta$ only be the global maximum of the reward $I(\eta, \hat{\eta})$, concavity is not required.

∎

Using the probability elicitation view and Savage's main theorem, we have shown that the conditional risk of a loss function ϕ will be quasi convex with respect to $\hat{\eta}$ irrespective of the convexity of ϕ itself. But we have not shown that the transformation of variable performed by the link function will preserve the quasi convexity of the conditional risk such that it is still at least quasi convex with respect to f. In general, a transformation of variable does not preserve convexity or quasi convexity. Yet, it is important to have some notion of what happens to the conditional risk in terms of f since in practice many learning algorithms such as boosting and SVM's work directly in the f space and proceed to minimize the risk over the space of f decision functions.

Here we show that the conditional risk is quasi convex with respect to f (irrespective of the convexity of ϕ) and thus has a unique minimum at f^*.

Theorem 17. *The conditional risk of (II.10) is quasi-convex with respect to f irrespective of the convexity of the loss ϕ.*

Proof. We take the derivative of $C(\eta, f(\hat{\eta}))$ with respect to f and show that it changes sign and is zero only at $f = f^*$.

$$C = \eta\phi(f(\hat{\eta})) + (1-\eta)\phi(-f(\hat{\eta})) = \eta\phi(v) + (1-\eta)\phi(-v) \tag{II.79}$$

where $v = f(\hat{\eta})$. taking the derivative of $C(\eta, f)$ with respect to f we have

$$\begin{aligned}
&\frac{\partial C}{\partial v} = \eta\frac{\partial \phi(v)}{\partial v} + (1-\eta)\frac{\partial \phi(-v)}{\partial v} = \\
&\eta\frac{\partial[-J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v))]}{\partial v} \\
&+(1-\eta)\frac{\partial[-J(f^{-1}(v)) + f^{-1}(v)J'(f^{-1}(v))]}{\partial v} = \\
&-\eta\frac{\partial J(f^{-1}(v))}{\partial v} - \eta\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&+\eta\frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) + \eta f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&-\frac{\partial J(f^{-1}(v))}{\partial v} + \frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) + f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&\eta\frac{\partial J(f^{-1}(v))}{\partial v} - \eta\frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) - \eta f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} = \\
&(f^{-1}(v)-\eta)\frac{\partial J'(f^{-1}(v))}{\partial v} + \frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) - \frac{\partial J(f^{-1}(v))}{\partial v} = \\
&(f^{-1}(v)-\eta)\frac{\partial J'(f^{-1}(v))}{\partial v} + \frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) \\
&-\frac{\partial J(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v} = \\
&(f^{-1}(v)-\eta)\frac{\partial J'(f^{-1}(v))}{\partial v} + \frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) \\
&-\frac{\partial f^{-1}(v)}{\partial v}J'(f^{-1}(v)) = \\
&(f^{-1}(v)-\eta)\frac{\partial J'(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v} = \\
&(f^{-1}(v)-\eta)J''(f^{-1}(v))\frac{\partial f^{-1}(v)}{\partial v}
\end{aligned}$$

Since $J''(f^{-1}(v)) > 0$ and $\frac{\partial f^{-1}(v)}{\partial v} > 0$ the above changes sign and is zero only at $f^{-1}(v) = \eta$ and hence $C(\eta, f)$ is quasi convex over f.

■

We have shown above that the conditional risk is quasi convex with respect to f. We also need to explore the properties of the risk and empirical risk with respect to f, because in practice we are trying to minimize the empirical risk. We need to show that the empirical risk does not have local minimum which would prevent the effective minimization of the empirical risk. If we show that the empirical risk does not have any local minimum, then in practice we can do functional gradient descent (such as boosting) to minimize the empirical risk and have a working classification algorithm.

Before we proceed any further we first prove that $\phi(v)$ is itself quasi convex. Meaning that all Bayes consistent loss functions found from our approach are quasi convex functions.

Theorem 18. *The Bayes consistent loss functions derived from* $\phi(v) = -J(f^{-1}(v)) - (1 - f^{-1}(v))J'(f^{-1}(v))$ *are quasi convex with respect to* f.

Proof.

$$\begin{aligned}
\frac{\partial\phi(v)}{\partial v} &= \frac{\partial[-J(f^{-1}(v)) - (1 - f^{-1}(v))J'(f^{-1}(v))]}{\partial v} \\
&= -\frac{\partial J(f^{-1}(v))}{\partial v} - \frac{\partial J'(f^{-1}(v))}{\partial v} \\
&+ \frac{\partial f^{-1}(v)}{\partial v} J'(f^{-1}(v)) + f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&= -\frac{\partial f^{-1}(v)}{\partial v} J'(f^{-1}(v)) - \frac{\partial J'(f^{-1}(v))}{\partial v} \\
&+ \frac{\partial f^{-1}(v)}{\partial v} J'(f^{-1}(v)) + f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&= -\frac{\partial J'(f^{-1}(v))}{\partial v} + f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial v} \\
&= \frac{\partial J'(f^{-1}(v))}{\partial v}(f^{-1}(v) - 1) \\
&= \frac{\partial J'(f^{-1}(v))}{\partial (f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}(f^{-1}(v) - 1) \\
&= J''(f^{-1}(v))\frac{\partial f^{-1}(v)}{\partial v}(f^{-1}(v) - 1)
\end{aligned}$$

Again since $J''(f^{-1}(v)) > 0$ and $\frac{\partial f^{-1}(v)}{\partial v} > 0$ the above changes sign and is zero only at $f^{-1}(v) = 1$ and hence $\phi(v)$ is quasi convex. ∎

The above theorem can in fact be generalized to all Bayes consistent loss functions using the theorem below.

Theorem 19. *Any Bayes consistent loss function $\phi(v)$ (in the form of (II.9)) is quasi convex with respect to f.*

Proof. From (II.19) we can write

$$\phi(v) = C^*\{[f^*]^{-1}(v)\} + (1 - [f^*]^{-1}(v))[C^*]'\{[f^*]^{-1}(v)\} \tag{II.80}$$

taking the derivative

$$\phi'(v) = -[C^*]''\{[f^*]^{-1}(v)\}\frac{\partial[f^*]^{-1}(v)}{\partial v}([f^*]^{-1}(v) - 1) \tag{II.81}$$

Since $[C^*]''\{[f^*]^{-1}(v)\} < 0$ (Theorem-2) and $\frac{\partial[f^*]^{-1}(v)}{\partial v} > 0$ ($[f^*]^{-1}(v)$ is monotonically increasing given that it is invertible and Bayes consistent) the above changes sign and is zero only at $[f^*]^{-1}(v) = 1$ and hence $\phi(v)$ is quasi convex. ■

Th above theorems allow us to categorize Bayes consistent loss functions into four shape varieties.

Theorem 20. *Bayes consistent loss functions are of four basic shape varieties. (1) Convex and nondecreasing, (2) quasi convex and nondecreasing, (3) convex and nondecreasing up to a point and then non increasing from then on (4) quasi convex and nondecreasing up to a point and then non increasing from then on.*

Proof. From Theorem 18 we know that

$$\frac{\partial\phi(v)}{\partial v} = -[C^*]''\{[f^*]^{-1}(v)\}\frac{\partial[f^*]^{-1}(v)}{\partial v}([f^*]^{-1}(v) - 1) \tag{II.82}$$

Again since $[C^*]''\{[f^*]^{-1}(v)\} < 0$ and $\frac{\partial[f^*]^{-1}(v)}{\partial v} > 0$ the above changes sign and is zero only at $f^{-1}(v) = 1$. This means that if the link is such that $f^{-1}(v) < 1$ for all v (as in the case of the logit link), then all loss functions derived from this link will be nondecreasing (such as the exp loss), if the link is such that it can be $f^{-1}(v) > 1$ for some v, then we get losses that are nondecreasing up to a point and then non increasing from then on (such as the LS loss). So if a link $f^{-1}(v) < 1$ for all v is used to derive a loss function, the resulting loss will be of variety (1) or (2) i.e. a nondecreasing loss. for example the exp and log loss functions that use the logistic link fall into this category. If a link $f^{-1}(v) > 1$ for some v, is used to derive a loss function, the resulting loss will be of variety (3) or (4) i.e. nondecreasing up to a point and then non increasing from then on. For example the least squares loss falls into this category. So the choice of $f^{-1}(v)$ can be used to design the shape of the loss, defining whether it is of variety (1) and (2) or of (3) and (4). designing a loss to be convex or quasi convex, i.e designing it to be of variety (1) as apposed to (2) is more complicated. It requires that we compute the second derivative of the loss. defining the first derivative of the loss to be $\phi'(v)$, the second derivative can be written as

$$\frac{\partial \phi'(v)}{\partial v} \tag{II.83}$$

$$= \frac{\partial [J''(f^{-1}(v))\frac{\partial f^{-1}(v)}{\partial v}(f^{-1}(v)-1)]}{\partial v} \tag{II.84}$$

$$= J'''(f^{-1}(v))(\frac{\partial f^{-1}(v)}{\partial v})^2 f^{-1}(v) + J''(f^{-1}(v))(\frac{\partial(\frac{\partial f^{-1}(v)}{\partial v})}{\partial v})f^{-1}(v) \tag{II.85}$$

$$+J''(f^{-1}(v))(\frac{\partial f^{-1}(v)}{\partial v})^2 - J'''(f^{-1}(v))(\frac{\partial f^{-1}(v)}{\partial v})^2 \tag{II.86}$$

$$-J''(f^{-1}(v))(\frac{\partial(\frac{\partial f^{-1}(v)}{\partial v})}{\partial v}) \tag{II.87}$$

Ensuring convexity of the loss would require that the above equation be positive. This cannot be simply characterized and depends on the third derivative of J as well as the range of $f^{-1}(v)$. In summary, by simply checking if $f^{-1}(v) < 1$ for all v or not, we can design a loss that is of variety (1) and (2) or (3) and (4). Choosing between (1) and (2) (or (3) and (4)) is not as easy and requires that the second derivative of the loss (above formula) be derived and checked. ∎

Theorem 21. *All Bayes consistent loss functions derived from $\phi(v) = -J(f^{-1}(v)) - (1 - f^{-1}(v))J'(f^{-1}(v))$ are margin enforcing (i.e. have a minimum at $v > 0$).*

Proof. From Theorem 18 we know that $\phi(v)$ is quasi-convex and so has a unique global minimum. From Theorems 20 we also know that if the loss is of varieties (1) or (2), then the loss has a minimum at $v = \infty > 0$ thus proving the theorem. If the loss is of types (3) or (4) then from the proof of theorem 18 we know that the minimum is at a v such that $f^{-1}(v) = 1$. Also since $f^{-1}(v)$ is invertible and

$f^{-1}(v) > 0.5$ for $v > 0$, then v must be greater than zero ($v > 0$) when $f^{-1}(v) = 1$ at the minimum. ■

The above theorem is not surprising given that we would expect a meaningful loss function to assign minimum loss to data points that have been classified correctly meaning that minimum loss should be at $v > 0$.

Now we prove that the empirical risk does not have local minimum and can be effectively minimized by a functional gradient descent algorithm.

Theorem 22. *The risk and empirical risk function*

$$\begin{aligned} R(g(x_i)) &= \sum_{i=1}^{n} \phi(y_i g(x_i)) \\ &= \sum_{i=1}^{n} -J(f^{-1}(y_i g(x_i))) - (1 - f^{-1}(y_i g(x_i)))J'(f^{-1}(y_i g(x_i))) \end{aligned}$$

has a single point of minimum in functional space (no local minimum) and can be minimized by functional gradient descent.

Proof. The first order variation of the empirical risk at point $g(x_i)$ along the direction $h(x_i)$ (also known as the Gateaux variation [91]) is

$$\begin{aligned} &\frac{d}{d\epsilon} R(g(x_i) + \epsilon h(x_i))|_{\epsilon=0} = \\ &\frac{d}{d\epsilon}\Bigg[\sum_{i=1}^{n} -J(f^{-1}(y_i\{g(x_i) + \epsilon h(x_i)\})) \\ &-(1 - f^{-1}(y_i\{g(x_i) + \epsilon h(x_i)\}))J'(f^{-1}(y_i\{g(x_i) + \epsilon h(x_i)\}))\Bigg]_{\epsilon=0} = \\ &\Bigg[\sum_{i=1}^{n} -\frac{\partial J(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon} - \frac{\partial J'(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon} \\ &+\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}J'(f^{-1}(v)) + f^{-1}(v)\frac{\partial J'(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}\Bigg]_{\epsilon=0} = \\ &\Bigg[\sum_{i=1}^{n} J''(f^{-1}(v))\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}(f^{-1}(v) - 1)\Bigg]_{\epsilon=0} = \\ &\sum_{i=1}^{n} J''(f^{-1}(y_i g(x_i))) \cdot \frac{\partial f^{-1}(y_i g(x_i))}{\partial(y_i g(x_i))} \cdot (y_i h(x_i)) \cdot (f^{-1}(y_i g(x_i)) - 1) \end{aligned}$$

where we have set $v = y_i\{g(x_i) + \epsilon h(x_i))\}$.

Here we consider two cases. The first case is when we have a nondecreasing loss function (loss variety 1 or 2). In this case $(f^{-1}(y_i g(x_i)) - 1) < 0$. We also know that $J''(f^{-1}(y_i g(x_i))) > 0$ given that J is strictly convex and $\frac{\partial f^{-1}(v)}{\partial(v)} > 0$ for all $-\infty < v < \infty$. This means that for any y_i, x_i and $g(x_i)$ a direction $h(x_i) \neq 0$ exists (for example by setting $h(x_i) = y_i$) such that the above variation is negative. This is the direction of descent and by moving in that direction the empirical risk can be further reduced. In other words, since a direction of descent always exists for any choice of y_i, x_i and $g(x_i)$, the empirical risk does not get stuck in a local minimum and can be minimized by gradient descent.

The second case we consider is when the loss is nondecreasing up to a single point and then non increasing (loss varieties 3 and 4). In this case the single point $g^*(x_i)$ can be found such that $f^{-1}(y_i g^*(x_i)) = 1$ for all y_i and x_i (note that this follows directly from the fact that $f^{-1}(v)$ is monotonic, otherwise multiple $g^*(x_i)$ could exists that would make $f^{-1}(y_i g^*(x_i)) = 1$). At this single point the above variation will be equal to zero for any choice of $h(x_i)$. This means that the empirical risk has no variation at the point $g^*(x_i)$ and so $g^*(x_i)$ is either an extremum or a saddle point. Next we take the second order variation of the empirical risk at the point $g^*(x_i)$ and show (see below) that the second order variation is positive for any choice of $h(x_i)$ meaning that $g^*(x_i)$ is a minimum. Since $g^*(x_i)$ is the only point that makes the first order variation zero for any choice of $h(x_i) \neq 0$, and the second order variation is always positive at that point, then $g^*(x_i)$ is the only minimum. For any point other than $g^*(x_i)$, the first order variation will not always be zero (for any choice of $h(x_i) \neq 0$) and a direction $h(x_i)$ can be found such that the first order variation is negative. This in turn means that a direction of descent exists for any point in functional space other than the minimum at $g^*(x_i)$.

Note that we make no claims of convexity or quasi convexity for the empirical risk, in fact they are not generally convex or quasi convex. What we have shown is that the minimum of the empirical risk can always be found by gradient descent and that we will not get stuck in local minimum or saddle points. Nevertheless, the proof was a direct result of the loss being quasi-convex. If the loss where to have more than one minimum and were not quasi convex, then the above argument could not be made and the empirical risk would also have multiple local minimum where a gradient descent algorithm could get stuck.

The second order variation at the point $g(x_i)$ in the direction of $h(x_i)$ is

$$
\begin{aligned}
&\frac{d^2}{d\epsilon^2}R(g(x_i)+\epsilon h(x_i))|_{\epsilon=0} = \\
&\frac{d^2}{d\epsilon^2}\Bigg[\sum_{i=1}^{n} -J(f^{-1}(y_i\{g(x_i)+\epsilon h(x_i))\})) \\
&-(1-f^{-1}(y_i\{g(x_i)+\epsilon h(x_i))\}))J'(f^{-1}(y_i\{g(x_i)+\epsilon h(x_i))\}))\Bigg]_{\epsilon=0} = \\
&\Bigg[\sum_{i=1}^{n}[\frac{\partial J''(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}]\frac{\partial f^{-1}(v)}{\partial v}(y_ih(x_i))f^{-1}(v) \\
&+J''(f^{-1}(v))[\frac{\partial\frac{\partial f^{-1}(v)}{\partial v}}{\partial v}\frac{\partial v}{\partial \epsilon}](y_ih(x_i))f^{-1}(v) \\
&+J''(f^{-1}(v))\frac{\partial f^{-1}(v)}{\partial v}[\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}](y_ih(x_i)) \\
&-[\frac{\partial J''(f^{-1}(v))}{\partial(f^{-1}(v))}\frac{\partial f^{-1}(v)}{\partial v}\frac{\partial v}{\partial \epsilon}]\frac{\partial f^{-1}(v)}{\partial v}(y_ih(x_i)) \\
&-J''(f^{-1}(v))[\frac{\partial\frac{\partial f^{-1}(v)}{\partial v}}{\partial v}\frac{\partial v}{\partial \epsilon}](y_ih(x_i))\Bigg]_{\epsilon=0} = \\
&\sum_{i=1}^{n} J'''(f^{-1}(y_ig(x_i)))(\frac{\partial f^{-1}(y_ig(x_i))}{\partial(y_ig(x_i))})^2(y_ih(x_i))^2f^{-1}(y_ig(x_i)) \\
&+J''(f^{-1}(y_ig(x_i)))(\frac{\partial f^{-1}(y_ig(x_i))}{\partial(y_ig(x_i))})(y_ih(x_i))^2f^{-1}(y_ig(x_i)) \\
&+J''(f^{-1}(y_ig(x_i)))(\frac{\partial f^{-1}(y_ig(x_i))}{\partial(y_ig(x_i))})^2(y_ih(x_i))^2 \\
&-J'''(f^{-1}(y_ig(x_i)))(\frac{\partial f^{-1}(y_ig(x_i))}{\partial(y_ig(x_i))})^2(y_ih(x_i))^2 \\
&-J''(f^{-1}(y_ig(x_i)))(\frac{\partial f^{-1}(y_ig(x_i))}{\partial(y_ig(x_i))})(y_ih(x_i))^2
\end{aligned}
\tag{II.88}
$$

where we have again used $v = y_i\{g(x_i)+\epsilon h(x_i))\}$. The second order variation at point $g^*(x_i)$ where $f^{-1}(y_i g^*(x_i)) = 1$ is

$$\begin{aligned}
&\sum_{i=1}^{n} J'''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})^2(y_i h(x_i))^2 \\
&+J''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})(y_i h(x_i))^2 \\
&+J''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})^2(y_i h(x_i))^2 \\
&-J'''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})^2(y_i h(x_i))^2 \\
&-J''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})(y_i h(x_i))^2 = \\
&= \sum_{i=1}^{n} J''(f^{-1}(y_i g^*(x_i)))(\frac{\partial f^{-1}(y_i g^*(x_i))}{\partial(y_i g^*(x_i))})^2(y_i h(x_i))^2 > 0
\end{aligned} \tag{II.89}$$

Given that J is strictly convex then $J'' > 0$ and the above is always positive for any $h(x_i) \neq 0$.

A very similar proof can be made for the risk itself by simply replacing summations with the integral. ■

In summary, the above series of proofs show that (1) Bayes consistent loss functions need not be convex but are restricted to being quasi convex, (2) Although the loss is not necessarily convex, but the conditional risk is quasi convex. (3) the risk and empirical risk have a unique minimum that can be found in practice with functional gradient descent algorithms.

II.F Bounded Bayes consistent loss functions

In Section-II.E we showed that there are only four varieties of Bayes consistent loss functions. These varieties were based on the convexity or quasi-convexity properties and monotonicity of the loss function. Unfortunately, the four varieties do not give much information when one is trying to design loss functions with certain shapes and properties. For example no information is given

Table II.8 Table of Bayes consistent loss functions.(*) Previously known loss functions.

#	Type	$J(\eta)$	$f^{-1}(v)$	$\phi(v)$
1$*$	I	LS	LS	$\frac{1}{16}(v-4)^2$
2	VI	LS	Exp	$1-\frac{2v}{\sqrt{4+v^2}}+\frac{v^2}{4+v^2}$
3	VI	LS	Log	$\frac{4}{(1+e^v)^2}$
4	I	LS	Cosh	$1+\frac{4}{a}\sinh^{-1}(\frac{-v}{a})+\frac{4}{a^2}(\sinh^{-1}(\frac{-v}{a}))^2$
5	V	LS	Sec	$1+\frac{4}{a}\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v})+\frac{4}{a^2}\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v})^2$
6	V	LS	Log-Cos	$(\frac{2}{a}\tan^{-1}(v)-1)^2$
7$*$	III	Exp	Log	$e^{\frac{-v}{2}}$
8	III	Exp	Exp	$\sqrt{\frac{\sqrt{4+v^2}-v}{\sqrt{4+v^2}+v}}=\frac{1}{2}(\sqrt{4+v^2}-v)$
9$*$	III	Log	Log	$\log(1+e^{-v})$
10	III	Log	Exp	$-\log(\frac{1}{2}+\frac{v}{2\sqrt{4+v^2}})$
11	VI	Sec	Log	$\sec(\frac{0.5a(1-e^v)}{1+e^v})[-1+(\frac{a}{1+e^v})\tan(\frac{0.5a(1-e^v)}{1+e^v})]$
12	VI	Sec	Exp	$-\sec(\frac{av}{2\sqrt{4+v^2}})[1+a(\frac{1}{2}-\frac{v}{2\sqrt{4+v^2}})\tan(\frac{av}{2\sqrt{4+v^2}})]$
13	I	Sec	Sec	$-\sec(\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v}))-(\frac{1}{2}+\frac{1}{a}\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v}))v$
14	I	Sec	Log-Cos	$-\sec(\tan^{-1}(v))[1+a(\frac{1}{2}-\frac{1}{a}\tan^{-1}(v))v]$
15	I	Cosh	LS	$-\cosh(\frac{-av}{8})+a(\frac{4-v}{8})\sinh(\frac{-av}{8})$
16	VI	Cosh	Log	$-\cosh(\frac{0.5a(1-e^v)}{1+e^v})+a(\frac{1}{1+e^v})\sinh(\frac{0.5a(1-e^v)}{1+e^v}))$
17	V	Cosh	Sec	$-\cosh(\sin^{-1}(A))+(\frac{1}{2}+\frac{1}{a}\sin^{-1}(A))\sinh(\sin^{-1}(A))$ $A=\frac{a-\sqrt{a^2+4v^2}}{2v}$
18	I	Cosh	Cosh	$-\cosh(\sinh^{-1}(\frac{-v}{a}))-(\frac{1}{2}+\frac{1}{a}\sinh^{-1}(\frac{-v}{a}))v$
19	VI	Cosh	Exp	$-\cosh(\frac{av}{2\sqrt{4+v^2}})-a(\frac{1}{2}-\frac{v}{2\sqrt{4+v^2}})\sinh(\frac{av}{2\sqrt{4+v^2}})$
20	V	Cosh	Log-Cos	$-\cosh(A)-(\frac{1}{2}-A)\sinh(A)$ $A=\frac{1}{a}\tan^{-1}(v)$
21	VI	Log-Cos	Log	$\frac{1}{a}\log(\frac{\cos(a(\frac{e^v}{1+e^v}-\frac{1}{2}))}{\cos(\frac{a}{2})})-(1-\frac{e^v}{1+e^v})\tan(a(\frac{e^v}{1+e^v}-\frac{1}{2}))$
22	VI	Log-Cos	Exp	$\frac{1}{a}\log(\frac{\cos(\frac{av}{2\sqrt{4+v^2}})}{\cos(\frac{a}{2})})-(\frac{1}{2}-\frac{v}{2\sqrt{4+v^2}})\tan(\frac{av}{2\sqrt{4+v^2}})$
23	I	Log-Cos	Sec	$\frac{1}{a}\log(\frac{\cos(-A)}{\cos(\frac{a}{2})})-(\frac{1}{2}+\frac{1}{a}A)\tan(-A)$ $A=\sin^{-1}(\frac{a-\sqrt{a^2+4v^2}}{2v})$
24	I	Log-Cos	Log-Cos	$-\frac{1}{a}\log(\cos(\frac{a}{2})\sqrt{1+v^2})-(\frac{1}{2}-\frac{1}{a}\tan^{-1}(v))v$

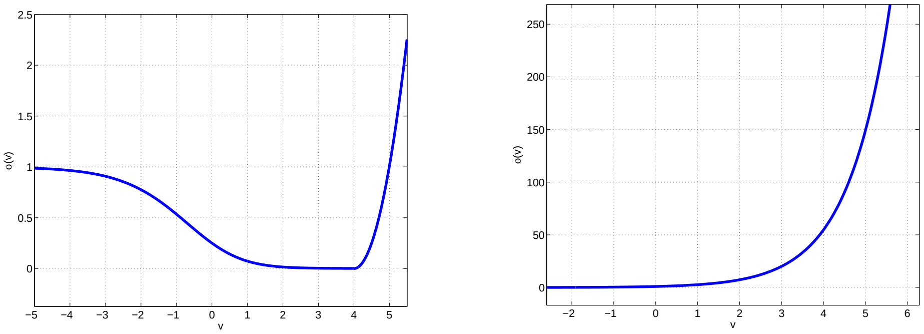


Figure II.8 Example of loss function shapes that are not possible.

on the boundedness properties of the loss function. Quasi convexity does not define boundedness since it is both possible to have a loss function that is quasi convex and unbounded as well as a loss function that is quasi convex and bounded. Also the four varieties are very loose in terms of the restrictions they impose on the shape of the loss. For example they do not deny the possibility of having loss functions with the shapes seen in Figure II.8, i.e. a loss that is bounded on the negative side and unbounded on the positive side. Yet, in this section we will show that such loss functions are not possible and will fully analyze and limit the possible loss function shapes, restricting them to seven types. Furthermore, we will provide simple conditions for designing and deriving each of the seven loss types.

Definition 5. *A loss function is said to have negative boundedness if* $\phi(-\infty) < \infty$ *and positive boundedness if* $\phi(\infty) < \infty$*. Conversely, a loss function is said to be negatively unbounded if* $\phi(-\infty) = \infty$ *and positively unbounded if* $\phi(\infty) = \infty$

It is easy to verify that the well known exponential, logistic and least squares loss functions are all negatively unbounded. Such unbounded loss functions will be more sensitive to outliers as such points will introduce an infinite loss. Having defined boundedness for a loss function, we must note that boundedness is not equivalent with having a zero derivative at infinity. A bounded function has zero derivative at infinity $\phi'(\infty) = 0$, but a function with zero derivative at infinity $\phi'(\infty) = 0$ is not necessarily bounded. For example the log function

$\log(x)$ has zero derivative at infinity since $\lim_{x\to\infty} \frac{1}{x} = 0$ but is unbounded since $\lim_{x\to\infty} \log(x) = \infty$.

We also define boundedness for link functions.

Definition 6. *A link function is said to be bounded if $f^{-1}(-\infty) = b > -\infty$. Conversely, a link function is said to be unbounded if $f^{-1}(-\infty) = -\infty$.*

Obviously for a bounded link function, using the symmetry property of equation (II.18), we can equivalently write $f^{-1}(\infty) = 1 - b < \infty$. Table II.6 shows the range of different link functions. The Exp, Log, Sec, and Log-Cos link functions are all bounded whereas the LS and Cosh link functions are unbounded.

The following theorem limits the possible shapes a Bayes consistent loss function can have in terms of boundedness.

Theorem 23. *If a Bayes consistent loss function is negatively bounded, then it must also be positively bounded .*

Proof. Given that the loss function is negatively bounded we can write

$$\phi(-\infty) = -J[f^{-1}(-\infty)] - (1 - f^{-1}(-\infty))J'[f^{-1}(-\infty)] = a. \tag{II.90}$$

Also from equation (II.16) we can write

$$J'(f^{-1}(-\infty)) = \phi(\infty) - \phi(-\infty) = \phi(\infty) - a. \tag{II.91}$$

or

$$\phi(\infty) = J'(f^{-1}(-\infty)) + a. \tag{II.92}$$

Two cases are possible given that $J(\eta)$ is convex and $J'(f^{-1}(-\infty)) < 0$. The first case is when $J'(f^{-1}(-\infty)) = c > -\infty$. In which case $\phi(\infty) = c + a$ is positively bounded thus proving the theorem.

The second case is when $J'(f^{-1}(-\infty)) = -\infty$. In which case we can write

$$\phi(\infty) = J'(f^{-1}(-\infty)) + a = -\infty. \tag{II.93}$$

which again proves that $\phi(\infty) = -\infty < \infty$ is positively bounded.

■

The converse of this theorem is obviously not correct and a positively bounded loss is not necessarily negatively bounded. Loss functions such as the exponential or logistic loss are examples of such loss functions that are negatively unbounded but positively bounded. Theorem 23 shows that a loss function such as the one in Figure II.8 is not possible, since it is negatively bounded yet positively unbounded.

The following theorem restricts the possible shapes a Bayes consistent loss function can take in terms of boundedness to seven different types.

Theorem 24. *The shape of a Bayes consistent loss function is restricted to seven types in terms of boundedness.*

I: $\phi(-\infty) = \infty$*,* $\phi(\infty) = \infty$ *and nondecreasing up to a point and then non increasing.*

II: $\phi(-\infty) = \infty$*,* $\phi(\infty) = c$ *and nondecreasing up to a point and then non increasing.*

III: $\phi(-\infty) = \infty$*,* $\phi(\infty) = c$ *and nondecreasing .*

IV: $\phi(-\infty) = \infty$*,* $\phi(\infty) = -\infty$ *and nondecreasing .*

V: $\phi(-\infty) = c_1$*,* $\phi(\infty) = c_2$ *and nondecreasing up to a point and then non increasing.*

VI: $\phi(-\infty) = c_1$*,* $\phi(\infty) = c_2$ *and nondecreasing .*

VII: $\phi(-\infty) = c$*,* $\phi(\infty) = -\infty$ *and nondecreasing .*

Proof. From theorem 20 we know that the loss cannot be non increasing. This means that $\phi(-\infty) \neq -\infty$, thus restricting $\phi(-\infty)$ to $\phi(-\infty) = \infty$ or $\phi(-\infty) = c$. On the other hand $\phi(\infty)$ can equal $\phi(\infty) \neq \infty$, $\phi(\infty) \neq c$ or $\phi(\infty) \neq -\infty$. From theorem 20 we also know that the loss can be (1) either nondecreasing or (2) nondecreasing up to a point and then non increasing. Theorem 23 excludes the case of

$$\phi(-\infty) = c, \phi(\infty) = \infty. \tag{II.94}$$

This coupled with the fact that the loss is either convex or quasi convex (theorem 20) leaves only the seven possibilities mentioned in the theorem. ■

The following theorems provide simple recipes for deriving each of the seven types of loss functions. These recipes are not restrictive but rather only constructive, meaning that certain loss types can be derived in more than one manner, sometimes involving complex interplay between the risk and link function. Rather, we provide simple recipes for deriving each of the loss types, ensuring that a loss is of a certain type without having to resort to trial and error.

Theorem 25. *A loss will be of Type-I if $f^{-1}(v)$ and $J(\eta)$ are a canonical pair and $f^{-1}(v) > 1$ for some v .*

Proof. Since the loss is canonical we know from Theorem 12 that the loss will be convex. Also, given that $f^{-1}(v) > 1$ for some value of v, Theorem 20 tells us that this loss will be nondecreasing up to a point and then non increasing. The loss is thus unbounded on both sides and is a Type-I loss. ■

Theorem 26. *A loss will be of Type-II if $f^{-1}(\infty) = c > 1$, $J(f^{-1}(\infty)) = \infty$, $J'(f^{-1}(\infty)) = \infty$ and $-J(f^{-1}(\infty)) - (1-c)J'(f^{-1}(\infty)) = a$.*

Proof. We can write

$$\phi(\infty) = -J(f^{-1}(\infty)) - (1-c)J'(f^{-1}(\infty)) = a \tag{II.95}$$

thus $\phi(v)$ is positively bounded. From equation II.16 we can also write

$$\phi(-\infty) - \phi(\infty) = J'(f^{-1}(\infty)) \tag{II.96}$$

or

$$\phi(-\infty) = a + J'(f^{-1}(\infty)) = \infty \tag{II.97}$$

thus proving that $\phi(v)$ is negatively unbounded. The fact that $f^{-1}(\infty) = c > 1$ means that from Theorem 20 we know that it will be nondecreasing up to a point and then non increasing. The loss is thus of Type-II. ■

Theorem 27. *A loss will be of Type-III if $f^{-1}(v)$ and $J(\eta)$ are a canonical pair, $f^{-1}(\infty) = 1$ and $c + J(1) = \lim_{v\to\infty} -(1 - f^{-1}(v))v$.*

Proof. Since the loss is canonical we know from Theorem 12 that the loss will be convex. given that $\phi(v)$ is convex and nondecreasing we can conclude that $\phi(-\infty) = \infty$. We can also write

$$\phi(\infty) = \lim_{v\to\infty} -J(f^{-1}(v)) - (1 - f^{-1}(v))v = c \tag{II.98}$$

where we have used the fact that $-J(f^{-1}(\infty)) = J(1)$ and $c + J(1) = \lim_{v\to\infty} -(1 - f^{-1}(v))v$, thus showing that $\phi(v)$ is of Type-III. ■

Theorem 28. *A loss will be of Type-IV if $f^{-1}(v)$ and $J(\eta)$ are a canonical pair, $f^{-1}(\infty) = b < 1$.*

Proof. Since the loss is canonical we know from Theorem 12 that the loss will be convex. given that $\phi(v)$ is convex and nondecreasing we can conclude that $\phi(-\infty) = \infty$. Also, given that $f^{-1}(\infty) = b < 1$ we can write

$$\phi'(\infty) = b - 1 \neq 0 \tag{II.99}$$

which means that the $\phi(-\infty) = -\infty$. Also since $f^{-1}(\infty) = b < 1$ we know that the loss will be nondecreasing and thus of Type-IV. ■

Theorem 29. *A loss will be of Type-V if $f^{-1}(-\infty) = b > 1$ and $J(b) = k_1 < \infty$ and $J'(b) = k_2 > -\infty$.*

Proof.

$$\begin{aligned}\lim_{v\to-\infty}\phi(v) &= \lim_{v\to-\infty} -J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v)) \qquad \text{(II.100)}\\ &= -J(b)-(1-b)J'(b) = -k_1-(1-b)k_2\end{aligned}$$

which shows that $\phi(v)$ is negatively bounded. we can also write

$$\begin{aligned}&\lim_{v\to\infty}\phi(v) = \lim_{v\to\infty} -J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v)) \qquad \text{(II.101)}\\ &= -J(1-b)-(1-(1-b))J'(1-b) = -J(b)+bJ'(b) = -k_1+bk_2\end{aligned}$$

which shows that the loss is also positively bounded and we have used the equalities $f^{-1}(v) = 1-f^{-1}(-v)$, $J(\eta) = J(1-\eta)$ and $J'(\eta) = -J'(1-\eta)$. Given that $f^{-1}(-\infty) = b > 1$, means that from Theorem 20 the loss will be nondecreasing up to a point and then non increasing and thus of Type-V. ■

Theorem 30. *A loss will be of Type-VI if $f^{-1}(-\infty) = b \leq 1$ and $J(b) = k_1 < \infty$ and $J'(b) = k_2 > -\infty$.*

Proof.

$$\begin{aligned}\lim_{v\to-\infty}\phi(v) &= \lim_{v\to-\infty} -J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v)) \qquad \text{(II.102)}\\ &= -J(b)-(1-b)J'(b) = -k_1-(1-b)k_2\end{aligned}$$

which shows that $\phi(v)$ is negatively bounded. we can also write

$$\begin{aligned}&\lim_{v\to\infty}\phi(v) = \lim_{v\to\infty} -J(f^{-1}(v)) - (1-f^{-1}(v))J'(f^{-1}(v)) \qquad \text{(II.103)}\\ &= -J(1-b)-(1-(1-b))J'(1-b) = -J(b)+bJ'(b) = -k_1+bk_2\end{aligned}$$

which shows that the loss is also positively bounded and we have used the equalities $f^{-1}(v) = 1-f^{-1}(-v)$, $J(\eta) = J(1-\eta)$ and $J'(\eta) = -J'(1-\eta)$. Given that $f^{-1}(-\infty) = b \leq 1$, means that from Theorem 20 the loss will be nondecreasing and thus of Type-VI. ■

Theorem 31. *A loss will be of Type-VII if $f^{-1}(-\infty) = c > 0$ and $J(f^{-1}(-\infty)) = \infty$ and $J'(f^{-1}(-\infty)) = -\infty$. and $-J(f^{-1}(-\infty)) - (1-c)J'(f^{-1}(-\infty)) = a$.*

Proof. We can write

$$\phi(-\infty) = -J(f^{-1}(-\infty)) - (1-c)J'(f^{-1}(-\infty)) = a \qquad \text{(II.104)}$$

thus $\phi(v)$ is negatively bounded. From equation II.16 we can also write

$$\phi(\infty) - \phi(-\infty) = J'(f^{-1}(-\infty)) \qquad \text{(II.105)}$$

or

$$\phi(\infty) = a + J'(f^{-1}(-\infty)) = -\infty. \qquad \text{(II.106)}$$

The fact that $f^{-1}(-\infty) = c > 0$ means that $f^{-1}(\infty) = 1 - c < 1$ for all v and thus from Theorem 20 we know that it will be nondecreasing. The loss is thus of Type-VII. ■

The 24 Bayes consistent loss functions presented in Table II.8 are categorized into the seven different types.

Type-1 losses include functions such as the well known Least Squares (loss #1) or the novel losses #13 and #14 which are convex and novel non convex losses such as losses #4 and #23 . Figure II.9-A plots a number of such losses. These loss functions might penalize correct classifications (positive margin) heavily and as much as incorrect classifications (negative margin) thus making then unsuitable for classification applications. This happens to be the case for the LS loss function which is not typically used for classification. Nevertheless these loss functions can be made to resemble Type-III, VI and even Type-V loss function with the correct choice of parameters as is the case with loss #23 which is unbounded and non convex and has an interesting shape making it similar to Type-VI or Type-V losses.

Type-III losses include the well known exponential loss #7 and Logistic loss #9 or the canonical losses of #8 and #9 which by Theorem 27 fall into this category. These type of losses are generally well suited for classification but can suffer from being sensitive to outliers. This comes from the unbounded loss assigned to the negative margin. Figure II.9-B plots a number of such losses which can be both convex and non convex in shape.

Type-VI loss functions are also generally well suited for classification problems especially when outliers are present in the data. The bounded nature of these loss functions assigns a bounded loss to outliers thus deemphasizing their affect. Figure II.9-C plots a number of such losses.

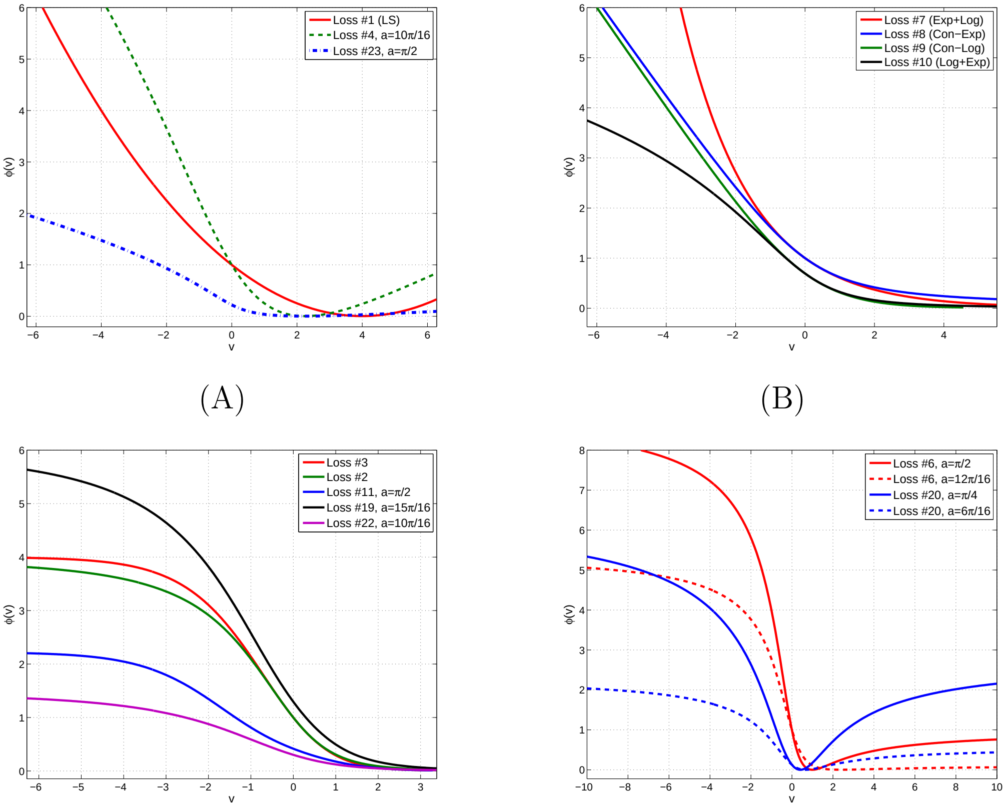
Finally, Type-V loss functions will not only be resistant to outliers but will also be robust to over training because a nonzero (yet bounded) loss is assigned to points that are classified "too correctly". Such loss functions are especially suitable for situations where the data is contaminated with noise and we would like to prevent the classification algorithm from learning the noise and over training. Figure II.9-D plots a number of such losses. Different values of the parameter a change the ratio between the positive margin bound and the negative margin bound, thus controlling how much loss is assigned to being "too correct" in relation to being "too wrong".

Examples of loss functions of Type II, IV and VII are not present in Table II.8 because we currently do not know of any closed form functions $J(\eta)$ and $f^{-1}(v)$ that satisfy the general symmetry requirements and also satisfy the requirements of Theorems 26, 28 or 31.

In summary, Table II.8 can be used as a reference for choosing different loss functions to use based on the particular problem and understanding of the data at hand and desired performance characteristics such as robustness to outliers and noise. Also, previously stated theorems can be used as a guide for designing and deriving other novel loss functions that are specially tailored for certain problems.

II.G Summary and discussion

In this chapter, we have presented a new framework for the design and analysis of Bayes consistent loss functions. The two fields of risk minimization



(A) (B)

(C) (D)

Figure II.9 Examples of (A) Type-I (B) Type-III (C) Type-VI (D) Type-V Loss functions.

in machine learning and probability elicitation in statistics were related. The probability elicitation view allowed us to obtain a generative formula for deriving novel Bayes consistent loss functions. Specifically, margin enforcing composite losses were considered. We showed that any functional form of the minimum conditional risk, which satisfies some mild constraints, supports many link and loss function pairs. Hence, by selecting a class of link functions, it is possible to tailor the loss, so as to guarantee classifiers with desirable traits.

Next, canonical loss functions were fully considered and analyzed. The general case of non canonical loss functions was also studied and the class of Bayes consistent loss functions were partitioned into four varieties based on their convexity properties. The convexity properties of the loss, risk and empirical risk of Bayes consistent loss functions were also studied in detail which, for the first time, enable the derivation of non convex Bayes consistent loss functions. Finally a taxonomy of Bayes consistent loss functions was provided based on their shape and boundedness properties.

II.H Acknowledgments

The author would like to thank Mohammad Saberian and Alireza Masnadi-Shirazi for helpful discussions and comments.

The text of Chapter II, in part, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “A Statistical View of Margin Based Risk Minimization”, to be submitted to *Annals of Statistics*, 2011. The dissertation author was a primary researcher and an author of the cited material.

Chapter III

The design of canonical Bayes consistent loss functions

III.A Introduction

Optimal classifiers minimize the expected value of a loss function, or risk. Losses commonly used in machine learning are upper-bounds on the zero-one classification loss of classical Bayes decision theory. When the resulting classifier converges asymptotically to the Bayes decision rule, as training samples increase, the loss is said to be Bayes consistent. Examples of such losses include the hinge loss, used in SVM design, the exponential loss, used by boosting algorithms such as AdaBoost, or the logistic loss, used in both classical logistic regression and more recent methods, such as LogitBoost. Unlike the zero-one loss, these losses assign a penalty to examples correctly classified but close to the boundary. This guarantees a classification margin, and improved generalization when learning from finite datasets [98]. Although the connections between large-margin classification and classical decision theory have been known since [35], the set of Bayes consistent large-margin losses has remained small. The design of such losses has been studied in Chapter II. By establishing connections to the classical literature in probability elicitation [82], we introduced a generic framework for the derivation of Bayes consistent losses. The main idea is that there are three quantities that matter in risk minimization: the loss function ϕ, a corresponding optimal link function f^*_ϕ, which maps posterior class probabilities to classifier predictions, and a minimum risk C^*_ϕ, associated with the optimal link.

While the standard approach to classifier design is to define a loss ϕ, and then optimize it to obtain f^*_ϕ and C^*_ϕ, we showed in Chapter II that there is an alternative: to specify f^*_ϕ and C^*_ϕ, and analytically derive the loss ϕ. The advantage is that this makes it possible to manipulate the properties of the loss, while *guaranteeing* that it is Bayes consistent. The main limitation of the framework in Chapter II is that it is not totally constructive. It turns out that many pairs (C^*_ϕ, f^*_ϕ) are compatible with any Bayes consistent loss ϕ. Furthermore, while there is a closed form relationship between ϕ and (C^*_ϕ, f^*_ϕ), this relationship is far from simple. This makes it difficult to understand how the properties of the loss are influenced by the properties of either C^*_ϕ or f^*_ϕ. In practice, the design has to resort to trial and error, by 1) testing combinations of the latter and, 2) verifying whether the loss has the desired properties. This is feasible when the goal is to enforce a broad loss property, e.g. that a robust loss should be bounded for negative margins [57], but impractical when the goal is to exercise a finer degree of control.

In this chapter, we consider one such problem: how to control the size of the margin enforced by the loss. We start by showing that, while many pairs (C^*_ϕ,f^*_ϕ) are compatible with a given ϕ, one of these pairs establishes a very tight connection between the optimal link and the minimum risk: that f^*_ϕ is the derivative of C^*_ϕ. We refer to the risk function associated with such a pair as a *canonical risk*, and show that it leads to an equally tight connection between the pair (C^*_ϕ,f^*_ϕ) and the loss ϕ. For a canonical risk, all three functions can be obtained from each other with one-to-one mappings of trivial analytical tractability. This enables a detailed analytical study of how C^*_ϕ or f^*_ϕ affect ϕ. We consider the case where the inverse of f^*_ϕ is a sigmoidal function, i.e. f^*_ϕ is *inverse-sigmoidal*, and show that this strongly constrains the loss. Namely, the latter becomes 1) convex, 2) monotonically decreasing, 3) linear for large negative margins, and 4) constant for large positive margins. This implies that, for a canonical risk, the choice of a particular link in the inverse-sigmoidal family *only* impacts the behavior of ϕ around the origin, i.e. the size of the margin enforced by the loss. This quantity is then shown to depend only on the slope of the sigmoidal inverse-link at the origin. Since this property can be controlled by a single parameter, the latter becomes a margin-tunning parameter, i.e. a parameter that determines the margin of the optimal classifier.

The requirements of 1) a canonical risk, and 2) an inverse-sigmoidal link are shown not to be unduly restrictive for classifier design. In fact, approaches like logistic regression or LogitBoost are special cases of the proposed framework. Furthermore, it is shown that a canonical loss can be derived from any cumulative distribution function associated with a symmetric probability density of zero mean. This guarantees that the family of margin controllable losses is at least as large as the family of zero mean symmetric probability densities. Overall, this work establishes a number of approaches to the derivation of canonical loss functions with *explicit control* of the classification margin: 1) variable margin extensions of existing losses, by reparametrization of their link functions, 2) derivation of new losses from the minimum risks associated with existing non-canonical losses, and 3) derivation of new losses from cumulative distribution functions. These possibilities are illustrated through the design of four new margin controllable losses.

The design of boosting algorithms, based on canonical losses is then considered. Starting from the gradientBoost framework of [36], it is shown that the choice of loss only impacts the weight mechanism of the resulting boosting algorithm. More precisely, the weighting function is shown to be the complement of the sigmoidal inverse link associated with the loss. This has two interesting consequences. First, it establishes a common boosting framework for all canonical losses, which is denoted *canonical gradientBoost.* Since all canonical gradientBoost algorithms are equivalent up to example weighting, this framework enables a direct comparison of the impact of the loss on classifier performance. Second, it guarantees that all canonical gradientBoost algorithms have a weighting mechanism with sensible properties, namely a saturating weight function that reduces the impact of outliers, guaranteeing robust classification. A number of experiments are conducted to verify these properties, and study the effect of margin-control on the classification accuracy of the four proposed variable-margin losses. These are shown to outperform the fixed-margin counterparts used by existing algorithms. Finally, it is shown that cross-validation of the margin parameter leads to classifiers with the best performance on all datasets tested, and that the impact of margin tuning is most significant as training set size decreases.

This chapter is organized as follows. Section III.B briefly reviews the problem of classifier design by risk minimization, and its connections to probability elicitation. Canonical risks and canonical risk minimization are introduced in Section III.C, which establishes mathematical relationships between loss, link, and minimum risk. The properties of canonical losses associated with inverse-sigmoidal links are then discussed in Section III.D, and used to derive the four proposed variable margin losses in Section III.E. The canonical gradientBoosting framework is then introduced in Section III.F. An experimental study of the algorithms derived from the proposed variable-margin losses are reported in Section III.G. Finally, a summary is provided in Section III.H.

III.B Loss functions for classification

We start by briefly reviewing the theory of Bayes consistent classifier design. See [35, 17, 119, 57] for further details. A classifier h maps a feature vector $\mathbf{x} \in \mathcal{X}$ to a class label $y \in \{-1, 1\}$, according to

$$h(\mathbf{x}) = sign[p(\mathbf{x})], \tag{III.1}$$

where $p : \mathcal{X} \to \mathbb{R}$ is denoted as the classifier predictor. Feature vectors and class labels are drawn from probability distributions $P_{\mathbf{X}}(\mathbf{x})$ and $P_Y(y)$ respectively. Given a non-negative loss function $L(\mathbf{x}, y)$, the classifier is optimal if it minimizes the risk

$$R = E_{\mathbf{X},Y}[L(h(\mathbf{x}), y)]. \tag{III.2}$$

This is equivalent to minimizing the conditional risk

$$E_{Y|\mathbf{X}}[L(h(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}] \tag{III.3}$$

for all $\mathbf{x} \in \mathcal{X}$. It is useful to express $p(\mathbf{x})$ as a composition of two functions,

$$p(\mathbf{x}) = f(\eta(\mathbf{x})), \tag{III.4}$$

where $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$, and $f : [0, 1] \to \mathbb{R}$ is a *link function*. Classifiers are frequently designed to be optimal with respect to the zero-one loss

$$L_{0/1}(f, y) = \frac{1 - sign(yf)}{2} = \begin{cases} 0, & \text{if } y = sign(f); \\ 1, & \text{if } y \neq sign(f), \end{cases} \tag{III.5}$$

where we omit the dependence on $\mathbf{x}$ for notational simplicity. The associated conditional risk is

$$C_{0/1}(\eta, f) = \eta\frac{1 - sign(f)}{2} + (1-\eta)\frac{1 + sign(f)}{2} = \begin{cases} 1-\eta, & \text{if } f \geq 0; \\ \eta, & \text{if } f < 0. \end{cases} \quad \text{(III.6)}$$

The risk is minimized if

$$\begin{cases} f(\mathbf{x}) > 0 & \text{if } \eta(\mathbf{x}) > \frac{1}{2} \\ f(\mathbf{x}) = 0 & \text{if } \eta(\mathbf{x}) = \frac{1}{2} \\ f(\mathbf{x}) < 0 & \text{if } \eta(\mathbf{x}) < \frac{1}{2}. \end{cases} \quad \text{(III.7)}$$

Examples of optimal link functions include

$$f^* = 2\eta - 1 \quad \text{and} \quad f^* = \log\frac{\eta}{1-\eta}. \quad \text{(III.8)}$$

The associated optimal classifier $h^* = sign[f^*]$ is the well known Bayes decision rule (BDR), and the associated minimum conditional (zero-one) risk is

$$C^*_{0/1}(\eta) = \eta\left(\frac{1}{2} - \frac{1}{2}sign(2\eta - 1)\right) + (1-\eta)\left(\frac{1}{2} + \frac{1}{2}sign(2\eta - 1)\right). \quad \text{(III.9)}$$

A loss which is minimized by the BDR is *Bayes consistent*. A number of Bayes consistent alternatives to the 0-1 loss are commonly used. These include the exponential loss of boosting, the log loss of logistic regression, and the hinge loss of SVMs. They have the form

$$L_\phi(f, y) = \phi(yf) \quad \text{(III.10)}$$

for different functions ϕ. These functions assign a non-zero penalty to small positive yf, encouraging the creation of a margin, a property not shared by the 0-1 loss. The resulting *large-margin* classifiers have better generalization than those produced by the latter [98]. The associated conditional risk

$$C_\phi(\eta, f) = \eta\phi(f) + (1-\eta)\phi(-f). \quad \text{(III.11)}$$

is minimized by the link

$$f^*_\phi(\eta) = \arg\min_f C_\phi(\eta, f) \quad \text{(III.12)}$$

Table III.1 Losses ϕ, optimal link $f^*_\phi(\eta)$, optimal inverse link $[f^*_\phi]^{-1}(v)$, and minimum conditional risk $C^*_\phi(\eta)$ for popular learning algorithms.

Algorithm	$\phi(v)$	$f^*_\phi(\eta)$	$[f^*_\phi]^{-1}(v)$	$C^*_\phi(\eta)$
SVM	$\max(1-v,0)$	$sign(2\eta-1)$	NA	$1-\|2\eta-1\|$
Boosting	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{e^{2v}}{1+e^{2v}}$	$2\sqrt{\eta(1-\eta)}$
Logistic Regression	$\log(1+e^{-v})$	$\log\frac{\eta}{1-\eta}$	$\frac{e^{v}}{1+e^{v}}$	$\text{-}\eta\log\eta-(1-\eta)\log(1-\eta)$

leading to the minimum conditional risk function

$$C^*_\phi(\eta) = C_\phi(\eta, f^*_\phi). \tag{III.13}$$

Table III.1 lists the loss, optimal link, and minimum risk of some of the most popular learning methods.

Conditional risk minimization is closely related to classical probability elicitation in statistics [82]. Here, the goal is to find the probability estimator $\hat{\eta}$ that maximizes the expected reward

$$I(\eta,\hat{\eta}) = \eta I_1(\hat{\eta}) + (1-\eta)I_{-1}(\hat{\eta}), \tag{III.14}$$

where $I_1(\hat{\eta})$ is the reward for prediction $\hat{\eta}$ when event $y=1$ holds and $I_{-1}(\hat{\eta})$ the corresponding reward when $y=-1$. The functions $I_1(\cdot), I_{-1}(\cdot)$ should be such that the expected reward is maximal when $\hat{\eta}=\eta$, i.e.

$$I(\eta,\hat{\eta}) \le I(\eta,\eta) = J(\eta), \;\; \forall\eta \tag{III.15}$$

with equality if and only if $\hat{\eta}=\eta$. The conditions under which this holds are as follows.

Theorem 32. *[82] Let $I(\eta,\hat{\eta})$ and $J(\eta)$ be as defined in (III.14) and (III.15). Then 1) $J(\eta)$ is convex and 2) (III.15) holds if and only if*

$$\begin{aligned} I_1(\eta) &= J(\eta) + (1-\eta)J'(\eta) & \text{(III.16)}\\ I_{-1}(\eta) &= J(\eta) - \eta J'(\eta). & \text{(III.17)} \end{aligned}$$

Hence, starting from any convex $J(\eta)$, it is possible to derive $I_1(\cdot), I_{-1}(\cdot)$ so that (III.15) holds. This enables the following connection to risk minimization from Chapter II.

Theorem 33. *[57] Let $J(\eta)$ be as defined in (III.15) and f a continuous function. If the following properties hold*

1. *$J(\eta) = J(1-\eta)$,*

2. *f is invertible with symmetry*

$$f^{-1}(-v) = 1 - f^{-1}(v), \tag{III.18}$$

then the functions $I_1(\cdot)$ and $I_{-1}(\cdot)$ derived with (III.16) and (III.17) satisfy the following equalities

$$\begin{aligned} I_1(\eta) &= -\phi(f(\eta)) \qquad &\text{(III.19)}\\ I_{-1}(\eta) &= -\phi(-f(\eta)), \qquad &\text{(III.20)}\end{aligned}$$

with

$$\phi(v) = -J[f^{-1}(v)] - (1 - f^{-1}(v))J'[f^{-1}(v)]. \tag{III.21}$$

Under the conditions of the theorem, $I(\eta, \hat{\eta}) = -C_\phi(\eta, f)$. This establishes a new path for classifier design [57]. Rather than specifying a loss ϕ and minimizing $C_\phi(\eta, f)$, so as to obtain whatever optimal link f^*_ϕ and minimum expected risk $C^*_\phi(\eta)$ results, it is possible to specify f^*_ϕ and $C^*_\phi(\eta)$ and derive, from (III.21) with $J(\eta) = -C^*_\phi(\eta)$, the underlying loss ϕ. The main advantage is the ability to control directly the quantities that matter for classification, namely the predictor and risk of the optimal classifier.The only conditions are that $C^*_\phi(\eta) = C^*_\phi(1-\eta)$ and (III.18) holds for f^*_ϕ.

III.C Canonical risk minimization

In general, given $J(\eta) = -C^*_\phi(\eta)$, there are multiple pairs (ϕ, f^*_ϕ) that satisfy (III.21). Hence, specification of either the minimum risk or optimal link does not completely characterize the loss. This makes it difficult to control some important properties of the latter, such as the margin. In this chapter, we consider an important special case, where such control is possible. We start with a lemma that relates the symmetry conditions, on $J(\eta)$ and $f^*_\phi(\eta)$, of Theorem 33.

Lemma 34. *Let $J(\eta)$ be a strictly convex and differentiable function such that $J(\eta) = J(1-\eta)$. Then $J'(\eta)$ is invertible and*

$$[J']^{-1}(-v) = 1 - [J']^{-1}(v). \tag{III.22}$$

Proof. From the strict convexity of $J(\eta)$ it follows that $J'(\eta)$ has positive derivative for all η . Hence, $J'(\eta)$ is invertible. From the symmetry of $J(\eta)$,

$$J'(\eta) \;=\; -J'(1-\eta)$$

and, for any v such that $\eta = [J']^{-1}(v)$,

$$\begin{aligned} v &= -J'(1-[J']^{-1}(v)) \\ [J']^{-1}(-v) &= 1-[J']^{-1}(v). \end{aligned}$$

■

Hence, under the conditions of Theorem 33, the derivative of $J(\eta)$ has the *same* symmetry as $f^*_\phi(\eta)$. Since this symmetry is the only constraint on f^*_ϕ, the former can be used as the latter. Whenever this holds, the risk is said to be in canonical form, and (f^*, J) are denoted a canonical pair [17] .

Definition 7. *Let $J(\eta)$ be as defined in (III.15), and $C^*_\phi(\eta) = -J(\eta)$ a minimum risk. If the optimal link associated with $C^*_\phi(\eta)$ is*

$$f^*_\phi(\eta) = J'(\eta) \tag{III.23}$$

the risk $C_\phi(\eta, f)$ is said to be in canonical form. $f^*_\phi(\eta)$ is denoted a canonical link and $\phi(v)$, the loss given by (III.21), a canonical loss.

Note that (III.23) does not hold for all risks. For example, the risk of boosting is derived from the convex, differentiable, and symmetric $J(\eta) = -2\sqrt{\eta(1-\eta)}$. Since this has derivative

$$J'(\eta) = \frac{2\eta - 1}{\sqrt{\eta(1-\eta)}} \neq \frac{1}{2}\log\frac{\eta}{1-\eta} = f^*_\phi(\eta), \tag{III.24}$$

the risk is not in canonical form. What follows from (III.23) is that *it is possible* to derive a canonical risk for *any* maximal reward $J(\eta)$, including that of boosting $(J(\eta) = -2\sqrt{\eta(1-\eta)})$. This is discussed in detail in Section III.E.

While canonical risks can be easily designed by specifying either $J(\eta)$ or $f^*_\phi(\eta)$, and then using (III.21) and (III.23), it is much less clear how to directly specify a loss $\phi(v)$ for which (III.21) holds with a canonical pair (f^*, J). The following result solves this problem.

Theorem 35. *Let $C_\phi(\eta, f)$ be the canonical risk derived from a convex and symmetric $J(\eta)$. Then*

$$\phi'(v) = -[J']^{-1}(-v) = [f^*_\phi]^{-1}(v) - 1. \tag{III.25}$$

Proof. Given that $C_\phi(\eta, f)$ is a canonical risk and (III.23), the loss function of (III.21) can be simplified into

$$\begin{aligned}\phi(v) &= -J\{[f^*_\phi]^{-1}(v)\} - (1 - [f^*_\phi]^{-1}(v))J'\{[f^*_\phi]^{-1}(v)\} \\ &= -J\{[J']^{-1}(v)\} - (1 - [J']^{-1}(v))J'\{[J']^{-1}(v)\} \\ &= -J\{[J']^{-1}(v)\} - (1 - [J']^{-1}(v))v. \end{aligned} \tag{III.26}$$

The proof follows from taking derivatives on both sides,

$$\begin{aligned}\phi'(v) &= -J'\{[J']^{-1}(v)\}\{[J']^{-1}\}'(v) - (1 - [J']^{-1}(v)) + \{[J']^{-1}\}'(v)v \\ &= -v\{[J']^{-1}\}'(v) - (1 - [J']^{-1}(v)) + \{[J']^{-1}\}'(v)v \\ &= -(1 - [J']^{-1}(v)) \\ &= -[J']^{-1}(-v),\end{aligned}$$

where we have also used (III.22). Furthermore, using (III.23),

$$\begin{aligned}\phi'(v) &= -(1 - [J']^{-1}(v)) && \text{(III.27)} \\ &= -(1 - [f^*]^{-1}(v)) && \text{(III.28)} \\ &= [f^*]^{-1}(v) - 1. && \text{(III.29)}\end{aligned}$$

where we have also used the symmetry of (III.18). ■

This theorem has various interesting consequences. First, it establishes an easy-to-verify necessary condition for the canonical form. For example, logistic regression has

$$[f^*_\phi]^{-1}(v) = \frac{1}{1+e^{-v}} \quad \text{and} \quad \phi'(v) = -\frac{e^{-v}}{1+e^{-v}} = [f^*_\phi]^{-1}(v) - 1, \qquad \text{(III.30)}$$

while for boosting

$$[f^*_\phi]^{-1}(v) = \frac{1}{1+e^{-2v}} \quad \text{and} \quad \phi'(v) = -e^{-v} \neq [f^*_\phi]^{-1}(v) - 1. \qquad \text{(III.31)}$$

This, plus the symmetry of J and f^*_ϕ, shows that the former is in canonical form but the latter is not. Second, it makes it clear that, up to additive constants, the three components (ϕ, C^*_ϕ, and f^*_ϕ) of a canonical risk are related by one-to-one relationships. Hence, it is possible to control the properties of the *three* components of the risk by manipulating a *single* function (which can be any of the three). Finally, it enables a very detailed characterization of the losses compatible with most optimal links of Table III.1.

III.D Inverse-sigmoidal links

Inspection of Table III.1 suggests that the classifiers produced by boosting, logistic regression, and variants have sigmoidal inverse links $[f^*_\phi]^{-1}$. Due to this, we refer to the links f^*_ϕ as *inverse-sigmoidal* (IS).

Definition 8. *An invertible link function $f(\eta)$ is inverse sigmoidal if its inverse, $f^{-1}(v)$, has the following properties*

1. $[f^*_\phi]^{-1}(v) \in (0,1)$
2. $[f^*_\phi]^{-1}(v)$ *is monotonically increasing*
3. $\lim_{v\rightarrow -\infty}[f^*_\phi]^{-1}(v) = 0$
4. $\lim_{v\rightarrow \infty}[f^*_\phi]^{-1}(v) = 1$
5. $\lim_{v\rightarrow \pm\infty}([f^*_\phi]^{-1})^{(n)}(v) = 0, n \geq 1,$
6. $[f^*_\phi]^{-1}(0) = .5$

where $f^{(n)}$ is the n^{th} order derivative of f.

The following theorem shows that, when the risk is in canonical form and f^*_ϕ is IS, the loss ϕ is strongly constrained.

Theorem 36. *Let $C_\phi(\eta, f)$ be a canonical risk associated with a canonical loss ϕ and optimal link f^*_ϕ. f^*_ϕ is inverse sigmoidal if and only if the loss ϕ has the following properties*

1. $\phi(v)$ *is monotonically decreasing*
2. $\phi(v)$ *is convex*
3. $\lim_{v\rightarrow -\infty} \phi'(v) = -1$
4. $\lim_{v\rightarrow \infty} \phi'(v) = 0$
5. $\lim_{v\rightarrow \pm\infty} \phi^{(n+1)}(v) = 0, n \geq 1$

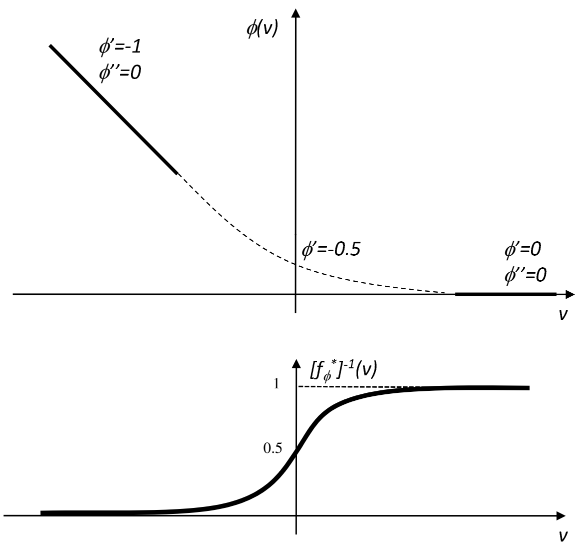


Figure III.1 Canonical losses compatible with an IS optimal link.

6. $\phi'(0) = -.5$

where $\phi^{(n)}$ *is the* n^{th} *order derivative of* ϕ*.*

Proof. The theorem follows from the definition of IS link and (III.25), from which the following equivalences can be trivially derived

$$\begin{aligned}
[f_\phi^*]^{-1}(v) \in (0,1) \quad &\Leftrightarrow \quad \phi(v) \text{ monotonically decreasing} \\
[f_\phi^*]^{-1}(v) \text{ monotonically increasing} \quad &\Leftrightarrow \quad \phi(v) \text{ convex} \\
\lim_{v\to-\infty} [f_\phi^*]^{-1}(v) = 0 \quad &\Leftrightarrow \quad \lim_{v\to-\infty} \phi^{(1)}(v) = -1 \\
\lim_{v\to\infty} [f_\phi^*]^{-1}(v) = 1 \quad &\Leftrightarrow \quad \lim_{v\to\infty} \phi^{(1)}(v) = 0 \\
\lim_{v\to\pm\infty} ([f_\phi^*]^{-1})^{(n)}(v) = 0, n \geq 1 \quad &\Leftrightarrow \quad \lim_{v\to\pm\infty} \phi^{(n+1)}(v) = 0, n \geq 1 \\
[f_\phi^*]^{-1}(0) = .5 \quad &\Leftrightarrow \quad \phi^{(1)}(0) = -.5.
\end{aligned}$$

■

The theorem shows that, as illustrated in Figure III.1, the optimal link is IS if and only if the loss $\phi(v)$ is convex, monotonically decreasing, linear (with slope -1) for large negative v, constant for large positive v, and has slope $-.5$ at the origin. The set of losses compatible with an IS link is, thus, strongly constrained. The only degrees of freedom are in the behavior of the function around the origin. This is not surprising, since the only degrees of freedom of the sigmoid itself are in its behavior within this region. What is interesting is that these are the degrees of freedom that control the margin characteristics of the loss ϕ. Hence, by controlling the behavior of the IS link around the origin, it is possible to control the margin of the optimal classifier. In particular, the margin is a decreasing function of the curvature of the loss at the origin, $\phi^{(2)}(0)$. Since, from (III.25), $\phi^{(2)}(0) = \{[f^*_\phi]^{-1}\}'(0)$, the margin can be controlled by varying the slope of $[f^*_\phi]^{-1}$ at the origin.

III.E Variable margin loss functions

In this section we use the results above to derive families of canonical losses with controllable margin.

III.E.1 Canonical boosting loss

In Section III.C, we have seen that the boosting loss is not canonical, but there is a canonical loss for the minimum risk of boosting. We consider a parametric extension of this risk,

$$J(\eta; a) = \frac{-2}{a}\sqrt{\eta(1-\eta)}, \quad a > 0. \tag{III.32}$$

From (III.23), the canonical optimal link is

$$f^*_\phi(\eta; a) = \frac{2\eta - 1}{a\sqrt{\eta(1-\eta)}} \tag{III.33}$$

and it can be shown that

$$[f^*_\phi]^{-1}(v; a) = \frac{1}{2} + \frac{av}{2\sqrt{4 + (av)^2}}, \tag{III.34}$$

which is an IS link. Using (III.21), the corresponding canonical loss can be shown to be

$$\phi(v; a) = \frac{1}{2a}(\sqrt{4 + (av)^2} - av). \tag{III.35}$$

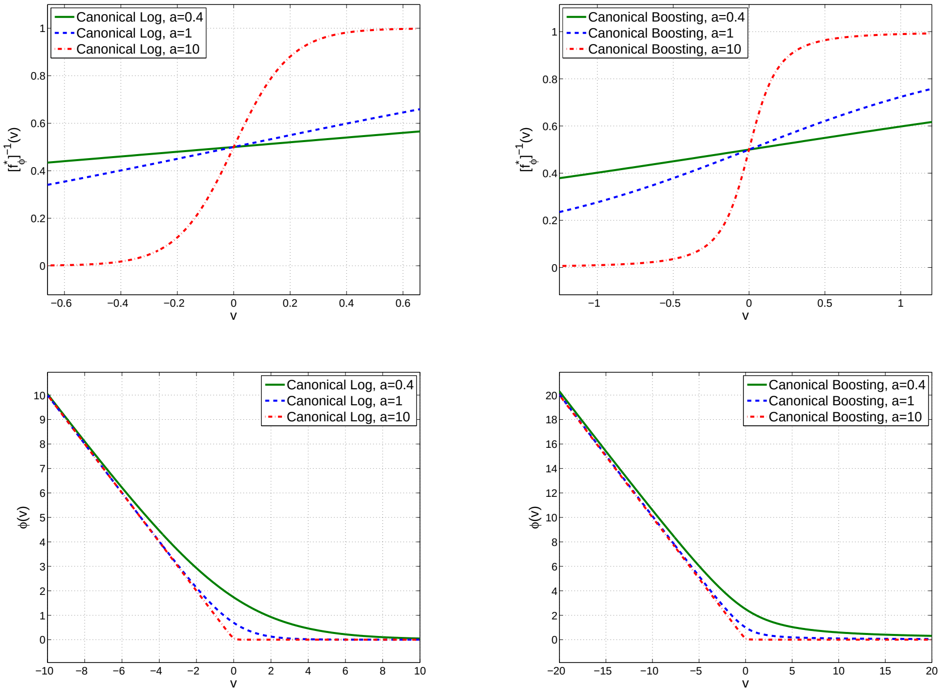


Figure III.2 Canonical link (top) and loss (bottom) for various values of a. (Left) canonical logistic, (right) canonical boosting.

See Appendix A for complete derivations. Because it shares the minimum risk of boosting, we refer to this loss as the *canonical boosting loss*. It is plotted in Figure III.2, along with the inverse link, for various values of a. Note that the inverse link is indeed sigmoidal, and that the margin is determined by a. Since $\phi^{(2)}(0; a) = \frac{a}{4}$, the margin increases with decreasing a.

III.E.2 Variable margin extensions of existing losses

It is also possible to derive variable margin extensions of existing canonical losses. For example, consider the parametric extension of the minimum risk of logistic regression

$$J(\eta; a) = \frac{1}{a}\eta \log(\eta) + \frac{1}{a}(1 - \eta)\log(1 - \eta). \qquad \text{(III.36)}$$

From (III.23),

$$[f_\phi^*](v;a) = \frac{1}{a}\log\frac{\eta}{1-\eta} \quad [f_\phi^*]^{-1}(v;a) = \frac{e^{av}}{1+e^{av}}. \tag{III.37}$$

This is again a sigmoidal inverse link and, from (III.25),

$$\phi(v;a) = \frac{1}{a}\left[\log(1+e^{av}) - av\right]. \tag{III.38}$$

See Appendix B for complete derivations. We denote this loss the *canonical logistic loss*. It is plotted in Figure III.2, along with the corresponding inverse link for various values of a. Since $\phi^{(2)}(0;a) = \frac{a}{4}$, the margin again increases with decreasing a.

Note that, in (III.35) and (III.38), margin control is not achieved by simply rescaling the domain of the loss function, e.g. just replacing $\log(1+e^{-v})$ by $\log(1+e^{-av})$ in the case of logistic regression. This would have no impact in classification accuracy, since it would just amount to a change of scale of the original feature space. While this type of re-scaling occurs in both families of loss functions above (which are both functions of av), it is localized around the origin, and only influences the margin properties of the loss. As can be seen in Figure III.2 all loss functions are identical away from the origin. Hence, varying a is conceptually similar to varying the bandwidth of an SVM kernel. This suggests that the margin parameter a could be cross-validated to achieve best performance. We will explore this possibility in Section III.G.

III.E.3 Canonical loss functions from cumulative distribution functions

One classical result in probability and statistics is that the cumulative distribution function (cdf) associated with any symmetric probability distribution function (pdf) of zero mean is a sigmoidal function. It follows from (III.25) that a canonical loss can be derived from any continuous cdf.

Corollary 37. *Let $c(v)$ be a continuous cdf associated with a symmetric pdf $p(v)$ of zero-mean. Then*

$$\phi(v) = \int [c(v) - 1]dv \tag{III.39}$$

is a canonical loss and the risk $C_\phi(\eta, f)$ has the IS optimal link $f^*_\phi(\eta)$ given by

$$[f^*_\phi]^{-1}(v) = c(v). \tag{III.40}$$

Proof. Let $[f^*_\phi]^{-1}(v) = c(v)$. Then, f^*_ϕ is IS. Use (III.25) to derive $\phi(v)$. ■

Many canonical losses with the properties of Figure III.1 can be derived from this result. Consider, for example, a Gaussian pdf with zero mean and variance a^2. The corresponding cdf is

$$c(v) = \frac{1}{2}\left[1 + erf\left(\frac{v}{\sqrt{2a^2}}\right)\right] \tag{III.41}$$

where $erf(\cdot)$ is the Gaussian error function. Application of (III.39) produces a novel Bayes consistent loss, which we denote by *canonical Gaussian loss*

$$\phi(v) \quad = \quad \frac{v}{2}\left[erf\left(\frac{v}{\sqrt{2a^2}}\right) - 1\right] + \frac{a}{\sqrt{2\pi}}e^{-\frac{v^2}{2a^2}}. \tag{III.42}$$

The canonical risk associated with this loss has optimal link

$$f^*_\phi(\eta) = \sqrt{2a^2} \cdot erf^{-1}(2\eta - 1) \tag{III.43}$$

and minimum risk

$$C^*_\phi(\eta) = -\sqrt{2a^2}\int erf^{-1}(2\eta - 1)d\eta \tag{III.44}$$

See Appendix C for complete derivations. The loss, inverse link (cdf), and underlying pdf are shown in Figure III.3, for different values of the margin parameter a (Gaussian standard deviation). Note that the margin enforced by the loss increases with this parameter.

A similar derivation can be performed for the Laplacian pdf, whose cdf is

$$c(v) = \frac{1}{2}\left[1 + sign(v)\left(1 - e^{-\frac{|v|}{a}}\right)\right]. \tag{III.45}$$

Unlike the Gaussian, the optimal link $f^*_\phi(\eta)$ and risk $C^*_\phi(\eta)$ can be derived in closed form

$$f^*_\phi(\eta) \quad = \quad -a\, sign(2\eta - 1)\log(1 - |2\eta - 1|) \tag{III.46}$$

$$C^*_\phi(\eta) \quad = \quad \frac{a}{2}(1 - |2\eta - 1|)[1 - \log(1 - |2\eta - 1|)]. \tag{III.47}$$

The *Canonical Laplacian Loss* can then be shown to be

$$\phi(v) \quad = \quad \frac{1}{2}[ae^{\frac{-|v|}{a}} + |v| - v]. \tag{III.48}$$

See Appendix D for complete derivations.

The *Canonical Laplacian Loss* is plotted, along with the corresponding inverse link (cdf) and pdf, in Figure III.3 for different values of the margin parameter a (variance of the Laplacian distribution). Note that the margin enforced by the loss increases with a.

III.E.4 New pdfs

The procedure of the previous section can be used to derive a canonical loss from any symmetric zero mean pdf. Interestingly, the reverse, i.e. that a symmetric zero-mean pdf can be derived from any canonical loss, also follows from (III.39) and (III.40). This observation can be used to derive novel pdfs from losses commonly used in machine learning. For example, we have seen that the canonical boosting loss of (III.35) has inverse link

$$[f^*_\phi]^{-1}(v) = \frac{1}{2} + \frac{av}{2\sqrt{4 + (av)^2}}. \tag{III.49}$$

Using (III.40) and taking the derivative with respect to v leads to the *canonical boosting pdf*

$$p(v) \quad = \quad \frac{2a}{(4 + (av)^2)^{\frac{3}{2}}} \tag{III.50}$$

with scaling parameter a. This pdf has some similarities to the Student-t distribution but, to the best of our knowledge, has not been presented in the literature. On the other hand, the canonical logistic loss of (III.38) has inverse link

$$[f^*_\phi]^{-1}(v) = \frac{e^{av}}{1 + e^{av}}, \tag{III.51}$$

leading to the pdf

$$p(v) = \frac{ae^{av}}{(1 + e^{av})^2} \tag{III.52}$$

of variance $\frac{\pi^2}{3a^2}$. This is the well known logistic distribution. The two pdfs are plotted in Figure III.4, for different values of parameter a.

III.F Variable margin boosting algorithms

Given a Bayes consistent loss function, a number of algorithms can be used to minimize the associated empirical risk, and design a classifier. Boosting algorithms accomplish this by gradient descent in the functional space spanned by a set of weak learners. While there are many variants, in this chapter we adopt the GradientBoost algorithm of [36]. This algorithm is especially attractive when the goal is to compare losses, since the implementations derived with different losses differ 1) uniquely and 2) explicitly in loss-dependent parameters. This is not the case for all boosting procedures. GradientBoost is a gradient descent procedure for determining the predictor $F(\mathbf{x})$ that minimizes the empirical risk on a training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$,

$$R(F) = \sum_{n=1}^{N} \phi(y_n F(\mathbf{x}_n)). \tag{III.53}$$

The gradient direction at iteration t is given by the weak learner $f_t(x)$ that satisfies

$$\begin{aligned} f_t(x) &= \arg\max_f \sum_{n=1}^{N} -y_n \phi'(y_n F_{t-1}(\mathbf{x}_n)) f(\mathbf{x}_n) && \text{(III.54)} \\ &= \arg\max_f \sum_{n=1}^{N} y_n w_t(\mathbf{x}_n) f(\mathbf{x}_n) && \text{(III.55)} \end{aligned}$$

where

$$w_t(\mathbf{x}_n) = -\phi'(y_n F_{t-1}(\mathbf{x}_n)) \tag{III.56}$$

is the weight of example $\mathbf{x}_n$ at iteration t. For canonical losses, it follows from (III.25) that

$$w_t(\mathbf{x}_n) = 1 - [f^*_\phi]^{-1}(y_n F_{t-1}(\mathbf{x}_n)). \tag{III.57}$$

We refer to the algorithm with these weights as *canonical gradientBoost*. It is summarized in Algorithm 1.

Canonical gradientBoost has the interesting property of not requiring the evaluation of the loss $\phi(v)$. Instead, it only requires evaluation of the optimal

Algorithm 1 Canonical GradientBoost

```
**Input:** Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y_i \in \{1, -1\}$ is the class label of example $\mathbf{x}$, and number $M$ of weak learners in the final decision rule.
**Initialization:** Set $F^{(0)}(\mathbf{x}_i) = 0$ and $w^{(1)}(\mathbf{x}_i) = 1 - [f^*_\phi]^{-1}(y_i F^{(0)}(\mathbf{x}_i))\ \ \forall \mathbf{x}_i$ .
**for** $m = \{1, \ldots, M\}$ **do**
    choose weak learner
        $f(\mathbf{x}) = \arg\max_{f(\mathbf{x})} \sum_{i=1}^{n} y_i w^{(m)}(\mathbf{x}_i) f(\mathbf{x}_i)$    (III.58)
    update predictor F($\mathbf{x}$)
        $F^{(m)}(\mathbf{x}) = F^{(m-1)}(\mathbf{x}) + f(\mathbf{x})$    (III.59)
    update weights w($\mathbf{x}$)
        $w^{(m+1)}(\mathbf{x}_i) = 1 - [f^*_\phi]^{-1}(y_i F^{(m)}(\mathbf{x}_i))\ \ \forall \mathbf{x}_i$    (III.60)
**end for**
**Output:** decision rule $h(\mathbf{x}) = \text{sgn}[F^{(M)}(\mathbf{x})]$.
```

inverse link $[f^*_\phi]^{-1}(v)$. Hence, it can be implemented with any IS link function, e.g. any cdf, independently of whether the integration of (III.39) is tractable or not. This has some interesting consequences. First, since many cdfs are non-trivial to integrate, it leads to a much larger number of variants on the algorithm than would be possible if the loss had to be computed. Second, it provides some assurances with respect to robustness of the resulting algorithms. Robustness is a concern for large-margin algorithms since, as illustrated in Figure III.1, the losses $\phi(yf)$ tend to be unbounded for negative margins, i.e.

$$\lim_{yf \to -\infty} \phi(yf) = \infty. \tag{III.61}$$

This is particularly problematic for algorithms such as AdaBoost [53, 25, 60, 57], which are derived from a loss, $\phi(yf) = e^{-yf}$, which grows exponentially with the negative margin. It has long been known that algorithms such as LogitBoost [35, 61, 49] have much less sensitivity to outliers, a fact that has been attributed to the linear increase of the loss with the negative margin (as shown in Figure III.1). The

Table III.2 Weighting function of the gradientBoost algorithm for different canonical losses.

Loss	$w(x) = -\phi'(yF(x))$
Canonical Logistic	$\frac{e^{-ayF(x)}}{1+e^{-ayF(x)}}$
Canonical Boosting	$\frac{1}{2}\frac{(1-(ayF(x)))}{\sqrt{4+(ayF(x))^2}}$
Canonical Gaussian	$\frac{1}{2} - \frac{1}{2}erf(\frac{yF(x)}{\sqrt{2a^2}})$
Canonical Laplacian	$\frac{1}{2} - \frac{1}{2}sign(yF(x))(1 - e^{\frac{-\lvert yF(x)\rvert}{a}})$

discussion of the previous sections shows that this holds for *all* canonical losses. The connection between robustness and loss is made more explicit by the derivation of canonical gradientBoost. Note that, for non-canonical losses, the weights are given by (III.56). For the exponential loss of AdaBoost $\phi'(v) = -\phi(v)$ and

$$w_t(\mathbf{x}_n) = e^{-y_n F_{t-1}(\mathbf{x}_n)}, \tag{III.62}$$

i.e. outliers of large negative margin have a dominant weight in classifier design. On the other hand, for canonical losses, the weights have the form of (III.57), i.e. one minus a sigmoid. Hence, the weight function saturates for small negative margins, and potential outliers are assigned the same weight as most other misclassified examples. This is illustrated in Figure III.5, which shows the weight functions for a number of canonical losses. The weight functions themselves are presented in Table III.2. Again, the margin parameter a controls the behavior of the weights in the neighborhood of the classification boundary, determining the extent of the region of correctly classified examples that receive non-zero weight, i.e. the margin.

III.G Experiments

Various experiments were conducted to validate the theoretical results of the previous sections. They were divided in three main groups. The first aimed to determine how the importance of controlling the classification margin varied with training set size. The second, based on 10 UCI datasets of relatively small size, aimed to compare the performance of the different losses studied above. Finally, the third set relied on a larger and higher dimensional UCI dataset, and aimed to compare the performance of the variable-margin losses against popular losses in the literature.

III.G.1 Experiments on two Gaussian classes

All Bayes consistent losses produce classifiers that converge asymptotically to the Bayes decision rule. Hence, all losses discussed above should have identical performance as training sets increase. On the other hand, since generalization is most important for small training sets, the margin parameter a should have the greatest effect on classification error in this scenario. By measuring the classification performance obtained with different values of a, as a function of training set size, it is possible to test this hypothesis explicitly. For this, we designed a number of experiments involving a simple classification problem composed of two Gaussian classes of identity covariance, $\mathbf{\Sigma} = \mathbf{I}$, on a two-dimensional space. The means were set to $(0, 0)$ and $(0.7416, 0.7416)$, so as to produce a problem with a Bayes error of 30%. Classifiers were learned with training sets of n examples per class, where $n \in \{10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000, 5000, 10000\}$, and evaluated with a test set of 10000 examples. All classifiers were learned with canonical gradientBoost, with histogram-based weak learners, for both the canonical logistic and boosting losses, and 19 values of the margin parameter a, $a \in \{0.1, 0.2, ..., 0.9, 1, 2, ..., 10\}$. 50 iterations of boosting were applied to each training set.

Figure III.6 presents a plot of the average classification error, and its standard deviation, as a function of the training set size. The average is taken over the 19 values of the margin parameter, and 100 repetitions of the entire experiment. As expected, the average error decreases as the training set size increases. While none of the losses produces an error of 30%, the error approaches this lower bound for the largest training sets. In this regard, the canonical logistic loss performs somewhat better than the canonical boosting loss. More interestingly, the standard deviation of the error also decreases with the increase in training set size. In fact, it is close to zero even for relatively small training sets ($100 - 200$ examples per class). While this was not unexpected, given the simplicity of the problem, it is clear that the classification performance is much more sensitive to the value of the margin parameter for small training sets. For example, in the case of the canonical logistic loss, a variation of as much as 2% in test error can result from varying the margin parameter, for the smallest datasets considered (10 samples per class). These results support the hypothesis that the margin parameter is most relevant when the training data is scarce.

III.G.2 Experiments on small size high dimensional UCI datasets

We next performed a number of experiments that tested the importance of controlling the loss shape and margin parameters. Given the findings of the previous section, and to enable extensive comparisons, we selected ten binary UCI data sets of relatively small size: (#1) sonar, (#2) breast cancer prognostic, (#3) breast cancer diagnostic, (#4) original Wisconsin breast cancer, (#5) Cleveland heart disease, (#6) tic-tac-toe, (#7) echo-cardiogram, (#8) Haberman's survival, (#9) Pima-diabetes, and (#10) liver disorder. The data was split into five folds, four used for training and one for testing. This produced five training-test pairs per dataset. The GradientBoost algorithm with histogram-based weak learners was then used to design boosted classifiers which minimize all canonical losses discussed in the previous sections, for various margin parameters. 50 boosting iterations were applied to each training set, for 19 values of $a \in \{0.1, 0.2, ..., 0.9, 1, 2, ..., 10\}$. The classification accuracy was then computed per dataset, by averaging over its five train/test pairs.

Since popular algorithms in the literature, such as LogitBoost, are special cases of the proposed losses, with $a = 1$, it is natural to inquire whether other values of the margin parameter will achieve best performance. This question is addressed by Figure III.7, which presents the average rank of the classifier designed with each loss and margin parameter a. To produce the plot, a classifier was trained on each dataset, for all 19 values of a. The results were then ranked, with rank 1 (19) being assigned to the a parameter of smallest (largest) error. The ranks achieved with each a were then averaged over the ten datasets, as suggested in [24]. For the canonical logistic loss, the best values of a are in the range $0.2 \leq a \leq 0.3$. Note that the average rank for this range (between 5 and 6), is better than that (close to 7) achieved with the logistic loss of LogitBoost [35] ($a = 1$). In fact, as can be seen from Table III.3, the canonical logistic loss with $a = 1$ did not achieve rank 1 on any dataset, whereas canonical logistic losses with $0.2 \leq a \leq 0.3$ were top ranked on 3 datasets (and with $0.1 \leq a \leq 0.4$ on 6). For the canonical boosting loss, there is also a range ($0.8 \leq a \leq 2$) that produces best results. For the canonical Gaussian loss the best values of a are in the range $6 \leq a \leq 10$, and for the canonical Laplacian loss in the range $7 \leq a \leq 10$. Note that in the case of the canonical Gaussian and Laplacian losses, larger values of a correspond to larger margins. This is in contrast to the canonical logistic and boosting losses, where large margins are associated with smaller values of a. In general, the a values of different losses are not directly comparable. This can be seen from Figure III.2 where, even when the comparison is restricted to the canonical logistic and boosting losses, $a = 0.4$ produces a loss of much larger margin for the latter. Furthermore, the canonical boosting loss has a heavier tail and approaches zero more slowly than the canonical logistic loss. In any case, Figure III.7 shows that all canonical losses display better performance at larger margins. This is sensible, given the relatively small dataset sizes.

Although certain ranges of margin parameters seem to produce best results for all canonical loss functions, the optimal parameter value is likely to be dataset dependent. This is confirmed by Table III.3 which presents the parameter value of rank 1 for each of the ten datasets. Improved performance should thus be possible by cross-validating the margin parameter a. Table III.4 presents the

Table III.3 Value of the margin parameter a of rank 1, on each of the ten UCI datasets.

UCI dataset#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Canonical Log	0.4	0.5	0.6	0.3	0.1	2	0.5	0.1	0.2	0.2
Canonical Boost	0.9	6	2	2	0.4	3	0.2	4	0.2	0.9
Canonical Gauss	1	0.6	6	10	8	0.7	10	0.7	10	8
Canonical Laplace	0.7	0.3	9	5	0.8	0.6	10	10	8	8

Table III.4 Cross validated classification error for each loss function and UCI dataset.

UCI dataset#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Canonical Log	**11.2**	**11.4**	**8**	5.6	**12.4**	11.8	7	18.8	38.2	**27**
LogitBoost ($a = 1$)	11.6	12.4	10	6.6	13.4	48.6	6.8	21.2	39.6	28.4
Canonical Boost	12.6	11.6	21	18.6	17.6	7.2	**6**	21.8	**37.6**	28.6
Canonical Boost, $a = 1$	13.2	12.4	21	18.6	18.6	50.8	7.2	21.2	39.4	28.2
Canonical Gauss	13.6	14	9	6.4	13	10.2	6.8	**18.4**	38.8	29.8
Canonical Gauss, $a = 1$	14.8	15	9.2	6	14	21	7	21.2	40.2	29.6
Canonical Laplace	12	15	9	**4.2**	**12.4**	**6.6**	7.4	19.2	40.4	31.4
Canonical Laplace, $a = 1$	13	13.6	11.4	4.8	13.4	34.8	6.8	21	40	29.4
AdaBoost	11.4	**11.4**	9.4	6.4	14	28	6.6	21.8	41.2	28.2

5-fold cross validation test error (# of misclassified points) obtained for each UCI dataset and canonical loss. The table also shows the results of AdaBoost, LogitBoost (canonical logistic, $a = 1$), and other canonical losses with $a = 1$. When compared to the fixed margin ($a = 1$) counterpart, cross validating the margin results in better performance for 9 out of 10 datasets for the canonical logistic loss, 8 out of 10 datasets for the canonical boosting and Gaussian losses, and 6 out of 10 datasets for the canonical Laplacian loss. When compared to the existing algorithms, at least one of the margin-tunned classifiers is better than both Logit and AdaBoost for each dataset. For several datasets (#3,#4, #5, #6, #8, and #9) this holds for at least three of the four margin-tunned classifiers. In dataset #6 (tic-tac-toe) the error of the worst margin-tunned classifier (canonical logistic loss, 11.8) is less than half that of AdaBoost (28) and four times smaller than

Table III.5 Classification error for each loss function and UCI dataset.

UCI dataset#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Canonical Log, $a = 0.2$	13.2	15	8.4	5	11.2	56.2	6.8	24	39.8	**25.8**
Canonical Boost, $a = 0.2$	12.6	**14.8**	17.2	18.6	12	56.8	6.8	**23.2**	**38.4**	26.4
Canonical Gaussian, $a = 8$	13.4	17.4	7.4	**4.6**	**11**	56.4	**6.6**	24	40	26.6
Canonical Laplacian, $a = 8$	13.8	17.2	**6**	5.2	**11**	56.4	**6.6**	**23.2**	38.6	26.4
LogitBoost ($a = 1$)	12.4	15.4	8.6	5.6	11.4	46	7.2	25	40.4	26.4
AdaBoost	**11.4**	15.2	9.2	6	11.4	**21.6**	7.4	**23.2**	42.8	26.6

that of LogitBoost (48.9). The best of the margin-tunned classifiers (canonical Laplace loss) further halves this error (6.6%). In summary, cross-validation of the margin parameter leads to substantial improvements over the standard boosting algorithms.

While, computationally, cross-validation of the margin parameter is not different from, say, cross-validating the bandwidth of an SVM kernel, it may not be possible or computationally feasible for some applications. Even in this case, it may be better to use a value of a other than the standard $a = 1$. Table III.5 presents results for the case where the margin parameter is fixed at $a = 0.2$ for the canonical logistic and boosting losses and $a = 8$ for the canonical Gaussian and Laplacian losses. Even in this case, there is at least one canonical loss that outperforms LogitBoost on each dataset and at least one canonical loss which outperforms AdaBoost in 7 of the ten datasets. Furthermore, canonical logistic and Laplacian outperform *both* LogitBoost and AdaBoost in 7 of the ten datasets, and canonical boosting and Gaussian in 5 of the ten datasets. The converse, i.e. AdaBoost or LogitBoost outperforming all canonical losses only happens in 2 datasets. In summary, even without the benefit of cross-validation, it is possible to find values of the margin parameter for which the performance of the margin-tunned classifiers is better than those learned with the standard boosting algorithms.

III.G.3 Experiments on large scale datasets

To confirm the results of the previous section on experiments of larger scale, we considered the ADULT, LETTER.p1 and LETTER.p2 datasets, which are widely used for comparing ensemble methods [71, 18]. Missing values in the ADULT training and testing sets were omitted, leading to 30,162 training examples, of which 7,508 are positive and 22,654 negative. The testing set consists of 15,060 examples, of which 3,700 are positive and 11,360 negative. The LETTER data set was converted into two binary datasets, following the method outlined in [18]. The LETTTER.p1 dataset treats the confusable letter "O" as the positive class, and the remaining 25 letters of the alphabet as the negative class, resulting in a highly unbalanced classification problem. LETTER.p2 uses the first 13 letters of the alphabet as the negative class and the last 13 as the positive class, resulting in a balanced but difficult problem. Both datasets consist of 4,000 training and 16,000 test examples. As before, all classifiers were learned with gradientBoost, using histogram weak learners. The performance of the canonical logistic and Laplacian losses were compared against that of the exponential loss, used by AdaBoost. Each boosting algorithm was run for 100 iterations.

Table III.6 presents the smallest error achieved by each method, and the corresponding margin parameter value. It can be seen that 1) the best performance is never attained when the canonical logistic loss uses $a = 1$ (LogitBoost), 2) the performance achieved with the best margin parameter can be substantially superior to those of *both* AdaBoost and LogitBoost, and 3) both canonical losses outperform the two standard boosting algorithms. This is in agreement with the findings of the previous section. The case where a reduced number of training points are available was also considered, by randomly subsampling the LETTER.p2 dataset by a factor of 2 (DIV2) and 4 (DIV4). The size of the test set was not changed. Table III.7 presents the maximum difference between the number of testing errors produced by the exponential and variable margin losses, for each training set size. For example, in the original training set (DIV1), canonical logistic produces 2831

Table III.6 Smallest errors, and corresponding values of the margin parameter a, for the three losses considered.

UCI dataset	ADULT	LETTER1	LETTER2
Canonical Log	**2406** $a = 4$	427 $a = 3$	**2831** $a = 2$
Canonical Laplacian	2680 $a = 0.9$	**411** $a = 0.4$	2844 $a = 0.6$
Exponential	2696	529	2940
LogitBoost ($a = 1$)	2673	464	2867

Table III.7 Maximum difference between the exponential and canonical loss testing errors, and corresponding values of the margin parameter a, for various fractions of the LETTER.p2 training data.

LETTER.p2	DIV1	DIV2	DIV4
Canonical Log	109 $a = 2$	179 $a = 0.6$	260 $a = 0.5$
Canonical Laplacian	96 $a = 0.6$	178 $a = 2$	186 $a = 4$

and exponential 2940 errors (see Table III.6). Hence, the difference is 109 errors, as reported in the table. Overall, the table confirms that 1) the benefits of tuning the margin are larger for smaller training sets, and 2) they are obtained with values of a that enforce larger margins.

III.H Summary and discussion

The probability elicitation approach to loss function design, introduced in Chapter II, enables the derivation of new Bayes consistent loss functions. Yet, because the procedure is not fully constructive, this requires trial and error. In general, it is difficult to anticipate the properties, and shape, of a loss function that results from combining a certain minimal risk with a certain link function. In this chapter, we have addressed this problem for the class of canonical risks. We have shown that the associated canonical loss functions lend themselves to analysis, due to a simple connection between the associated minimum conditional risk and optimal link functions. This analysis was shown to enable a precise characterization of 1) the relationships between loss, optimal link, and minimum risk, and 2) the properties of the loss whenever the optimal link is in the family of inverse sigmoid functions. A number of approaches to the derivation of canonical loss functions with *explicit control* of the classification margin was then introduced: 1) variable margin extensions of existing losses, 2) derivation of new losses from the minimum risks associated with existing non-canonical losses, and 3) derivation of new losses from cumulative distribution functions. These possibilities were exploited to design four parametric families of loss functions with explicit margin control. The design of boosting algorithms based on canonical losses was also considered. Starting from the gradientBoost framework, it was shown that the choice of loss only impacts the weight mechanism of the resulting boosting algorithm, which was itself shown to be sigmoidal. This has two interesting consequences. First, it establishes a common boosting framework, canonical gradientBoost, for all canonical losses, which enables a direct comparison of the impact of the loss on classifier performance. Second, it guarantees that all canonical gradientBoost algorithms have some robustness to outliers. A number of experiments were conducted to verify these properties, and study the effect of margin-control on the classification accuracy of the four proposed variable-margin losses. These were shown to outperform the fixed-margin counterparts used by existing algorithms, such as AdaBoost and LogitBoost.

III.I Acknowledgments

The text of Chapter III, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “Variable Margin Bayes Consistent Losses for Classifier Design” , submitted for publication to *Journal of Machine*

Learning Research, 2010. The dissertation author was a primary researcher and an author of the cited material.

III.J Appendix

III.J.1 Appendix A: derivation of canonical boosting loss

Consider the parametric extension of the minimum risk of boosting.

$$J(\eta; a) = \frac{-2}{a}\sqrt{\eta(1-\eta)}, \quad a > 0. \tag{III.63}$$

From (III.23)

$$f^*_\phi(\eta; a) = \frac{2\eta - 1}{a\sqrt{\eta(1-\eta)}} \tag{III.64}$$

and, from (III.21) and (III.23),

$$\begin{aligned}\phi(v; a) &= -J\{[f^*_\phi]^{-1}(v)\} - (1 - [f^*_\phi]^{-1}(v))v \\ &= \frac{2}{a}\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]} - [1 - [f^*_\phi]^{-1}(v)]v\end{aligned}$$

Using $\eta = [f^*_\phi]^{-1}(v)$ in both sides of (III.64),

$$v = \frac{2[f^*_\phi]^{-1}(v) - 1}{a\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \tag{III.65}$$

and

$$\begin{aligned}\phi(v; a) &= \frac{2}{a}\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]} - \frac{[1 - [f^*_\phi]^{-1}(v)][2[f^*_\phi]^{-1}(v) - 1]}{a\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \\ &= \frac{2[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)] - [1 - [f^*_\phi]^{-1}(v)][2[f^*_\phi]^{-1}(v) - 1]}{a\sqrt{[f^*_\phi]^{-1}(v)[1 - [f^*_\phi]^{-1}(v)]}} \\ &= \frac{1}{a}\sqrt{\frac{1 - [f^*_\phi]^{-1}(v)}{[f^*_\phi]^{-1}(v)}} \end{aligned} \tag{III.66}$$

Finally, solving (III.65) for $[f^*_\phi]^{-1}(v)$,

$$[f^*_\phi]^{-1}(v; a) = \frac{1}{2} \pm \frac{1}{2}\frac{av}{\sqrt{4 + (av)^2}}.$$

Of the two solutions, one is monotonically increasing (+ in between the two terms) with v, and the other decreasing ($-$). Enforcing the constraint of an increasing link function leads to

$$[f^*_\phi]^{-1}(v;a) = \frac{1}{2} + \frac{1}{2}\frac{av}{\sqrt{4+(av)^2}},$$

and

$$\phi(v;a) = \frac{1}{a}\sqrt{\frac{\sqrt{4+(av)^2}-av}{\sqrt{4+(av)^2}+av}} = \frac{1}{2a}(\sqrt{4+(av)^2}-av).$$

III.J.2 Appendix B: derivation of canonical logistic loss

Consider the parametric extension of the minimum risk of logistic regression

$$J(\eta;a) = \frac{1}{a}\eta\log(\eta) + \frac{1}{a}(1-\eta)\log(1-\eta). \qquad \text{(III.67)}$$

From (III.23),

$$\begin{aligned} [f^*_\phi](v;a) &= \frac{1}{a}\log\frac{\eta}{1-\eta} && \text{(III.68)}\\ [f^*_\phi]^{-1}(v;a) &= \frac{e^{av}}{1+e^{av}}. && \text{(III.69)} \end{aligned}$$

From (III.26),

$$\begin{aligned} \phi(v;a) &= -J\{[f^*_\phi]^{-1}(v;a)\} - (1-[f^*_\phi]^{-1}(v;a))v \\ &= -\frac{1}{a}(\frac{e^{av}}{1+e^{av}})\log(\frac{e^{av}}{1+e^{av}}) \\ &-\frac{1}{a}(1-\frac{e^{av}}{1+e^{av}})\log(1-\frac{e^{av}}{1+e^{av}}) - (1-\frac{e^{av}}{1+e^{av}})v \\ &= \frac{1}{a}[\log(1+e^{av})-av] && \text{(III.70)} \end{aligned}$$

III.J.3 Appendix C: derivation of canonical Gaussian loss

Consider a Gaussian pdf with zero mean and variance a^2. The corresponding cdf is

$$c(v) = \frac{1}{2}\left[1+erf\left(\frac{v}{\sqrt{2a^2}}\right)\right] \qquad \text{(III.71)}$$

where $erf(\cdot)$ is the Gaussian error function. Application of (III.39) produces the canonical Gaussian loss

$$\begin{aligned}\phi(v) &= \int \left[\frac{1}{2}\left[1+erf\left(\frac{v}{\sqrt{2a^2}}\right)\right]-1\right]dv \\ &= \frac{\sqrt{2a^2}}{2}\left[\frac{v}{\sqrt{2a^2}}erf(\frac{v}{\sqrt{2a^2}})+\frac{1}{\sqrt{\pi}}e^{-(\frac{v}{\sqrt{2a^2}})^2}\right]-\frac{1}{2}v \\ &= \frac{v}{2}\left[erf\left(\frac{v}{\sqrt{2a^2}}\right)-1\right]+\frac{a}{\sqrt{2\pi}}e^{-\frac{v^2}{2a^2}}. \end{aligned} \tag{III.72}$$

The canonical risk associated with this loss has optimal link

$$f^*_\phi(\eta) = c^{-1}(\eta) = \sqrt{2a^2}\cdot erf^{-1}(2\eta-1). \tag{III.73}$$

The minimum risk can be found directly from (III.23)

$$C^*_\phi(\eta) = -\int f^*_\phi(\eta)d\eta = -\sqrt{2a^2}\int erf^{-1}(2\eta-1)d\eta. \tag{III.74}$$

III.J.4 Appendix D: derivation of canonical Laplacian loss

The Laplacian cdf (inverse link function) is

$$c(v) = [f^*_\phi]^{-1}(v;a) = \frac{1}{2}\left[1+sign(v)\left(1-e^{-\frac{|v|}{a}}\right)\right]. \tag{III.75}$$

Unlike the Gaussian, the optimal link $f^*_\phi(\eta)$ and risk $C^*_\phi(\eta)$ can be derived in closed form

$$f^*_\phi(\eta) = \begin{cases} a\log(2\eta) & \text{if } \eta < 0.5 \\ -a\log(-2\eta+2) & \text{if } \eta \geq 0.5 \end{cases} \tag{III.76}$$

$$= -a\, sign(2\eta-1)\log(1-|2\eta-1|) \tag{III.77}$$

$$C^*_\phi(\eta) = \int f(\eta)d\eta$$

$$= \begin{cases} a\left[\eta\log(2\eta)-\eta\right] & \text{if } \eta < 0.5 \\ -a\left[(\eta-1)\log(-2\eta+2)-(\eta-1)\right] & \text{if } \eta \geq 0.5 \end{cases} \tag{III.78}$$

$$= \frac{a}{2}(1-|2\eta-1|)[1-\log(1-|2\eta-1|)]. \tag{III.79}$$

The canonical Laplacian loss can be derived from (III.21), (III.77) and (III.79), as

$$\begin{aligned}
&\phi(v) = \\
&-\frac{a}{2}[(1-|2A(v)-1|)(\log(1-|2A(v)-1|)-1)] - (1-A(v))v \\
&= -\frac{a}{2}[(e^{\frac{-|v|}{a}})(\log(e^{\frac{-|v|}{a}})-1)] - (\frac{v}{2} - \frac{v}{2}sign(v) + \frac{v}{2}sign(v)e^{\frac{-|v|}{a}}) \\
&= (\frac{|v|}{2}e^{\frac{-|v|}{a}} + \frac{a}{2}e^{\frac{-|v|}{a}}) - (\frac{v}{2} - \frac{|v|}{2} + \frac{|v|}{2}e^{\frac{-|v|}{a}}) \\
&= \frac{1}{2}[ae^{\frac{-|v|}{a}} + |v| - v]. \qquad \text{(III.80)}
\end{aligned}$$

where $A(v) = c(v) = \frac{1}{2}\left[1 + sign(v)\left(1 - e^{-\frac{|v|}{a}}\right)\right]$ and we have used the equality

$$1 - |2A(v)-1| = 1 - |sing(v)||1-e^{-\frac{|v|}{a}}| = 1 - |1-e^{-\frac{|v|}{a}}| = e^{-\frac{|v|}{a}}. \qquad \text{(III.81)}$$

Alternatively, the loss can be derived from (III.39) and (III.75) as

$$\begin{aligned}
\phi(v) &= \int [c(v)-1]dv = \int \frac{1}{2}[sign(v)(1-e^{\frac{-|v|}{a}})-1]dv \\
&= \begin{cases} \frac{a}{2}e^{\frac{v}{a}} - v & \text{if } v < 0; \\ \frac{a}{2}e^{\frac{-v}{a}} & \text{if } v \geq 0. \end{cases} \\
&= \frac{1}{2}[ae^{\frac{-|v|}{a}} + |v| - v]. \qquad \text{(III.82)}
\end{aligned}$$

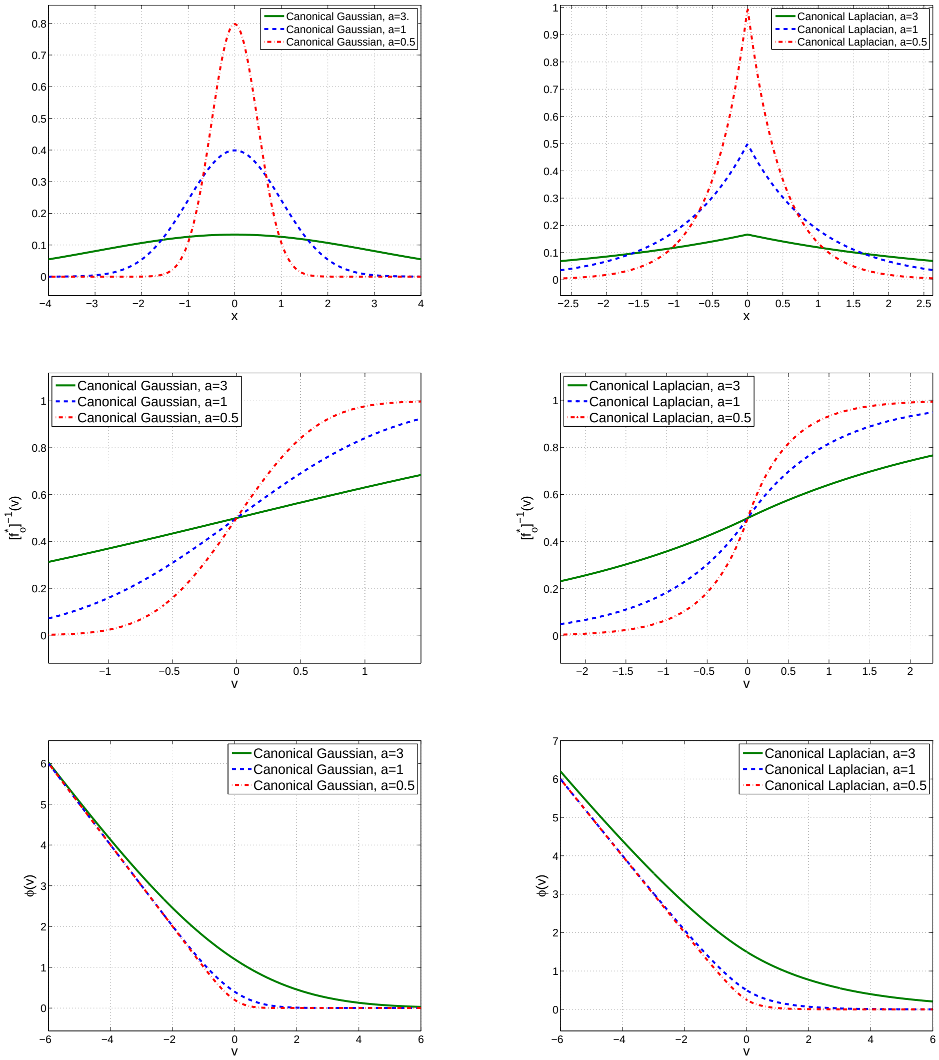


Figure III.3 Canonical pdf (top), link (middle), and loss (bottom) for various values of a. (Left) canonical Gaussian, (right) canonical Laplacian.

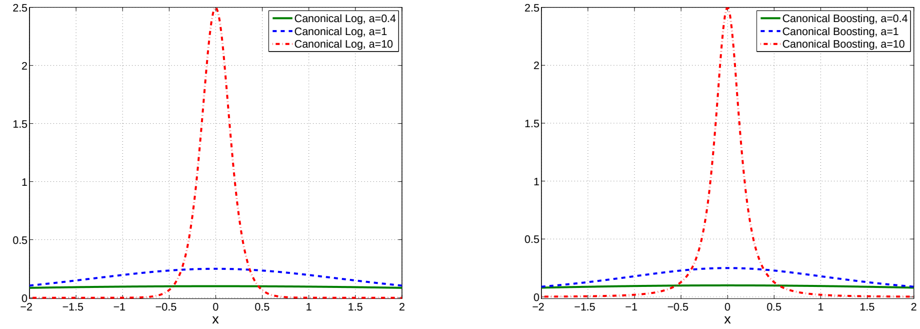


Figure III.4 Canonical logistic (Left) and boosting (right) pdfs, for various values of a .

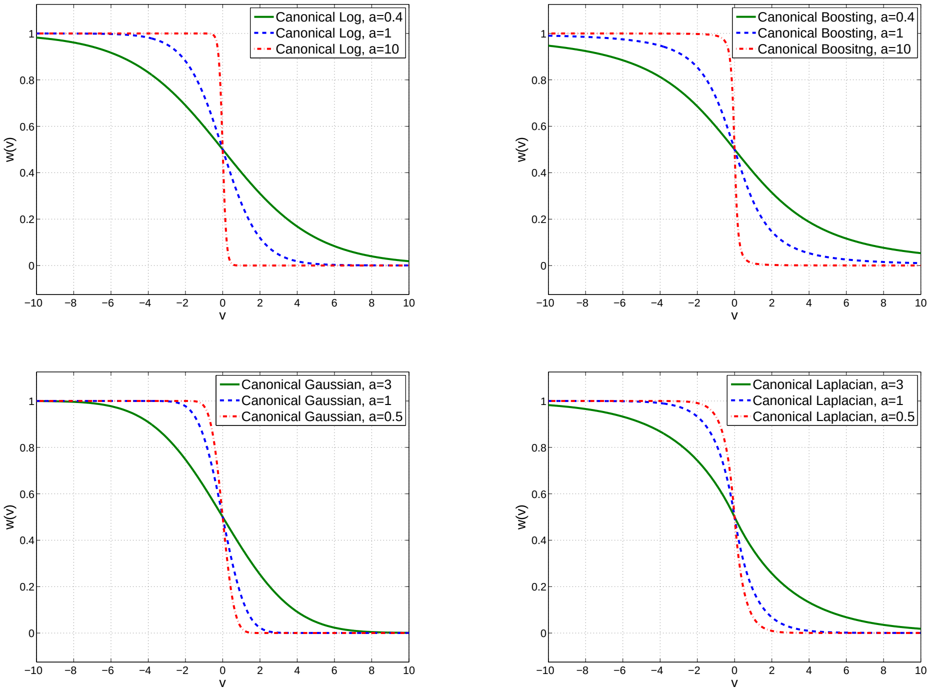


Figure III.5 Weighting function of gradientBoost for various values of a. (Top-left) canonical Logistic, (top-right) canonical Boosting, (bottom-left) canonical Gaussian and (bottom-right) canonical Laplacian.

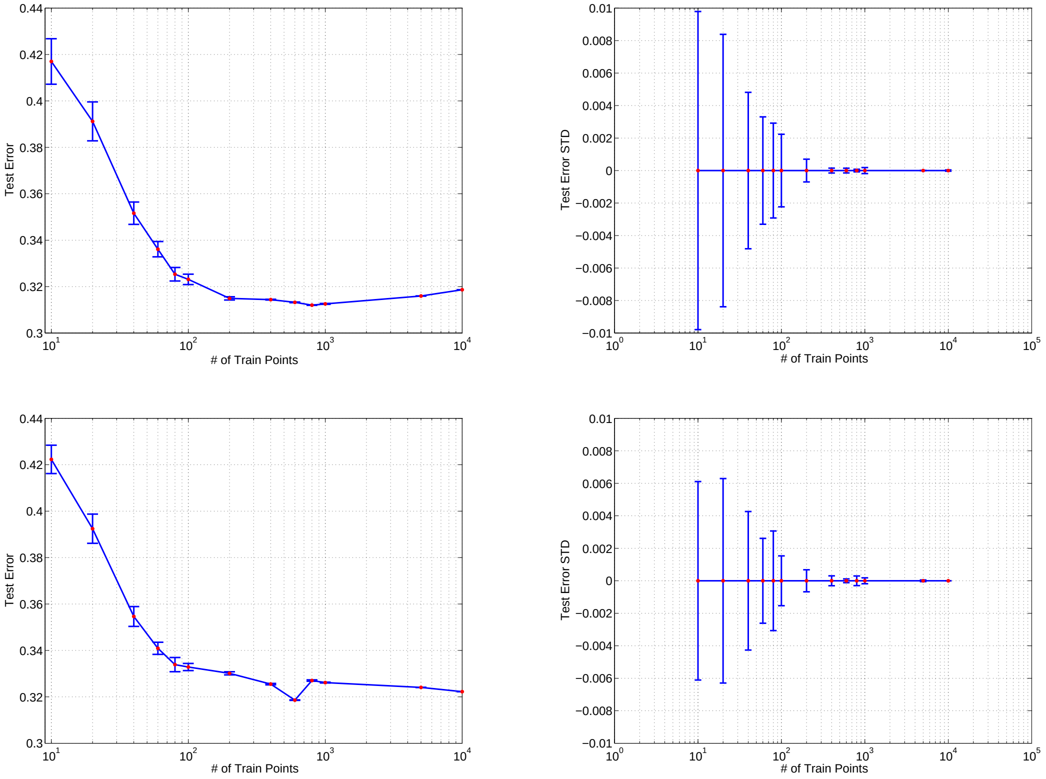


Figure III.6 Average (left) and standard deviation (right), across margin sizes, of the classification error as a function of training set size. (Top) canonical logistic loss. (Bottom) canonical boosting loss.

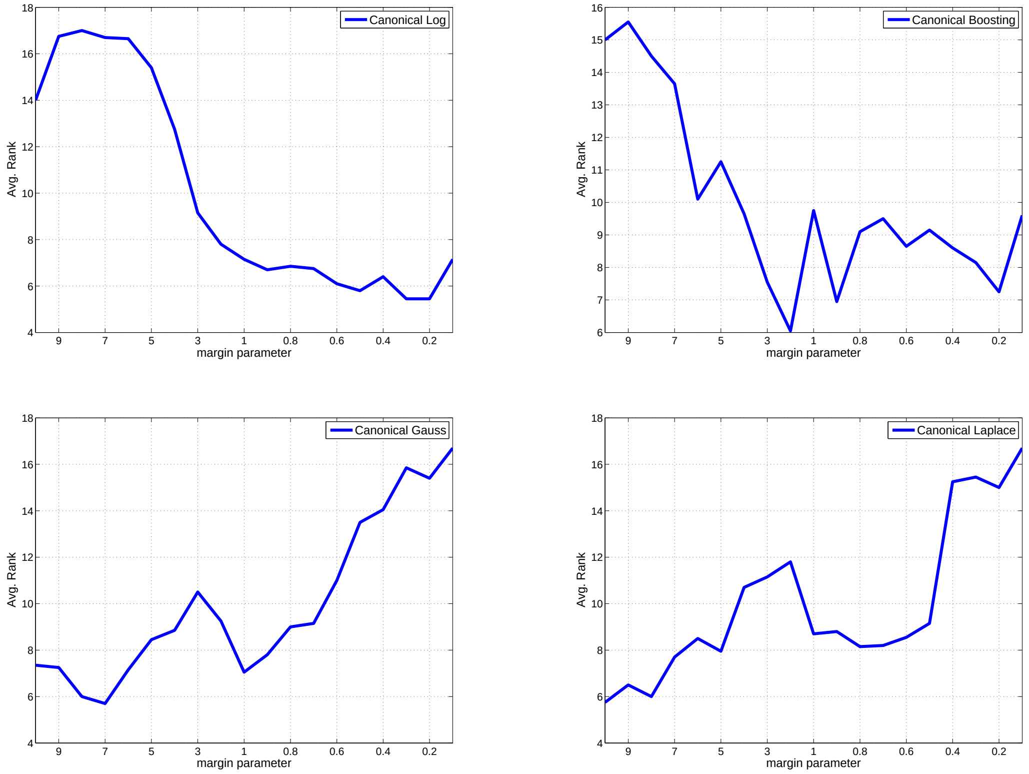


Figure III.7 Average classification rank as a function of margin parameter, on the UCI data.

Chapter IV

Robust loss function design

IV.A Introduction

Over the last decade, tremendous advances have been achieved in computer vision tasks that can be formulated as classification problems. Examples include object detection [105] and recognition [99], object tracking [7], image classification and retrieval [78, 77], among others. Much of this progress is due to the widespread adoption of classification techniques, such as the support vector machine (SVM) [98], boosting [33], or logistic regression [35], which minimize the expected value of a *margin enforcing* loss. Such losses, see Figure IV.1 for examples, apply a large penalty to points with large *negative margin* (i.e. incorrectly classified and far from the boundary), some penalty to points of small *positive margin* (correctly classified but close to the boundary), and zero penalty to points of large positive margin (correctly classified and far from the boundary). The assignment of non-zero loss to correct classifications close to the boundary is critical to assuring a classifier of maximal margin. This, in turn, is critical to guarantee good generalization [98].

While the positive impact of large margin classifiers is indisputable, they do not overcome all challenges posed by computer vision. This is due to the prevalence, in most vision applications, of noise, outliers, ambiguity, lack of labels, small training sizes, and imbalance of positive/negative coverage by training sets. For example, patch-based image classification usually involves much more negative than positive examples per class, and is inherently outlier ridden: an image from the *buildings* class invariably contains patches from the *people*, *garden*, or *car* class [55]. Furthermore, patches are inherently ambiguous (e.g. the same circular shape could correspond to a car wheel or a boat window) [78], and “noise” is plentiful (in the form of shadows, occlusions, perspective distortions, etc.). In applications such as tracking, where a classifier is incrementally learned from data (as it is being classified), it is impossible to guarantee that there is no leakage between the sets of positives and negatives used for training [7, 104, 54, 8]. While some of these problems can be mitigated by careful human labeling, human labeled data can itself be error prone. In large-scale problems, where labeling is expensive, there is frequently a need to resort to unlabeled datasets, or labels of low-quality. In some cases, exact labels cannot even be assigned to every sample point, and there is a need to resort to a multiple instance learning (MIL) formalism, where labels only exist for bags of points [55, 26, 76, 118, 6].

Different areas of computer vision have taken varied approaches to dealing with these problems. These include resorting to MIL algorithms for scene classification [55], object detection [104], or tracking [8], modeling context to reduce ambiguity in scene analysis [95], adopting parts-based models of greater flexibility with respect to occlusions and deformation [32, 31], etc. While such improvements in representation robustness are necessary, they cannot completely eliminate the ambiguity, noise, and outlier propensity of tasks such as image classification or tracking. Hence, there is an equally important need for more robust classifiers. In this context, an issue of particular concern is a well known limitation of most current margin-enforcing losses: their *unbounded growth* with negative margins. In statistics, this type of loss growth is classically known to produce inference procedures that are too *sensitive to outliers* [43, 80], a problem that has also been extensively studied in computer vision [64, 13, 83]. This research has shown that, for many vision applications, better results are obtained with losses of tapered growth. However, most of these results only apply to regression problems, such as surface fitting or optic flow estimation, and do not generalize to classification.

Robust classifier design has been studied in machine learning, namely in the boosting literature. Boosting algorithms, such as AdaBoost [33], have found multiple applications in vision, e.g. real-time object detection [105], tracking [7], and segmentation [109]. Yet, Adaboost is known to be particularly sensitive to noisy data [25], due to the exponential growth of its loss. Non-trivial improvements are due to [35], which introduced losses that grow *linearly* with the negative margin. The resulting boosting algorithms, e.g. LogitBoost, are known to be substantially more outlier resistant than AdaBoost [61]. Central to this contribution was the establishment, by this work, of a formal connection between the large margin approaches and classical decision theory. A number of other attempts to introduce robust classification losses, e.g. the noisy-OR [104] or sigmoidal non-linearities [60], lack this property. The resulting classifiers are not Bayes consistent, i.e. are not guaranteed to converge to the optimal Bayes decision rule [28] as datasets increase.

In Chapter II we established a framework for the derivation of novel Bayes consistent loss functions. In this chapter we first propose a new robust Bayes consistent loss, denoted as *Savage loss* and an associated *SavageBoost* algorithm . Unlike all previous Bayes consistent loss functions, the one now proposed remains constant for strongly negative values of its argument and trades convexity for boundedness. This is akin to robust loss functions proposed in the statistics literature to reduce the impact of outliers. We derive a new boosting algorithm, denoted SavageBoost, by combination of the new loss and the procedure used by Friedman to derive RealBoost [35]. Experimental results show that the new boosting algorithm is indeed more outlier resistant than classical methods, such as AdaBoost, RealBoost, and LogitBoost.

Unfortunately, the added robustness of SavagaBoost does not make a tremendous difference for all vision problems. We argue that this requires a more subtle constraint on the loss than simply bounding its growth for large negative margins: in addition to this, robustness requires *penalizing large positive margins*. We present a simple classification problem that demonstrates this point, and show how all existing methods (including SavageBoost) fail in this case. We then derive a set of necessary conditions that any Bayes consistent loss function must satisfy, in order to guarantee a bounded penalty for *both* large negative and positive margins. These conditions are used to derive a novel robust loss, which we denote by *Tangent loss*, and an associated boosting algorithm, denoted *TangentBoost*. Experiments involving various computer vision problems, including scene classification, object tracking, recognition, and MIL show that the proposed algorithm consistently

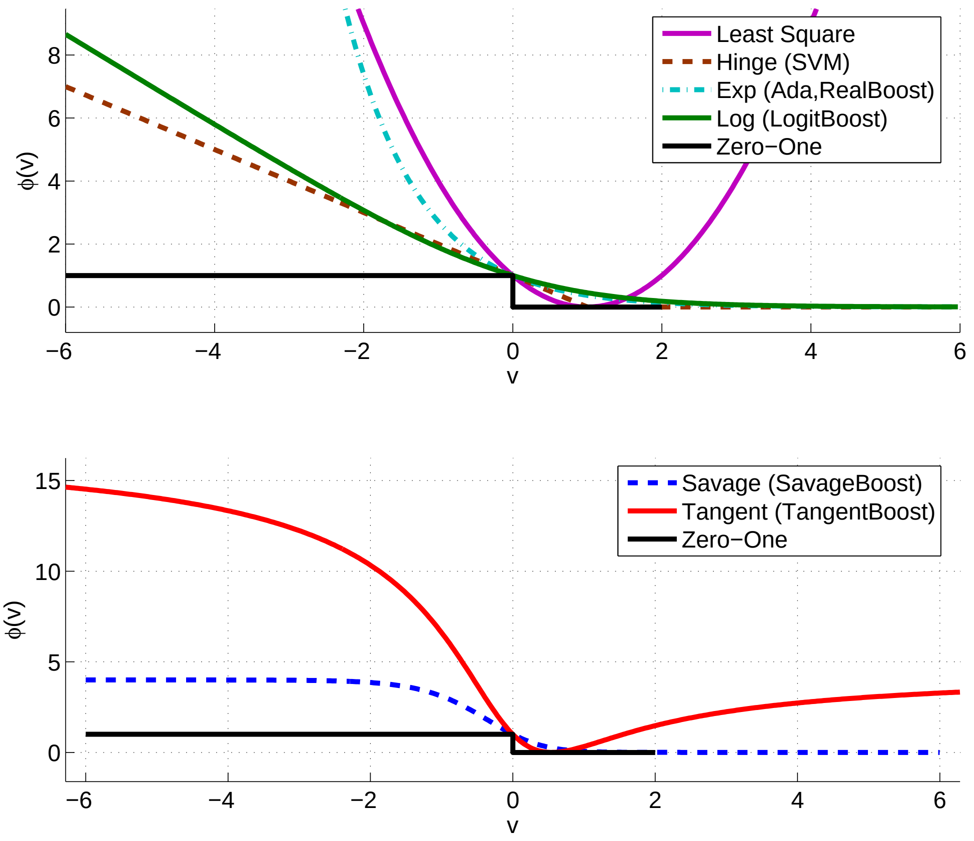


Figure IV.1 Loss functions used for classifier design in alternative to the non-margin enforcing $0-1$ loss. Top: classical non-robust losses. Bottom: robust losses of SavageBoost and TangentBoost.

outperforms previous boosting algorithms. In fact, for some of these problems, it is shown to achieve the best results reported to date on the literature.

IV.B Loss functions for classification

We start by briefly reviewing the theory of Bayes consistent classifier design. See [35, 17, 119, 57] for further details.

IV.B.1 Risk minimization

A classifier h maps a feature vector $\mathbf{x} \in \mathcal{X}$ to a class label $y \in \{-1, 1\}$. This mapping can be written as $h(\mathbf{x}) = sign[f(\mathbf{x})]$ for some function $f : \mathcal{X} \to \mathbb{R}$, which is denoted as the classifier predictor. Feature vectors and class labels are drawn from probability distributions $P_{\mathbf{X}}(\mathbf{x})$ and $P_Y(y)$ respectively. Given a non-negative loss function $L(\mathbf{x}, y)$, the classifier is optimal if it minimizes the risk $R(f) = E_{\mathbf{X},Y}[L(h(\mathbf{x}), y)]$. This is equivalent to minimizing the conditional risk $E_{Y|\mathbf{X}}[L(h(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}]$ for all $\mathbf{x} \in \mathcal{X}$. Classifiers are frequently designed to be optimal with respect to the zero-one loss

$$L_{0/1}(f, y) = \frac{1 - sign(yf)}{2} = \begin{cases} 0, & \text{if } y = sign(f); \\ 1, & \text{if } y \neq sign(f), \end{cases} \tag{IV.1}$$

where we omit the dependence of f on $\mathbf{x}$ for notational simplicity. The associated conditional risk is

$$\begin{aligned} C_{0/1}(\eta, f) &= \eta \frac{1 - sign(f)}{2} + (1 - \eta) \frac{1 + sign(f)}{2} \\ &= \begin{cases} 1 - \eta, & \text{if } f \geq 0; \\ \eta, & \text{if } f < 0 \end{cases} \end{aligned}$$

with $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$. Optimal predictors f^* that minimize this risk include $f^* = 2\eta - 1$, $f^* = \log \frac{\eta}{1-\eta}$, or any other function such that $f^* \geq 0$ if and only if $\eta \geq \frac{1}{2}$. The associated optimal classifier $h^* = sign[f^*]$ is the well known Bayes decision rule (BDR) and has minimum conditional risk

$$\begin{aligned} C^*_{0/1}(\eta) = \quad & \eta \left(\frac{1}{2} - \frac{1}{2} sign(2\eta - 1) \right) + \\ & (1 - \eta) \left(\frac{1}{2} + \frac{1}{2} sign(2\eta - 1) \right). \end{aligned} \tag{IV.2}$$

A loss which is minimized by the BDR is denoted as Bayes consistent. A number of Bayes consistent alternatives to the 0-1 loss are commonly used in machine learning. These include the exponential loss of boosting, the log loss of logistic regression, and the hinge loss of SVMs, which are shown in the top of

Table IV.1 Loss ϕ, predictor $f^*_\phi(\eta)$, minimum conditional risk $C^*_\phi(\eta)$ and predictor inverse $[f^*_\phi]^{-1}(v)$ for different machine learning algorithms.

Algorithm	$\phi(v)$	$f^*_\phi(\eta)$	$C^*_\phi(\eta)$	$[f^*_\phi]^{-1}(v)$
Least squares	$(1-v)^2$	$2\eta - 1$	$4\eta(1-\eta)$	$\frac{1}{2}(v+1)$
SVM	$\max(1-v,0)$	$sign(2\eta-1)$	$1-\lvert 2\eta-1\rvert$	NA
Boosting	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$2\sqrt{\eta(1-\eta)}$	$\frac{e^{2v}}{1+e^{2v}}$
Logistic Regression	$\log(1+e^{-v})$	$\log\frac{\eta}{1-\eta}$	-$\eta\log\eta-(1-\eta)\log(1-\eta)$	$\frac{e^{v}}{1+e^{v}}$

Figure IV.1. They have the form $L_\phi(f,y) = \phi(yf)$, for different functions ϕ of the margin yf. The non-zero penalty assigned to small positive margins encourages the creation of a margin, a property not shared by the 0-1 loss. The resulting *large-margin* classifiers have better generalization than those produced by the latter [98]. The associated conditional risk

$$C_\phi(\eta, f) = \eta\phi(f) + (1-\eta)\phi(-f) \tag{IV.3}$$

is minimized by the predictor

$$f^*_\phi(\eta) = \arg\min_f C_\phi(\eta, f) \tag{IV.4}$$

and has minimum $C^*_\phi(\eta) = C_\phi(\eta, f^*_\phi)$. The $\phi(v)$, $f^*_\phi(\eta)$, and $C^*_\phi(\eta)$ associated with popular algorithms for classifier design are shown in Table IV.1. See [119] for their derivations.

IV.B.2 Probability elicitation

Conditional risk minimization is closely related to classical probability elicitation in statistics [82]. Here, the goal is to find the probability estimator $\hat{\eta}$ that maximizes the expected reward

$$I(\eta, \hat{\eta}) = \eta I_1(\hat{\eta}) + (1-\eta)I_{-1}(\hat{\eta}), \tag{IV.5}$$

where $I_1(\hat{\eta})$ is the reward for prediction $\hat{\eta}$ when event $y = 1$ holds and $I_{-1}(\hat{\eta})$ the corresponding reward when $y = -1$. The functions $I_1(\cdot), I_{-1}(\cdot)$ must be such that the expected reward is maximal when $\hat{\eta} = \eta$, i.e.

$$I(\eta, \hat{\eta}) \leq I(\eta, \eta) = J(\eta), \ \ \forall \eta \tag{IV.6}$$

with equality if and only if $\hat{\eta} = \eta$. It can be shown [82] that (IV.6) holds if and only if 1) the maximal reward function $J(\eta)$ is strictly convex and 2)

$$\begin{aligned} I_1(\eta) &= J(\eta) + (1-\eta)J'(\eta) \quad &\text{(IV.7)} \\ I_{-1}(\eta) &= J(\eta) - \eta J'(\eta). \quad &\text{(IV.8)} \end{aligned}$$

The connection between risk minimization and probability elicitation has been studied in Chapter II. It was shown that if 1) $J(\eta) = J(1-\eta)$, and 2) the predictor f is invertible and has symmetry $f^{-1}(-v) = 1 - f^{-1}(v)$, the functions $I_1(\cdot)$ and $I_{-1}(\cdot)$ of (IV.7) and (IV.8) satisfy the following equalities

$$\begin{aligned} I_1(\eta) &= -\phi(f(\eta)) \quad &\text{(IV.9)} \\ I_{-1}(\eta) &= -\phi(-f(\eta)), \quad &\text{(IV.10)} \end{aligned}$$

for the loss

$$\phi(v) = -J[f^{-1}(v)] - (1 - f^{-1}(v))J'[f^{-1}(v)]. \quad \text{(IV.11)}$$

In this case, probability elicitation by maximization of (IV.5) is equivalent to risk minimization with (IV.3), and the minimum conditional risk is related to the maximal expected reward through $C^*_\phi(\eta) = -J(\eta)$. This establishes a new path for the design of learning algorithms. Rather than specifying a loss ϕ and minimizing $C_\phi(\eta, f)$, so as to obtain whatever optimal predictor f^*_ϕ and minimum expected risk $C^*_\phi(\eta)$ results, it is possible to specify f^*_ϕ and $C^*_\phi(\eta)$ and derive, from (IV.11) with $J(\eta) = -C^*_\phi(\eta)$, the underlying loss ϕ. The only conditions are that $C^*_\phi(\eta)$ is strictly concave, f^*_ϕ is invertible, and

$$\begin{aligned} C^*_\phi(\eta) &= C^*_\phi(1-\eta) \quad &\text{(IV.12)} \\ [f^*_\phi]^{-1}(-v) &= 1 - [f^*_\phi](v). \quad &\text{(IV.13)} \end{aligned}$$

IV.C The Savage loss

The main observation is that, under the customary specification of ϕ, both $C^*_\phi(\eta)$ and $f^*_\phi(\eta)$ are immediately set, leaving no open degrees of freedom.

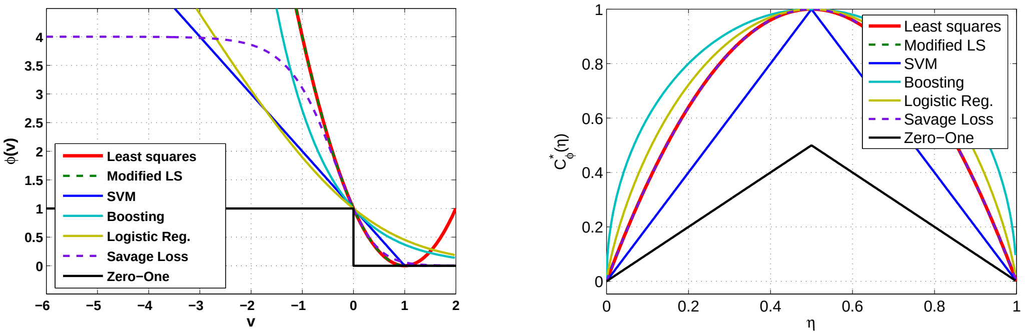


Figure IV.2 Loss function $\phi(v)$ (left) and minimum conditional risk $C^*_\phi(\eta)$ (right) associated with the different methods discussed in the text.

In fact, the selection of ϕ can be seen as the indirect selection of a link function $(f^*_\phi)^{-1}$ and a minimum conditional risk $C^*_\phi(\eta)$. The latter is an approximation to the minimum conditional risk of the *0-1 loss*, $C^*_{\phi_{0/1}}(\eta) = 1 - \max(\eta, 1-\eta)$. The approximations associated with the existing algorithms are shown in Figure IV.2. The approximation error is smallest for the SVM, followed by least squares, logistic regression, and boosting, but all approximations are comparable. The alternative, suggested by the probability elicitation view, is to start with the selection of the approximation directly. In addition to allowing direct control over the quantity that is usually of interest (the minimum expected risk of the classifier), the selection of $C^*_\phi(\eta)$ (which is equivalent to the selection of $J(\eta)$) has the added advantage of leaving one degree of freedom open. It is further possible to select across ϕ functions, by controlling the link function f_ϕ. This allows tailoring properties of detail of the classifier, while maintaining its performance constant, in terms of the expected risk.

We demonstrate this point, by proposing a new loss function ϕ. We start by selecting the minimum conditional risk of least squares (using Savage's version with $k = -l = 1, m = 0$) $C^*_\phi(\eta) = \eta(1-\eta)$, because it provides the best approximation to the Bayes error, while avoiding the lack of differentiability of the SVM. We next replace the traditional link function of least squares by the logistic link function (classically used with logistic regression) $f^*_\phi = \frac{1}{2}\log\frac{\eta}{1-\eta}$. When used in the context of boosting (LogitBoost [35]), this link function has been found less sensitive to outliers than other variants [61]. We then resort to (IV.11) to find the ϕ function, which we denote by *Savage loss*,

$$\phi(v) = \frac{1}{(1+e^{2v})^2}. \tag{IV.14}$$

A plot of this function is presented in Figure IV.2, along with those associated with all the algorithms of Table IV.1. Note that the proposed loss is very similar to that of least squares in the region where $|v|$ is small (the margin), but quickly becomes constant as $v \rightarrow -\infty$. This is unlike all other previous ϕ functions, and suggests that classifiers designed with the new loss should be more robust to outliers.

It is also interesting to note that the new loss function is not convex, violating what has been an hallmark of the ϕ functions used in the literature. The convexity of ϕ is, however, not important, a fact that is made clear by the elicitation view and elaborated upon in Chapter II.

IV.D SavageBoost

We have hypothesized that classifiers designed with (IV.14) should be more robust than those derived from the previous ϕ functions. To test this we designed a boosting algorithm based on the new loss, using the procedure proposed by Friedman to derive RealBoost [35]. At each iteration the algorithm searches for the weak learner $G(\mathbf{x})$ which further reduces the conditional risk $E_{Y|\mathbf{X}}[\phi(y(f(\mathbf{x})+G(\mathbf{x}))|\mathbf{X}=\mathbf{x}]$ of the current $f(\mathbf{x})$, for every $\mathbf{x} \in \mathcal{X}$. The optimal weak learner is

$$G^*(\mathbf{x}) = \arg\min_{G(\mathbf{x})} \big\{\eta(\mathbf{x})\phi_w(G(\mathbf{x})) + (1-\eta(\mathbf{x}))\phi_w(-G(\mathbf{x}))\big\} \tag{IV.15}$$

where

$$\phi_w(yG(\mathbf{x})) = \frac{1}{(1+w(\mathbf{x},y)^2e^{2y(G(\mathbf{x}))})^2} \tag{IV.16}$$

Algorithm 2 SavageBoost

```
**Input:** Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y \in \{1, -1\}$ is the class label of example $\mathbf{x}$, and number $M$ of weak learners in the final decision rule.
**Initialization:** Select uniform weights $w_i^{(1)} = \frac{1}{|\mathcal{D}|}, \forall i$.
**for** $m = \{1, \ldots, M\}$ **do**
  compute the gradient step $G_m(\mathbf{x})$ with (IV.18).
  update weights $w_i$ according to $w_i^{(m+1)} = w_i^{(m)} \times e^{y_i G_m(\mathbf{x}_i)}$.
**end for**
**Output:** decision rule $h(\mathbf{x}) = sgn[\sum_{m=1}^{M} G_m(\mathbf{x})]$.
```

and

$$w(\mathbf{x}, y) = e^{yf(\mathbf{x})} \tag{IV.17}$$

The minimization is by gradient descent. Setting the gradient with respect to $G(\mathbf{x})$ to zero results in

$$G^*(\mathbf{x}) = \frac{1}{2}\left(\log \frac{P_w(y=1|\mathbf{x})}{P_w(y=-1|\mathbf{x})}\right) \tag{IV.18}$$

where $P_w(y = i|\mathbf{x})$ are probability estimates obtained from the re-weighted training set. At each iteration the optimal weak learner is found from (IV.18) and reweighing is performed according to (IV.17). We refer to the algorithm as *SavageBoost*, and summarize it in the inset.

IV.E Experimental results

We compared SavageBoost to AdaBoost [33], RealBoost [35], and LogitBoost [35]. The latter is generally considered more robust to outliers [61] and thus a good candidate for comparison. Ten binary UCI data sets were used: Pima-diabetes, breast cancer diagnostic, breast cancer prognostic, original Wisconsin breast cancer, liver disorder, sonar, echo-cardiogram, Cleveland heart disease, tic-tac-toe and Haberman's survival. We followed the training/testing procedure outlined in [114] to explore the robustness of the algorithms to outliers. In all cases, Table IV.2 (number of wins, average error%) for each method and outlier percentage.

Method	0% outliers	5% outliers	40% outliers
Savage Loss (SavageBoost)	$(\mathbf{4}, \mathbf{19.22}\%)$	$(\mathbf{4}, \mathbf{19.91}\%)$	$(\mathbf{6}, \mathbf{25.9}\%)$
Log Loss(LogitBoost)	$(4, 20.96\%)$	$(4, 22.04\%)$	$(3, 31.73\%)$
Exp Loss(RealBoost)	$(2, 23.99\%)$	$(2, 25.34\%)$	$(0, 33.18\%)$
Exp Loss(AdaBoost)	$(0, 24.58\%)$	$(0, 26.45\%)$	$(1, 38.22\%)$

five fold validation was used with varying levels of outlier contamination. Figure IV.3 shows the average error of the four methods on the Liver-Disorder set. Table IV.2 shows the number of times each method produced the smallest error (#wins) over the ten data sets at a given contamination level, as well as the average error% over all data sets (at that contamination level). Our results confirm previous studies that have noted AdaBoost's sensitivity to outliers [25]. Among the previous methods AdaBoost indeed performed the worst, followed by RealBoost, with LogitBoost producing the best results. This confirms previous reports that LogitBoost is less sensitive to outliers [61]. SavageBoost produced generally better results than Ada and RealBoost at all contamination levels, including 0% contamination. LogitBoost achieves comparable results at low contamination levels $(0\%, 5\%)$ but has higher error when contamination is significant. With 40% contamination SavageBoost has 6 wins, compared to 3 for LogitBoost and, on average, about 6% less error. Although, in all experiments, each algorithm was allowed 50 iterations, SavageBoost converged much faster than the others, requiring an average of 25 iterations at 0% cantamination. This is in contrast to 50 iterations for LogitBoost and 45 iterations for RealBoost. We attribute fast convergence to the bounded nature of the new loss, that prevents so called ”early stopping” problems [120]. Fast convergence is, of course, a great benefit in terms of the computational efficiency of training and testing.

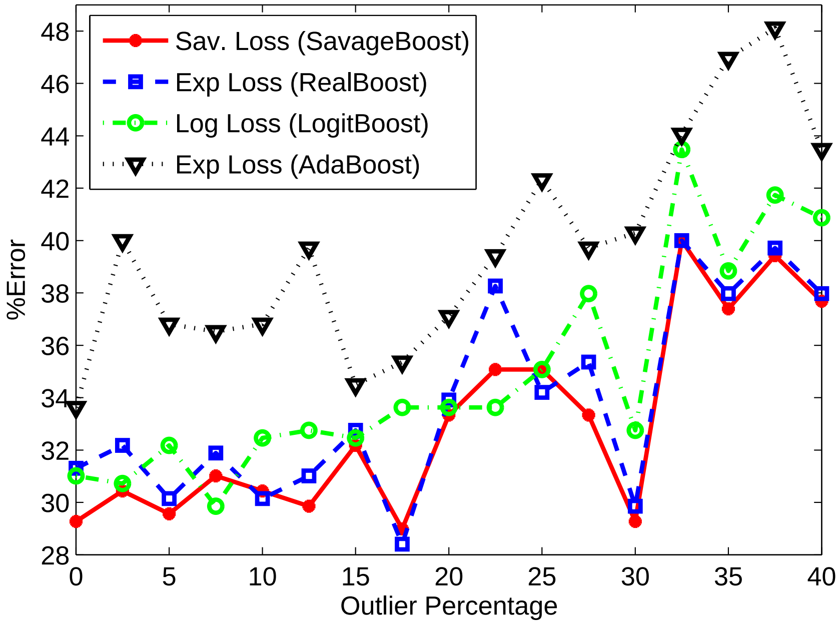


Figure IV.3 Average error for four boosting methods at different contamination levels.

IV.F Robust loss functions for computer vision

Computer vision problems frequently deviate from the canonical classification problem, due to the prevalence of noise, outliers, ambiguity, and imbalance of positive/negative training set sizes, in many vision applications. In this context, the losses shown at the top of Figure IV.1 are problematic in two ways. The first is their unbounded growth with negative values of the margin yf. This type of growth is well known to produce inference procedures that are too sensitive to outliers [43, 80]. For vision applications, better results are invariably obtained with loss functions of tapered growth [64, 13]. The second is the null penalty assigned to very large positive margins. This creates an incentive for the classifier to push, as far as possible from the boundary, the maximum possible number of points. Although less studied than the first problem, we contend that this can have an equally nefarious effect in terms of sensitivity to outliers.

We illustrate this point in Figure IV.4. The figure depicts the linearly separable problem that motivates the design of large-margin classifiers. The data come from two distributions that are uniform in the vertical direction and Gaussian, with equal variance and means $\mu = \pm 3$, in the horizontal direction. Given these distributions, the BDR is the vertical line $x = 0$. Figure IV.4 (top) shows ten data points sampled from each class and the decision boundary resulting from the minimization of the (empirical) risk associated with each loss. All losses of Figure IV.1 produce approximately the same boundary, close to the BDR.

Figure IV.4 (bottom) shows the impact of adding a single negative at location $(-2, 0)$. Both the classical losses and the robust Savage loss move the boundary substantially, to the vicinity of $x = -2.3$. This is due to the fact that this boundary classifies all points correctly, and the existing losses assign small penalty to correctly classified points. The result in as unwarranted leverage on the boundary by the outlier at $(-2, 0)$, compromising the generalization ability of the classifier. Also shown in the figure is the boundary produced by the loss (the tangent loss) proposed in this chapter. This loss, which is derived in the following sections, penalizes *both* large positive and large negative margins. The penalty assigned to large positive margins discourages solutions where large numbers of points are classified "too correctly". The force to classify the outlier correctly is countered by the force to avoid large numbers of points far away from the boundary. In result, the boundary remains close to the BDR ($x = -0.303$).

IV.F.1 Robust losses

The discussion above suggests that a robust loss for classifier design should have the following properties:

1. saturate for large margins: $\phi'(\infty) = \phi'(-\infty) = 0$;

2. bounded penalty for large negative margins: $\phi(-\infty) = k_1 < \infty$;

3. smaller positive penalty for large positive margins:

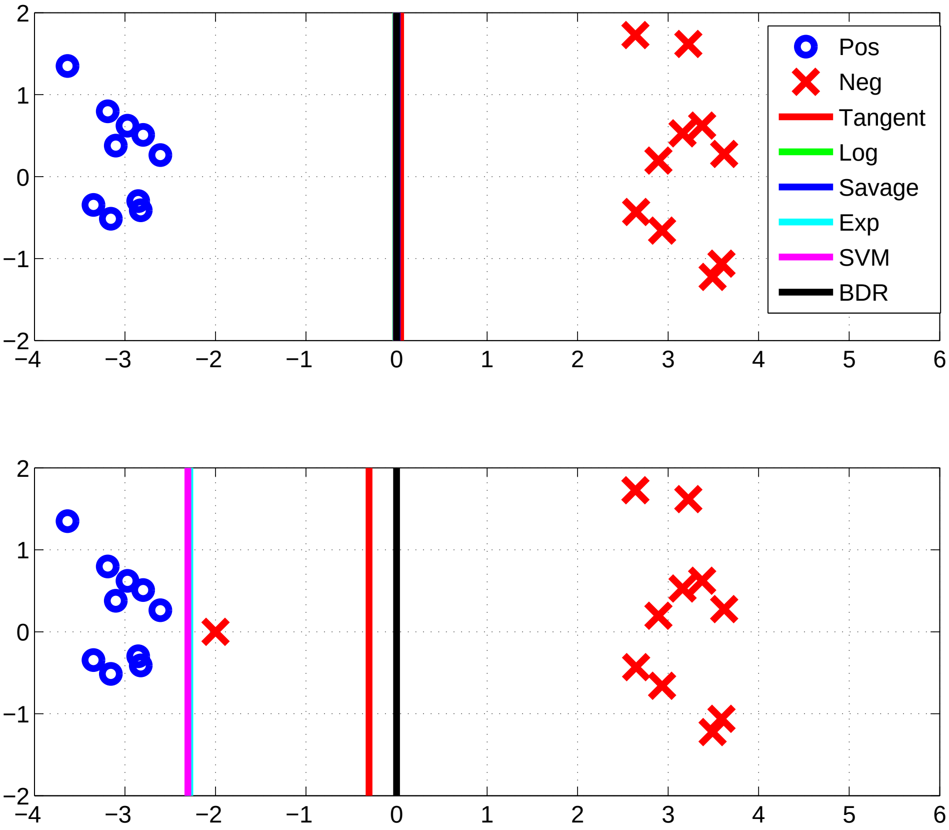


Figure IV.4 Minimum risk decision boundary for different loss functions. Top: outlier free problem. Bottom: impact of a single outlier.

$0 < \phi(\infty) = k_2 < k_1$;

4. margin enforcing: $\phi(0) > 0$

where we use the simplified notation $\phi(\infty) = \lim_{v \to \infty} \phi(v)$. As usual, the loss should be non-negative.

It can be shown, from (IV.11), that

$$\phi'(v) = -[1 - f^{-1}(v)] \times J''[f^{-1}(v)] \times [f^{-1}]'(v) \tag{IV.19}$$

From the strict convexity of $J(\eta)$, and (IV.13), it follows that property 1 holds if

$$[f^{-1}]'(\infty) = [f^{-1}]'(-\infty) = 0. \tag{IV.20}$$

This implies that the optimal predictor saturates as $v \to \pm\infty$. Furthermore, using the fact that $J(\eta) = J(1-\eta)$, $J'(\eta) = -J'(1-\eta)$, and (IV.13),

$$\begin{aligned} \phi(v) - \phi(-v) &= -J'[f^{-1}(v)] \\ (\phi(v) - \phi(-v))' &= -J''[f^{-1}(v)] \times [f^{-1}(v)]'. \end{aligned}$$

It follows from (IV.20) that $|\phi(v) - \phi(-v)|$ is maximum as $|v| \to \infty$. The condition $k_2 < k_1$ requires that $J'\left[f^{-1}(\infty)\right] > 0$. From the convexity and symmetry of $J(\eta)$ ($J'(1/2) = 0$) this holds whenever

$$f^{-1}(\infty) > \frac{1}{2}.$$

Defining $\gamma(v) = f^{-1}(-v) \times J'[f^{-1}(-v)]$, $k_2 > 0$ requires that $-J[f^{-1}(\infty)] > -\gamma(\infty)$, or $0 < C^*_\phi[f^{-1}(\infty)] + \gamma(\infty)$. Similarly, $k_1 < \infty$ requires that $C^*_\phi[f^{-1}(\infty)] + \gamma(-\infty) < \infty$. Finally, from (IV.13), $f^{-1}(0) = \frac{1}{2}$ and, from (IV.11) and $J'(1/2) = 0$, it follows that $\phi(0) = -J(1/2) = C^*_\phi(1/2) > 0$. In summary, the four properties are satisfied if

$$\begin{aligned} [f^{-1}]'(\infty) &= [f^{-1}]'(-\infty) = 0 && \text{(IV.21)} \\ f^{-1}(\infty) &> \frac{1}{2} = f^{-1}(0) && \text{(IV.22)} \\ C^*_\phi(1/2) &> 0 && \text{(IV.23)} \\ C^*_\phi[f^{-1}(\infty)] &+ \gamma(\infty) > 0 && \text{(IV.24)} \\ C^*_\phi[f^{-1}(\infty)] &+ \gamma(-\infty) < \infty && \text{(IV.25)} \end{aligned}$$

IV.F.2 The Tangent loss

In this section we seek to design a loss with the four properties discussed above, through the selection of a predictor $f^*_\phi(\eta)$ and minimum risk $C^*_\phi(\eta)$ that comply with conditions (IV.21)-(IV.25). We start by noting that some of these conditions hold for any sensible choice of these functions. For example, (IV.22) and (IV.23) are met by all methods of Table IV.1. On the other hand, (IV.21) disqualifies the predictor of least squares, but leaves the sigmoidal predictors of boosting and logistic regression as potential solutions. This suggests that conditions (IV.24) and (IV.25) are the most stringent. In fact, they fail to hold for all methods of Table IV.1.

Consider any of the sigmoidal predictors. Since $f^{-1}(\infty) = 1$, for any of the C^*_ϕ in the table, $C^*_\phi[f^{-1}(\infty)] = 0$. This simplifies (IV.24) and (IV.25) into

$$\gamma(\infty) = -f^{-1}(-\infty) \times [C^*_\phi]'[f^{-1}(-\infty)] > 0 \qquad \text{(IV.26)}$$

$$\gamma(-\infty) = -f^{-1}(\infty) \times [C^*_\phi]'[f^{-1}(\infty)] < \infty. \qquad \text{(IV.27)}$$

Since $f^{-1}(-\infty) = 0$, (IV.26) requires $[C^*_\phi]'(0) = -\infty$. In fact, because the sigmoid converges to 0 *exponentially* fast, (IV.26) requires the derivative of $[C^*_\phi](\eta)$ to decay to $-\infty$ (as $\eta \to 0$) at a (faster) exponential rate. This is not easy to guarantee, and does certainly not hold for any of the risks of Table IV.1. In summary, it appears that none of the predictors in the table is suitable for robust loss design. What is needed is a predictor such that $f^{-1}(v)$ saturates at $\pm\infty$, so as to satisfy (IV.21), but at a *slower than exponential* rate.

One possibility is the tangent

$$\begin{aligned} f(\eta) &= \tan(\eta - 0.5) & \text{(IV.28)} \\ f^{-1}(v) &= .5 + \arctan(v). & \text{(IV.29)} \end{aligned}$$

It has the symmetry of (IV.13), a *quadratic* decay rate ($[f^{-1}]'(v) = (1+x^2)^{-1}$) and is compatible for combination with the minimal conditional risk of least squares, $C^*_\phi(\eta) = 4\eta(1-\eta)$, resulting in

$$\begin{aligned} C^*_\phi[f^{-1}(\infty)] &+ \gamma(\infty) = (1-\pi)^2 > 0 \\ C^*_\phi[f^{-1}(\infty)] &+ \gamma(-\infty) = (1+\pi)^2 < \infty. \end{aligned}$$

It can be easily verified that conditions (IV.21)-(IV.23) also hold. Using (IV.11) it is possible to derive the ϕ function, which we denote by *Tangent loss*,

$$\phi(v) = (2\arctan(v) - 1)^2. \qquad \text{(IV.30)}$$

Figure IV.1 (bottom) shows that the Tangent loss is similar to the Savage loss in the sense that it is non convex, and bounded for large negative margins. It, however, also penalizes points of large *positive* margin. This penalty is, once again, bounded and of smaller value than that assigned to large negative margins. Overall, the Tangent loss is margin enforcing, and encourages all points to be classified correctly. However, it discourages situations where a large number of points are classified "too correctly". We will see, in Section IV.H, that this leads to superior performance for a number of vision problems.

IV.G The TangentBoost algorithm

In this section we derive a boosting algorithm based on the Tangent loss. This consists of minimizing the empirical risk

$$R = \sum_i \phi(yf(x)) \tag{IV.31}$$

by gradient descent on the space of linear combinations of weak learners. The fact that this is a sum of squared values, suggests performing the minimization with the Gauss algorithm. For a general sum of squares problem

$$S(x) = \sum_{i=1}^{N} r_i^2(x) \tag{IV.32}$$

this has update step

$$x^{n+1} = x^n + \frac{-r(x)}{\frac{\partial r}{\partial x}} \tag{IV.33}$$

As in the case of LogitBoost [35], it is more convenient to work with the intermediate probability estimates $\eta(x_i)$ than the points x_i. For the Tangent loss

$$r(\eta) = 2\arctan(yf(\eta)) - 1 \tag{IV.34}$$

the optimal solution is

$$f^* = \arg\min_f \sum_{i=1}^{N} (2\arctan(yf(\eta(x_i))) - 1)^2. \tag{IV.35}$$

The Gauss update is

$$\begin{aligned} f(\eta)^{n+1} = &\quad f(\eta)^n + \Delta f(\eta) = f(\eta)^n - \frac{r(\eta)}{\frac{\partial r}{\partial \eta}} \\ = &\quad f(\eta)^n - \frac{2\arctan(yf(\eta)) - 1}{\frac{2y}{1+f(\eta)^2}} \\ = &\quad f(\eta)^n - \frac{(2\arctan yf(\eta) - 1)(1 + f(\eta)^2)}{2y}. \end{aligned} \tag{IV.36}$$

Using the known form of the optimal predictor $f(\eta) = \tan(\eta - 0.5)$ and its inverse $\eta = \arctan(f(\eta)) + 0.5$ we redefine the above updates as follows. For $y = 1$,

$$\begin{aligned} z(\eta)_1 = &\quad -\frac{(2\arctan(f(\eta)) - 1)(1 + f(\eta)^2)}{2} \\ &\quad = -(\eta - 1)(1 + \tan^2(\eta - 0.5)) \end{aligned} \tag{IV.37}$$

and for $y = -1$ as

$$\begin{aligned} z(\eta)_{-1} = &\quad -\frac{(-2\arctan(f(\eta)) - 1)(1 + f(\eta)^2)}{-2} \\ &\quad = -\eta(1 + \tan^2(\eta - 0.5)) \end{aligned} \tag{IV.38}$$

The linear regression model can now be used to approximate $z(\eta)$, as is done in logistic regression. This leads to the TangentBoost algorithm described in Algorithm 1.

IV.H Experiments

In this section we describe several experiments designed to test the performance of TangentBoost in classification problems involving outliers and noisy data.

We start with a simple classification problem, which provides some insight on the benefits of the Tangent loss. This problem involves the Letter-1 dataset, from the UCI database. It addresses the classification of the highly confusable letter "O" from the other letters of the alphabet, resulting in an unbalanced problem with many outliers. Figure IV.5 shows the histogram of the positive margins

Algorithm 3 TangentBoost

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y \in \{1, -1\}$ is the class label of example $\mathbf{x}$, and number $M$ of weak learners in the final decision rule.
Initialization: Set uniformly distributed probabilities $\eta^{(1)}(\mathbf{x}_i) = \frac{1}{2} \ \forall \mathbf{x}_i$ and $\hat{f}^{(1)}(\mathbf{x}) = 0$.
for $m = \{1, \ldots, M\}$ do
    compute the working responses $z_i^{(m)}$ as in (IV.37) and (IV.38) and weights $w_i^{(m)} = \eta^{(m)}(x_i)(1 - \eta^{(m)}(x_i))$ .
    for $k = \{1, \ldots, K\}$ do
        compute the solution to the least squares problem,
            $a_{\phi_k} = \frac{\langle 1 \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$
            $b_{\phi_k} = \frac{\langle \phi_k(\mathbf{x}_i)^2 \rangle_w \cdot \langle z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$
        where we have defined
            $\langle q(\mathbf{x}_i) \rangle_w \doteq \sum_i w_i^{(m)} q(\mathbf{x}_i).$
    end for
    select the direction of minimal regression error according to $k^* = \arg\min_k \sum_i w_i^{(m)} (z_i - a_{\phi_k} \phi_k(\mathbf{x}_i) - b_{\phi_k})^2$ .
    set $\hat{f}^{(m+1)}(\mathbf{x}_i) = \hat{f}^{(m)}(\mathbf{x}_i) + (a_{\phi_k} \phi_k(\mathbf{x}_i) + b_{\phi_k})$.
    update $\eta^{(m+1)}(\mathbf{x}_i) = \arctan(\hat{f}^{(m+1)}(\mathbf{x}_i)) + 0.5$.
end for
Output: decision rule $h(\mathbf{x}) = \mathrm{sgn}[\hat{f}^{(M)}(\mathbf{x})]$.
```

on the test set (a very similar histogram exists on the train set), for classifiers learned with TangentBoost and Adaboost. Note that the TangentBoost margins are below 0.7 and much smaller than those of AdaBoost (largest margin greater than 2.5). On the other hand, the number of classification errors on the test set is 602 for TangentBoost and 3621 for AdaBoost. This shows that larger margins do not necessarily lead to better classification when there are outliers. In its effort to push points away from the boundary, AdaBoost sacrifices classification performance. On the other hand, TangentBoost has a much cleaner margin distribution, with no points of positive margin smaller than .25.

It should be noted that, while this problem is serious for AdaBoost, it

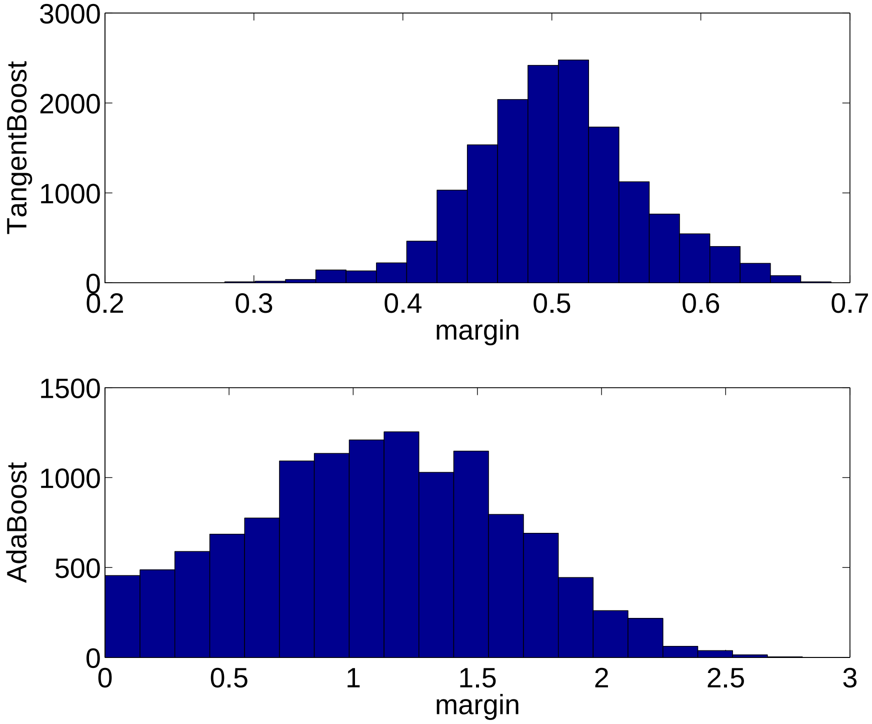


Figure IV.5 Histogram of positive test margins for the TangentBoost (top) and AdaBoost (bottom) algorithms on the Letter-1 dataset.

Table IV.3 Classification error of each boosting method on Letter-1.

Dataset	Ada	Real	Savage	Logit	Tangent
LETT.1	3621	2681	647	616	**602**

affects most boosting algorithms in current use. Table IV.3 presents the error rates achieved by some of these, after 1000 iterations of training. Adaboost and Realboost, which employ the exponential loss, have disproportionately high error. The bounded Savage loss and the linearly increasing loss of logistic regression produce a dramatic improvement. Finally, TangentBoost has the best performance. The benefits of employing a bounded loss function (Savage and Tangent) or a gradually sloping loss (logistic) are evident.

Table IV.4 MIL accuracy on the MUSK2 dataset.

Boosting Alg.	Real	Ada	Logit	Savage	Tangent	
MUSK2	67.25	82.69	84.07	85.19	85.39	

MIL Alg.	MI-NN[76]	mi-SVM[6]	DD [55]	MI-SVM [6]	EMDD [118]	IAPR [26]
MUSK2	82.5	83.6	84	84.3	84.9	89.2

IV.H.1 The MUSK dataset

Various authors have formulated outlier ridden vision problems, such as image classification [55], object detection [104], and tracking [8], as MIL problems. Unfortunately, these formulations are not directly comparable, and some of the datasets used are not in the public domain. An alternative is the MUSK [26] dataset, which is the standard benchmark for the broader MIL research community [26, 76, 118, 55, 26, 6]. It is a good dataset to evaluate outlier robustness, since it is naturally contaminated with misclassified data points. We learned classifiers with AdaBoost, RealBoost, LogitBoost, SavageBoost, and TangentBoost on the MUSK2 dataset, using the training and testing protocol of [26]: 10-fold cross validation, with the 10 dataset partitions defined by [26]. The test error achieved by each classifier is reported in Table IV.4, which also includes results from various MIL algorithms not based on boosting. Note that although SavageBoost and TangentBoost do not fit the traditional MIL definition (don't operate on bags of points), they outperform this broad selection of state-of-the-art MIL procedures. The only exception is IAPR [26] which is an algorithm specifically designed for the MUSK dataset.

IV.H.2 Results on scene classification

We next considered the vision problem of scene classification on the 15-class dataset of [47]. Here, label noise occurs naturally, as each picture can be attributed to multiple scene categories (e.g. an image containing patches of both highway and buildings). State-of-the-art results on this dataset were recently reported in [77, 78]. These methods represent images as points on a semantic space, where each feature is the probability of the image belonging to one of the 15 classes. The two methods differ in the computation of these probabilities, one using Gaussian mixtures [77] and the other mixtures of Dirichlet distributions [78]. The probability vectors are fed to an SVM classifier, which we replaced by one learned with TangentBoost.

Table IV.5 compares results to different methods reported in the literature. TangentBoost(A), learned from Gaussian mixture probabilities, achieved the *highest accuracy reported for this dataset in the literature*, with 76.28%. Note that this is 2% better than the accuracy achieved with Adaboost under the same setting. This gain can only be attributed to the increased robustness of TangentBoost to outliers and noise. Also reported are the results of TangentBoost(B), where we have combined the Gaussian and Dirichelet mixture probabilities, by simply concatenating the 15 class features of both into a 30 dimensional vector. This further increased performance to 76.74% accuracy. It is also interesting to note that the greatest improvements in accuracy are achieved for the classes where [77] performs worst. These are classes that 1) are easily confusable with other classes in the dataset, and 2) contain many outliers. For example, the classification of scenes of ”street”, ”highway”, and ”tall building” improves in accuracy by 21%, 12%, and 10%. Similarly, the easily confused classes of ”mountain”, ”open country”, ”forest”, and ”coast” have relative increase in accuracy of 14%, 7%, 6%, and 6% . Finally, ”bedroom” displays a 20% increase in accuracy.

IV.H.3 Results on object tracking

Discriminant tracking has recently been shown to be a very effective solution to the object tracking problem [7]. It is also a prime domain for testing the effectiveness of classifiers in the presence of noise and outliers. This arises from the fact that the positive and negative training sets are collected from windows centered at the location of the current detection. In challenging scenes, object

Table IV.5 Classification accuracy for 15 scene categories.

Method	Dimensions	Accuracy%
TangentBoost(B)	30	**76.74**
TangentBoost(A)	15	**76.28**
AdaBoost	15	74.79
Rasiwasia et al. [78]	15	72.5
Rasiwasia et al. [77]	15	72.2
Liu et al. [52]	20	63.32
Liu et al. [52]	200	75.16
Lazebnik et al. [47]	200	72.2

boundaries are not necessarily well defined, and the target object can be subject to occlusion, shadows, and others sources of “noise”. These cause drift, since a poor localization of the target will contaminate the training data with outliers, i.e. background features labeled as target and vice-versa.

The original ensemble tracker of [7] was based on AdaBoost. It has however been noted that, in the tracking context, AdaBoost is quite susceptible to the outlier problem, and various approaches have recently been shown to outperform it [54, 8]. We consider here the discriminant saliency tracker (DST) of [54], which maps the video frames into a feature space where the target is *salient* compared to the background. Tracking is implemented with a weak classifier, which simply sums the saliency maps produced by each feature. Here, we investigate the use of boosting to combine these saliency maps in a discriminant manner. We implemented both AdaBoost and TangentBoost to achieve this combination. The results of the boosted tracker, for 2 noisy clips used in [54], are shown in Table IV.6. The error rates are measured as defined in [54]. It can be seen that the tracker based on AdaBoost has substantially larger error, in fact losing the target at some point in these sequences. On the other hand, TangentBoost produces a tracker that does not loose the target, and has an overall low error rate. Two representative frames of the process are shown in Figure IV.6.

Table IV.6 Tracking error rates on two noisy sequences.

Clip	AdaBoost	TangentBoost
athlete	0.89	**0.29**
gravel	0.70	**0.04**

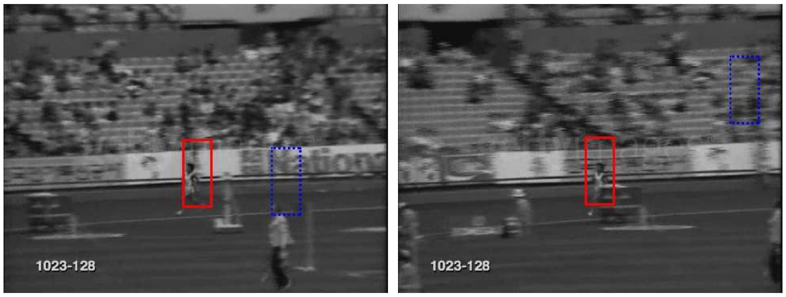


Figure IV.6 Frames comparing the performance of TangentBoost with AdaBoost in conjunction with a discriminant saliency tracker. Red box: TangentBoost, blue box: AdaBoost.

IV.I Summary and discussion

In this chapter, we have extended the probability elicitation view of loss function design introduced in Chapter II to the problem of designing robust loss functions for classification. The robust Savage loss and corresponding SavageBoost algorithm was derived and shown to outperform other boosting algorithms on a set of experiments designed to test the robustness of the algorithms to outliers in the training data. We next argued that a robust loss should penalizes both large positive and large negative margins. A set of four properties were derived that a robust loss function should have and the Tangent loss was introduced with the four properties discussed. Finally the associated TangentBoost algorithm was derived and shown to outperform other boosting algorithms on a variety of test sets involving various computer vision problems, including scene classification, object tracking, recognition, and MIL problems. Empirical evidence confirms the importance of using robust Bayes consistent loss functions when dealing with noise, outliers and class ambiguity within the data.

IV.J Acknowledgments

The author would like to thank Vijay Mahadevan for conducting the tracking experiments in section IV.H.3 and Nikhil Rasiwasia for providing the scene classification data in section IV.H.2.

The text of Chapter IV, in part, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “On the Design of Loss Functions for Classification: theory, robustness to outliers, and SavageBoost” , in *Neural Information Processing Systems (NIPS)*, 2008 and Hamed Masnadi-Shirazi, Vijay Mahadevan and Nuno Vasconcelos, “On the Design of Robust Classifiers for Computer Vision” , in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. The dissertation author was a primary researcher and an author of the cited material.

Chapter V

Cost sensitive loss function design and cost sensitive SVMs

V.A Introduction

The most popular strategy for the design of classification algorithms is to minimize the probability of error, assuming that all misclassifications have the same cost. The resulting decision rules are usually denoted as *cost-insensitive*. However, in many important applications of machine learning, such as medical diagnosis, fraud detection, or business decision making, certain types of error are much more costly than others. Other applications involve significantly unbalanced datasets, where examples from different classes appear with substantially different probability. It is well known, from Bayesian decision theory, that under any of these two situations (uneven costs or probabilities), the optimal decision rule deviates from the optimal cost-insensitive rule in the same manner. In both cases, reliance on cost insensitive algorithms for classifier design can be highly sub-optimal. While this makes it obviously important to develop *cost-sensitive* extensions of state-of-the-art machine learning techniques, the current understanding of such extensions is limited.

In this chapter we lay the theoretical foundation for cost sensitive loss function design but mostly consider the support vector machine (SVM) [22] application. The theory as applied to cost sensitive boosting algorithms is considered in Chapter VI. Although SVMs are based on a very solid learning-theoretic foundation, and have been successfully applied to many classification problems, it is not well understood how to design cost-sensitive extensions of the SVM learning algorithm. The standard, or cost-insensitive, SVM is based on the minimization of a symmetric loss function (the hinge loss) that does not have an obvious cost-sensitive generalization. In the literature, this problem has been addressed by various approaches, which can be grouped into three general categories. The first is to address the problem as one of data processing, by adopting resampling techniques that under-sample the majority class and/or over-sample the minority class [46, 20, 3]. Resampling is not easy when the classification unbalance is due to either different misclassification costs (not clear what the class probabilities should be) or an extreme unbalance in class probabilities (sample starvation for classes of very low probability). It also does not guarantee that the learned SVM will change, since it could have no effect on the support vectors. The second class of approaches [4, 110, 111] involves kernel modifications. These methods are based on conformal transformations of the input or feature space, by modifying the kernel used by the SVM. They are somewhat unsatisfactory, due to the implicit assumption that a linear SVM cannot be made cost-sensitive. It is unclear why this should be the case.

The third, and most widely researched, approach is to modify the SVM algorithm in order to achieve cost sensitivity. This is done in one of two ways. The first is a naive method, known as *boundary movement (BM-SVM),* which shifts the decision boundary by simply adjusting the threshold of the standard SVM [45]. Under Bayesian decision theory, this would be the optimal strategy if the class posterior probabilities were available. However, it is well known that SVMs do not predict these probabilities accurately. While a literature has developed in the area of probability calibration [75, 29], calibration techniques do not aid the cost-sensitive performance of threshold manipulation. This follows from the fact that all calibration techniques rely on an invertible (monotonic and one-to-one) transformation of the SVM output. Because the manipulation of a threshold at either the input or output of such a transformation produces the same receiver-operating-characteristic (ROC) curve, calibration does not change cost-sensitive classification performance. The boundary movement method is also obviously flawed when the data is non-separable, in which case cost-sensitive optimality is expected to require a modification of *both* the normal of the separating plane w and the classifier threshold b. The second proposal to modify SVM learning is known as the *biased penalties (BP-SVM)* method [9, 51, 23, 113, 19]. This consists of introducing different penalty factors C_1 and C_{-1} for the positive and negative SVM slack variables during training. It is implemented by transforming the primal SVM problem into

$$\begin{aligned}
&\arg\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\left[C_1 \sum_{\{i|y_i=1\}} \xi_i + C_{-1} \sum_{\{i|y_i=-1\}} \xi_i\right] \\
&\text{s.t. } y_i(w^T x + b) \geq 1 - \xi_i. \qquad \text{(V.1)}
\end{aligned}$$

The biased penalties method also suffers from an obvious flaw, which is converse to that of the boundary movement method: it has limited ability to enforce cost-sensitivity when the training data is separable. For large slack penalty C, the slack variables ξ_i are zero-valued and the optimization above degenerates into that of the standard SVM, where the decision boundary is placed midway between the two classes (rather than assigning a larger margin to one of them).

In this chapter we propose an alternative strategy for the design of cost-sensitive SVMs. This strategy is fundamentally different from previous attempts, in the sense that is does not directly manipulate the standard SVM learning algorithm. Instead, we extend the SVM hinge loss, and derive the optimal cost-sensitive learning algorithm as the minimizer of the associated risk. The derivation of the new cost-sensitive hinge loss draws on the connections between risk minimization and probability elicitation as mentioned in Chapter II. Such connections are generalized to the case of cost-sensitive classification.

It is shown that it is always possible to specify the predictor and conditional risk functions desired for the SVM classifier, and derive the loss for which these are optimal. A sufficient condition for the cost-sensitive Bayes optimality of the predictor is then provided, as well as necessary conditions for conditional risks that approximate the cost-sensitive Bayes risk. Together, these conditions enable the design of a new hinge loss which is minimized by an SVM that 1) implements the cost-sensitive Bayes decision rule, and 2) approximates the cost-sensitive Bayes risk. It is also shown that the minimization of this loss is a generalization of the classic SVM optimization problem, and can be solved by identical procedures. The resulting algorithm avoids the shortcomings of previous methods, producing cost-sensitive decision rules for *both* cases of separable and inseparable training data.

Experimental results show that these advantages result in better cost-sensitive classification performance than previous solutions.

This chapter is organized as follows. Section V.B briefly reviews the probability elicitation view of loss function design as seen in Chapter II. Section V.C then generalizes the connections between probability elicitation and risk minimization to the cost-sensitive setting. In Section V.D, these connections are used to derive the new SVM loss and algorithm. Finally, Section V.E presents an experimental evaluation that demonstrates improved performance of the proposed cost sensitive SVM over previous methods.

V.B Probability elicitation and the risk

A classifier h maps a feature vector $\mathbf{x} \in \mathcal{X}$ to a class label $y \in \{-1, 1\}$. This mapping can be written as $h(\mathbf{x}) = sign[f(\mathbf{x})]$ for some function $f : \mathcal{X} \to \mathbb{R}$, which is denoted as the classifier predictor. Feature vectors and class labels are drawn from probability distributions $P_{\mathbf{X}}(\mathbf{x})$ and $P_Y(y)$ respectively. Given a non-negative loss function $L(\mathbf{x}, y)$, the classifier is optimal if it minimizes the risk $R(f) = E_{\mathbf{X},Y}[L(h(\mathbf{x}), y)]$. This is equivalent to minimizing the conditional risk

$$\begin{aligned} E_{Y|\mathbf{X}}[L(h(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}] = P_{Y|\mathbf{X}}(1|\mathbf{x})L(h, 1) \\ +(1 - P_{Y|\mathbf{X}}(1|\mathbf{x}))L(h, -1), \end{aligned} \tag{V.2}$$

for all $\mathbf{x} \in \mathcal{X}$. Classifiers are frequently designed to be optimal with respect to the zero-one loss

$$\begin{aligned} L_{0/1}(f, y) &= \frac{1 - sign(yf)}{2} \\ &= \begin{cases} 0, & \text{if } y = sign(f); \\ 1, & \text{if } y \neq sign(f), \end{cases} \end{aligned} \tag{V.3}$$

where we omit the dependence on $\mathbf{x}$ for notational simplicity. The associated conditional risk is

$$\begin{aligned} C_{0/1}(\eta, f) &= \eta \frac{1 - sign(f)}{2} + (1-\eta)\frac{1+sign(f)}{2} \\ &= \begin{cases} 1-\eta, & \text{if } f \geq 0; \\ \eta, & \text{if } f < 0, \end{cases} \end{aligned} \tag{V.4}$$

with $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$. This risk is minimized by any predictor f such that

$$\begin{cases} f(\mathbf{x}) > 0 & \text{if } \eta(\mathbf{x}) > \gamma \\ f(\mathbf{x}) = 0 & \text{if } \eta(\mathbf{x}) = \gamma \\ f(\mathbf{x}) < 0 & \text{if } \eta(\mathbf{x}) < \gamma \end{cases} \tag{V.5}$$

and $\gamma = \frac{1}{2}$. Examples of optimal predictors include $f^* = 2\eta - 1$ and $f^* = \log \frac{\eta}{1-\eta}$. The associated optimal classifier $h^* = sign[f^*]$ is the well known Bayes decision rule, and the associated minimum conditional (zero-one) risk is

$$\begin{aligned} C^*_{0/1}(\eta) = \eta \left(\frac{1}{2} - \frac{1}{2} sign(2\eta - 1) \right) + \\ (1-\eta)\left(\frac{1}{2} + \frac{1}{2} sign(2\eta - 1) \right). \end{aligned} \tag{V.6}$$

A number of other losses have been proposed in the literature. Popular examples include the exponential loss of boosting, binomial loss of logistic regression, or hinge loss of SVMs. These losses are of the form $L_\phi(f, y) = \phi(yf)$, for different functions $\phi(\cdot)$. The associated conditional risk

$$C_\phi(\eta, f) = \eta\phi(f) + (1-\eta)\phi(-f). \tag{V.7}$$

is minimized by the predictor

$$f^*_\phi(\eta) = \arg\min_f C_\phi(\eta, f) \tag{V.8}$$

leading to the minimum conditional risk function $C^*_\phi(\eta) = C_\phi(\eta, f^*_\phi)$.

Conditional risk minimization is closely related to classical probability elicitation in statistics [82]. Here, the goal is to find the probability estimator $\hat{\eta}$ that maximizes the expected reward

$$I(\eta, \hat{\eta}) = \eta I_1(\hat{\eta}) + (1-\eta) I_{-1}(\hat{\eta}), \tag{V.9}$$

where $I_1(\hat{\eta})$ is the reward for prediction $\hat{\eta}$ when event $y = 1$ holds and $I_{-1}(\hat{\eta})$ the corresponding reward when $y = -1$. The functions $I_1(\cdot), I_{-1}(\cdot)$ should be such that the expected reward is maximal when $\hat{\eta} = \eta$, i.e.

$$I(\eta, \hat{\eta}) \leq I(\eta, \eta) = J(\eta), \;\; \forall \eta \tag{V.10}$$

with equality if and only if $\hat{\eta} = \eta$. The following theorem establishes the conditions under which this holds.

Theorem 38. *[82] Let $I(\eta, \hat{\eta})$ and $J(\eta)$ be as defined in (V.9) and (V.10). Then 1) $J(\eta)$ is convex and 2) (V.10) holds if and only if*

$$I_1(\eta) \;\; = \;\; J(\eta) + (1 - \eta)J'(\eta) \tag{V.11}$$

$$I_{-1}(\eta) \;\; = \;\; J(\eta) - \eta J'(\eta). \tag{V.12}$$

It follows from the theorem that, starting from any convex $J(\eta)$, it is possible to derive $I_1(\cdot), I_{-1}(\cdot)$ so that (V.10) holds. The next theorem as seen in Chapter II and [57] connects this result to the problem of classifier design.

Theorem 39. *Let $J(\eta)$ be as defined in (V.10) and f a continuous function. If the following properties hold*

1. *$J(\eta) = J(1 - \eta)$,*
2. *f is invertible with symmetry*

$$f^{-1}(-v) = 1 - f^{-1}(v), \tag{V.13}$$

then the functions $I_1(\cdot)$ and $I_{-1}(\cdot)$ derived with (V.11) and (V.12) satisfy the following equalities

$$I_1(\eta) \;\; = \;\; -\phi(f(\eta)) \tag{V.14}$$

$$I_{-1}(\eta) \;\; = \;\; -\phi(-f(\eta)), \tag{V.15}$$

with

$$\phi(v) = -J[f^{-1}(v)] - (1 - f^{-1}(v))J'[f^{-1}(v)]. \tag{V.16}$$

This theorem connects (V.9) and (V.7), establishing a new path for the design of learning algorithms. Rather than specifying a loss ϕ and minimizing $C_\phi(\eta, f)$, so as to obtain whatever optimal predictor f_ϕ^* and minimum expected risk $C_\phi^*(\eta)$ results, it is possible to specify f_ϕ^* and $C_\phi^*(\eta)$ and derive, from (V.16) with $J(\eta) = -C_\phi^*(\eta)$, the underlying loss ϕ. The only conditions are that $C_\phi^*(\eta) = C_\phi^*(1-\eta)$ and that (V.13) holds for f_ϕ^*. Note that 1) the symmetry of (V.13) guarantees that f meets the necessary conditions of (V.5) for predictor optimality (see Theorem 41) and 2) the condition of $C_\phi^*(\eta) = C_\phi^*(1-\eta)$ encodes the fact that there is no preference for different types of errors since the risk, or expected loss, is the same for any two $\mathbf{x}_1$ and $\mathbf{x}_2$ at the same distance from the boundary, where distance is measured is units of posterior probability ($|\eta(\mathbf{x}) - 1/2|$).

V.C Cost sensitive losses and classifier design

In this section we extend the connections between risk minimization and probability elicitation to the cost-sensitive setting. We start by reviewing cost-sensitive losses.

V.C.1 Cost-sensitive losses

The cost-sensitive extension of the zero-one loss is

$$\begin{aligned} & L_{C_1,C_{-1}}(f,y) = \\ & \frac{1-sign(yf)}{2}\left(C_1\frac{1-sign(f)}{2} + C_{-1}\frac{1+sign(f)}{2}\right) \\ & = \begin{cases} 0, & \text{if } y = sign(f); \\ C_1, & \text{if } y = 1 \text{ and } sign(f) = -1 \\ C_{-1}, & \text{if } y = -1 \text{ and } sign(f) = 1, \end{cases} \end{aligned} \tag{V.17}$$

where C_1 is the cost of a false negative, or miss, and C_{-1} that of a false positive. The associated conditional risk is

$$\begin{aligned} & C_{C_1,C_{-1}}(\eta, f) = \\ & C_1\eta\frac{1-sign(f)}{2} + (1-\eta)C_{-1}\frac{1+sign(f)}{2} = \\ & = \begin{cases} C_{-1}(1-\eta), & \text{if } f \geq 0; \\ C_1\eta, & \text{if } f < 0, \end{cases} \end{aligned} \tag{V.18}$$

and is minimized by any predictor that satisfies (V.5) with $\gamma = \frac{C_{-1}}{C_1+C_{-1}}$. Examples of optimal predictors include $f^*(\eta) = (C_1 + C_{-1})\eta - C_{-1}$ and $f^*(\eta) = \log\frac{\eta C_1}{(1-\eta)C_{-1}}$. The associated optimal classifier $h^* = sign[f^*]$ is the cost-sensitive Bayes decision rule, and the associated minimum conditional (cost-sensitive) risk is

$$\begin{aligned} C^*_{C_1,C_{-1}}(\eta) = C_1\eta\left(\frac{1}{2} - \frac{1}{2}sign\left[f^*(\eta)\right]\right) + \\ C_{-1}(1-\eta)\left(\frac{1}{2} + \frac{1}{2}sign\left[f^*(\eta)\right]\right) \end{aligned} \tag{V.19}$$

with $f^*(\eta) = (C_1 + C_{-1})\eta - C_{-1}$. To extend the other losses used in machine learning to the cost-sensitive setting, we consider the following set of loss functions

$$\begin{aligned} L_{\phi,C_1,C_{-1}}(f,y) &= \phi_{C_1,C_{-1}}(yf) \\ &= \begin{cases} \phi_1(f), & \text{if } y = 1 \\ \phi_{-1}(-f), & \text{if } y = -1. \end{cases} \end{aligned} \tag{V.20}$$

The associated conditional risk

$$C_{\phi,C_1,C_{-1}}(\eta, f) = \eta\phi_1(f) + (1-\eta)\phi_{-1}(-f) \tag{V.21}$$

is minimized by the predictor

$$f^*_{\phi,C_1,C_{-1}}(\eta) = \arg\min_f C_{\phi,C_1,C_{-1}}(\eta, f) \tag{V.22}$$

leading to the minimum conditional risk

$$\begin{aligned} C^*_{\phi,C_1,C_{-1}}(\eta) &= \eta\phi_1(f^*_{\phi,C_1,C_{-1}}(\eta)) \\ &\quad + (1-\eta)\phi_{-1}(-f^*_{\phi,C_1,C_{-1}}(\eta)). \end{aligned} \tag{V.23}$$

V.C.2 Cost-sensitive learning algorithms

It is currently not known which loss functions $\phi_i(\cdot)$ in (V.20) best extend the ones used in the design of cost-insensitive algorithms, so as to produce cost-sensitive extensions of boosting, or SVM classifiers. We address this problem by extending the approach of Chapter II.

Theorem 40. *Let $g(\eta)$ be any invertible function, $J(\eta)$ any convex function, and $\phi_i(\cdot)$ determined by the following steps:*

1. *use (V.11) and (V.12) to obtain the $I_1(\eta)$ and $I_{-1}(\eta)$, and let $C_{\phi,C_1,C_{-1}}(\eta, f)$ be defined by (V.21).*
2. *set $\phi_1(g(\eta)) = -I_1(\eta)$ and $\phi_{-1}(-g(\eta)) = -I_{-1}(\eta)$.*

Then $g(\eta) = f^*_{\phi,C_1,C_{-1}}(\eta)$ if and only if $J(\eta) = -C^*_{\phi,C_1,C_{-1}}(\eta)$.

Proof. From 1. and Theorem 38, it follows that

$$\eta I_1(\hat{\eta}) + (1-\eta)I_1(\hat{\eta})$$

has maximum value $J(\eta)$, when $\hat{\eta} = \eta$. From 2. the same holds for

$$-\eta\phi_1(g(\hat{\eta})) - (1-\eta)\phi_{-1}(-g(\hat{\eta}))$$

and

$$J(\eta) = -\eta\phi_1(g(\eta)) - (1-\eta)\phi_{-1}(-g(\eta)).$$

It follows from (V.21)-(V.23) that, $g(\eta) = f^*_{\phi,C_1,C_{-1}}(\eta)$ if and only if $J(\eta) = -C^*_{\phi,C_1,C_{-1}}(\eta)$.

■

The theorem shows that any loss with components $\phi_i(\cdot)$ designed according to steps 1. and 2. satisfies (V.21)-(V.23), when $g(\eta) = f^*_{\phi,C_1,C_{-1}}(\eta)$ and $J(\eta) = -C^*_{\phi,C_1,C_{-1}}(\eta)$. This implies that it is possible to specify any pair

$f^*_{\phi,C_1,C_{-1}}(\eta)$, $C^*_{\phi,C_1,C_{-1}}(\eta)$ and derive the underlying loss. The next question is how to choose the best pair of $f^*_{\phi,C_1,C_{-1}}(\eta)$, and $C^*_{\phi,C_1,C_{-1}}(\eta)$.

The following theorem provides a sufficient condition for the Bayes optimality of $f^*_{\phi,C_1,C_{-1}}(\eta)$.

Theorem 41. *Any invertible predictor $f(\eta)$ with symmetry*

$$f^{-1}(-v) = \frac{2C_{-1}}{C_1 + C_{-1}} - f^{-1}(v) \tag{V.24}$$

satisfies the necessary and sufficient conditions for cost-sensitive optimality of (V.5) with $\gamma = \frac{C_{-1}}{C_1+C_{-1}}$.

Proof. Assume that $f(\eta) = v$ is monotonically increasing. Note that $f^{-1}(0) = \frac{C_{-1}}{C_1+C_{-1}}$ which along with $\eta = f^{-1}(v)$ leads to $f(\frac{C_{-1}}{C_1+C_{-1}}) = 0$. If $\eta > \frac{C_{-1}}{C_1+C_{-1}}$ then from (V.24) we have $f^{-1}(-v) < \frac{C_{-1}}{C_1+C_{-1}}$, applying (V.24) again it follows that $f(\eta) > \frac{C_{-1}}{C_1+C_{-1}}$. Similarly, if $\eta < \frac{C_{-1}}{C_1+C_{-1}}$ then $f(\eta) < \frac{C_{-1}}{C_1+C_{-1}}$. ■

Hence, the specification of $f^*_{\phi,C_1,C_{-1}}(\eta)$ as any predictor that satisfies (V.24) guarantees that the conditional risk is minimized by the cost-sensitive Bayes decision rule. The specification of $C^*_{\phi,C_1,C_{-1}}(\eta)$ determines the risk of the optimal classifier. The goal is to approximate as best as possible the cost-sensitive Bayes risk, given in (V.19). The next theorem highlights some fundamental properties of this risk.

Theorem 42. *The risk of (V.19) has the following properties:*

1. *a maximum at $\eta^* = \frac{C_{-1}}{C_1+C_{-1}}$*
2. *symmetry defined by, $\forall \epsilon \in \left[0, \frac{1}{C_1+C_{-1}}\right]$,*

$$C^*\left(\eta^* - C_{-1}\epsilon\right) = C^*\left(\eta^* + C_1\epsilon\right), \tag{V.25}$$

Proof. Note that (V.19) can be written as

$$C^*_{C_1,C_{-1}}(\eta) = \begin{cases} C_{-1}(1-\eta), & \text{if } f^* \geq 0; \\ C_1\eta, & \text{if } f^* < 0, \end{cases} \tag{V.26}$$

The two lines $C_{-1}(1-\eta)$ and $C_1\eta$ intersect and form the maximum at $\eta = \frac{C_{-1}}{C_1+C_{-1}}$.

When $\epsilon = 0$ we have the trivial case of $C^*\left(\frac{C_{-1}}{C_1+C_{-1}}\right) = C^*\left(\frac{C_{-1}}{C_1+C_{-1}}\right)$.

When $0 < \epsilon \leq \frac{1}{C_1+C_{-1}}$ we have $\eta = \frac{C_{-1}}{C_1+C_{-1}} - C_{-1}\epsilon < \frac{C_{-1}}{C_1+C_{-1}}$ in which case from (V.5), $f^* < 0$ and

$$C^*_{C_1,C_{-1}}(\eta) = C_1\eta = C_1\left(\frac{C_{-1}}{C_1+C_{-1}} - C_{-1}\epsilon\right) = \frac{C_1C_{-1}}{C_1+C_{-1}} - C_1C_{-1}\epsilon \quad \text{(V.27)}$$

When $0 < \epsilon \leq \frac{1}{C_1+C_{-1}}$ we also have $\eta = \frac{C_{-1}}{C_1+C_{-1}} + C_1\epsilon > \frac{C_{-1}}{C_1+C_{-1}}$ in which case from (V.5), $f^* > 0$ and

$$\begin{aligned} C^*_{C_1,C_{-1}}(\eta) &= C_{-1}(1-\eta) = C_{-1}\left(1 - \frac{C_{-1}}{C_1+C_{-1}} - C_1\epsilon\right) \\ &= \frac{C_1C_{-1}}{C_1+C_{-1}} - C_1C_{-1}\epsilon \end{aligned} \quad \text{(V.28)}$$

Thus proving that

$$\begin{aligned} C^*_{C_1,C_{-1}}\left(\frac{C_{-1}}{C_1+C_{-1}} - C_{-1}\epsilon\right) &= C^*_{C_1,C_{-1}}\left(\frac{C_{-1}}{C_1+C_{-1}} + C_1\epsilon\right) \\ &= \frac{C_1C_{-1}}{C_1+C_{-1}} - C_1C_{-1}\epsilon \end{aligned} \quad \text{(V.29)}$$

■

As noted by the following lemma, property 2. is in fact a generalization of property 1.

Lemma 43. *Any concave function with the symmetry of (V.25) also has property 1. of Theorem 42.*

Proof. Taking the derivative of (V.25) at $\epsilon = 0$ leads to

$$C^{*'}\left(\frac{C_{-1}}{C_1+C_{-1}}\right)(-C_{-1}) = C^{*'}\left(\frac{C_{-1}}{C_1+C_{-1}}\right)(C_1) \quad \text{(V.30)}$$

which is satisfied only when $C^{*'}\left(\frac{C_{-1}}{C_1+C_{-1}}\right) = 0$. Given that C^* is a concave function, C^* is maximum at $\eta = \frac{C_{-1}}{C_2+C_{-1}}$. ■

Property 1. assigns the largest risk to the locations on the classification boundary. This can be seen as a minimal requirement for consistency of any $C^*_{\phi,C_1,C_{-1}}(\eta)$ with Bayesian decision theory. Enforcing Property 2. further guarantees that the optimal risk has the symmetry of the cost-sensitive Bayes risk. Theorem 42 hence suggests the following risk taxonomy.

Definition 9. *A minimum risk $C^*_{\phi,C_1,C_{-1}}(\eta)$ is of*

1. *Type-I if it satisfies property 1. but not 2. of Theorem 42.*

2. *Type-II if it satisfies both properties 1. and 2.*

Risks of type-II are closer approximations to the cost-sensitive Bayes risk than those of type I.

The combination of Theorems 40-42 leads to a generic procedure for the design of cost-sensitive classification algorithms, consisting of the following steps

1. select a predictor $f^*_{\phi,C_1,C_{-1}}(\eta)$ that satisfies (V.24).

2. select a concave minimum conditional risk $C^*_{\phi,C_1,C_{-1}}(\eta)$ of type-I or type-II, which reduces to $C^*_\phi(\eta)$ when $C_1 = C_{-1} = 1$.

3. use (V.11) and (V.12) with $J(\eta) = -C^*_{\phi,C_1,C_{-1}}(\eta)$ to obtain $I_1(\eta)$ and $I_{-1}(\eta)$.

4. find $\phi_i(\cdot)$ so that $I_1(\eta) = -\phi_1(f^*_{\phi,C_1,C_{-1}}(\eta))$ and $I_{-1}(\eta) = -\phi_{-1}(-f^*_{\phi,C_1,C_{-1}}(\eta))$.

5. derive an algorithm to minimize the conditional risk of (V.21).

We next illustrate the practical application of this framework by showing that the cost-sensitive exponential loss [56, 58] can be derived from a minimal conditional risk of Type-I. This loss function will later be used in Chapter VI to derive the cost sensitive AdaBoost and cost sensitive RealBoost algorithms.

V.C.3 Cost-sensitive exponential loss

We start by recalling that AdaBoost is based on the loss $\phi(yf) = \exp(-yf)$, for which it can be shown that

$$\begin{aligned} C^*_\phi(\eta) &= \eta\sqrt{\frac{1-\eta}{\eta}} + (1-\eta)\sqrt{\frac{\eta}{1-\eta}} \\ \text{and} \quad f^*_\phi &= \frac{1}{2}\log\frac{\eta}{1-\eta}. \end{aligned} \tag{V.31}$$

A natural cost-sensitive extension is $f^*_{\phi,C_1,C_{-1}}(\eta) = \frac{1}{C_1+C_{-1}}\log\frac{\eta C_1}{(1-\eta)C_{-1}}$, which is easily shown to satisfy (V.24). Noting that $C^*_\phi(\eta) = \eta\exp(-f^*_\phi) + (1-\eta)\exp(f^*_\phi)$, suggests the cost-sensitive extension

$$\begin{aligned} C^*_{\phi,C_1,C_{-1}}(\eta) \quad &= \eta\left(\frac{\eta C_1}{(1-\eta)C_{-1}}\right)^{\frac{-C_1}{C_1+C_{-1}}} + \\ &(1-\eta)\left(\frac{\eta C_1}{(1-\eta)C_{-1})}\right)^{\frac{C_{-1}}{C_1+C_{-1}}}. \end{aligned} \tag{V.32}$$

This does not have the symmetry of (V.25) but satisfies property 1. of Theorem 42. Hence, it is a Type-I risk. It is also equivalent to (V.31) when $C_1 = C_{-1} = 1$. Finally, steps 1. and 2. of Theorem 40 produce the loss

$$\phi_{C_1,C_{-1}}(yf) = \begin{cases} \exp(-C_1 f), & \text{if } y = 1 \\ \exp(C_{-1} f), & \text{if } y = -1 \end{cases} \tag{V.33}$$

proposed in [56, 58]. The resulting cost-sensitive boosting algorithm currently holds the best performance in the literature and will be discussed in greater detail in Chapter VI.

V.D Cost sensitive SVM

We next consider the case of the cost-sensitive SVM. We start by extending the hinge loss, using the framework of the previous section, and then derive the cost-sensitive SVM optimization problem.

V.D.1 Cost-sensitive hinge-loss

We start by recalling that the SVM minimizes the risk of the hinge loss $\phi(yf) = \lfloor 1 - yf \rfloor_+$, where $\lfloor x \rfloor_+ = \max(x, 0)$. This risk is minimized by [119]

$$f^*_\phi(\eta) = sign(2\eta - 1) \tag{V.34}$$

leading to the minimum conditional risk

$$\begin{aligned} &C^*_\phi(\eta) = 1 - |2\eta - 1| \\ &= \eta\lfloor 1 - sign(2\eta - 1)\rfloor_+ + (1-\eta)\lfloor 1 + sign(2\eta - 1)\rfloor_+. \end{aligned}$$

Again, we replace the optimal cost-insensitive predictor by its cost-sensitive counterpart

$$f^*_{\phi,C_1,C_{-1}}(\eta) = sign((C_1 + C_{-1})\eta - C_{-1}). \tag{V.35}$$

which is easily shown to satisfy (V.5). This suggests the cost-sensitive minimum conditional risk

$$\begin{aligned} &C^*_{\phi,C_1,C_{-1}}(\eta) = \\ &\eta\lfloor e - d \cdot sign((C_1 + C_{-1})\eta - C_{-1})\rfloor_+ + \\ &(1-\eta)\lfloor b + a \cdot sign((C_1 + C_{-1})\eta - C_{-1})\rfloor_+, \end{aligned} \tag{V.36}$$

which can be shown to satisfy (V.25) if and only if

$$d \geq e \qquad a \geq b \quad \text{and} \quad \frac{C_{-1}}{C_1} = \frac{a+b}{d+e}. \tag{V.37}$$

After steps 1. and 2. of Theorem 40,

$$\phi_{C_1,C_{-1}}(yf) = \begin{cases} \lfloor e - df \rfloor_+, & \text{if } y = 1 \\ \lfloor b + af \rfloor_+, & \text{if } y = -1. \end{cases} \tag{V.38}$$

This loss has four degrees of freedom, which control the margin and slope of the hinge components associated with the two classes: positive examples are classified with margin $\frac{e}{d}$ and hinge loss slope d, while for negative examples the margin is $\frac{b}{a}$ and slope a.

V.D.2 Cost-sensitive SVM learning

We consider the case where errors in the positive class are weighted more heavily, leading to the inequalities $\frac{b}{a} \leq \frac{e}{d}$ and $d \geq a$. Choosing $e = d = C_1$ normalizes the margin of positive examples to unity ($\frac{e}{d} = 1$). Selecting $b = 1$ then fixes the scale of the negative component of the hinge loss, leading to $a = 2C_{-1} - 1$. The resulting cost sensitive SVM minimal conditional risk is

$$\begin{aligned} &C^*_{\phi,C_1,C_{-1}}(\eta) = \\ &\eta\lfloor C_1 - C_1 \cdot sign((C_1 + C_{-1})\eta - C_{-1})\rfloor_+ + \\ &(1-\eta)\lfloor 1 + (2C_{-1} - 1) \cdot sign((C_1 + C_{-1})\eta - C_{-1})\rfloor_+ \end{aligned} \tag{V.39}$$

with $C_{-1} \geq 1$ and $C_1 \geq 2C_{-1} - 1$, so as to satisfy (V.37). Figure V.1 presents plots of (V.39) and (V.38), for both $C_1 = 4$, $C_{-1} = 2$ and the cost insensitive case of $C_1 = 1$, $C_{-1} = 1$ (standard SVM). Note that, for the cost-sensitive SVM, the positive class has a unit margin, while the negative class has a smaller margin of $\frac{1}{3}$. Also, the slope of the positive component of the loss is 4 while the negative component has a smaller slope of 3. In this way, the loss assigns a higher cost to errors in the positive class when the data is not separable, while enforcing a larger margin for positive examples when the data is separable.

Replacing the standard hinge loss with (V.38) in the standard SVM risk [67]

$$\begin{aligned} &\arg\min_{w,b} \sum_{\{i|y_i=1\}} \lfloor C_1 - C_1(w^T x_i + b)\rfloor_+ \\ &+ \sum_{\{i|y_i=-1\}} \lfloor 1 + (2C_{-1} - 1)(w^T x_i + b)\rfloor_+ + \mu||w||^2, \end{aligned} \tag{V.40}$$

leads to the primal problem

$$\begin{aligned} &\arg\min_{w,b} \frac{1}{2}||w||^2 + C\left[\beta \sum_{\{i|y_i=1\}} \xi_i + \lambda \sum_{\{i|y_i=-1\}} \xi_i\right] \\ &\text{s.t.}(w^T x_i + b) \geq 1 - \xi_i; \quad y_i = 1 \\ &(w^T x_i + b) \leq -\kappa + \xi_i; \quad y_i = -1 \end{aligned} \tag{V.41}$$

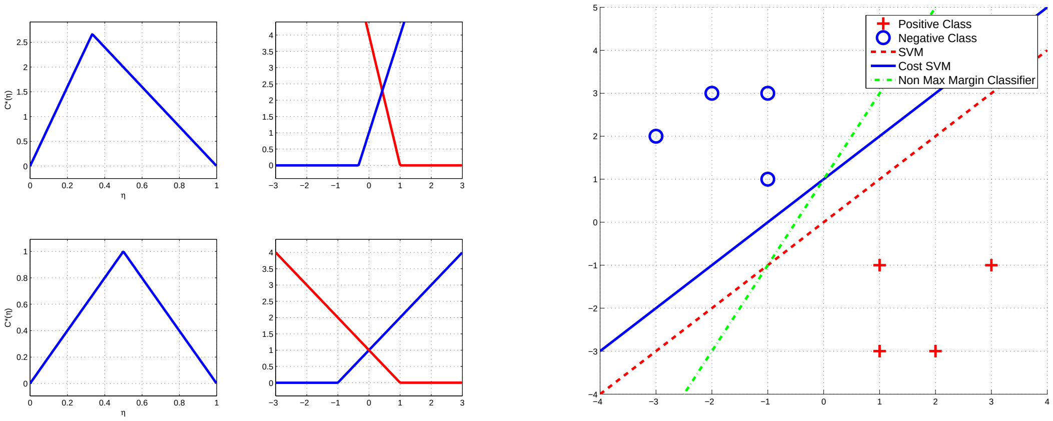


Figure V.1 Left: concave $C^*_{\phi,C_1,C_{-1}}(\eta)$ function and corresponding cost sensitive SVM loss function, top: $C_1 = 4$, $C_{-1} = 2$, bottom: $C_1 = C_{-1} = 1$. Right: linearly separable cost sensitive SVM.

with

$$\beta = C_1 \qquad \lambda = 2C_{-1} - 1 \qquad \kappa = \frac{1}{2C_{-1} - 1}. \tag{V.42}$$

This is a quadratic programming problem similar to that of the standard cost-insensitive SVM with soft margin weight parameter C. In this case, cost-sensitivity is controlled by the parameters β, λ, and κ. The parameter κ is responsible for cost-sensitivity in the separable case. Under the constraints $C_1 \geq 1$, $C_1 \geq 2C_{-1} - 1$ of a type-II risk, it imposes a smaller margin on negative examples. On the other hand, β and λ control the relative weights of margin violations, assigning more weight to positive violations. This allows control of cost-sensitivity when the data is not separable.

Obviously, this primal problem could be defined through heuristic arguments. However, it would be difficult to justify precise choices for the parameters of (V.42). Furthermore, the derivation above guarantees that the optimal classifier implements the Bayes decision rule of (V.5) with $\gamma = \frac{C_{-1}}{C_1 + C_{-1}}$, and its risk is a type-II approximation to the cost-sensitive Bayes risk. No such guarantees would be possible for an heuristic solution.

To obtain some intuition about the cost-sensitive extension, we consider the synthetic problem of Figure V.1, where the two classes are linearly separable. The figure shows three separating lines. The green line is an arbitrary separating line that does not maximize the margin. The red line is the standard SVM solution, which has maximum margin and is equally distant from the nearest examples of the two classes. The blue line is the solution of (V.41) for $C_1 = 4$ and $C_{-1} = 2$ (the C parameter is irrelevant when the data is separable). It is also a maximum margin solution, but trades-off the distance to positive and negative examples so as to enforce a larger positive margin, as specified. Overall, an increase in C_{-1} guarantees a larger positive margin. For a given C_{-1}, increasing C_1 (so that $C_1 \geq 2C_{-1} - 1$) increases the cost of errors on positive examples, enabling control of the miss rate when the classes are not separable.

Finally, the dual and kernelized formulation of the cost sensitive SVM can be obtained with the standard procedures, leading to

$$\begin{aligned} \arg\max_{\alpha_i} & \sum_i \alpha_i \left(\frac{y_i + 1}{2} - \frac{y_i - 1}{2(2C_{-1} - 1)} \right) - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad \text{(V.43)} \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq CC_1; \qquad\qquad y_i = 1 \\ & 0 \leq \alpha_i \leq C(2C_{-1} - 1); \quad y_i = -1. \end{aligned}$$

This reduces to the standard SVM dual when $C_1 = C_{-1} = 1$. Note that the derivation of the cost-sensitive SVM from a suitable loss function leads to an algorithm that performs regardless of the separability of the data and slack penalty, unlike the previous BM-SVM and BP-SVM algorithms. The improved performance of CS-SVM on real world data sets is demonstrated in the next section.

V.E Experimental results

The performance of the CS-SVM was evaluated with two sets of experiments. The first was based on ten binary UCI data sets [69]: Pima-diabetes, breast cancer diagnostic, breast cancer prognostic, original Wisconsin breast cancer, liver

Table V.1 Total loss in $ for each method on the German Credit dataset.

Method	CS-SVM	BP-SVM	SVM
Loss $	550$	878$	878$

Table V.2 mean error for each UCI data set and cost sensitive SVM method.

Dataset	Survive	Liver	Echo	Pima	Wisc	Tic	Heart	Diag	Prag	Sonar
CS-SVM	**195.8**	**163.8**	**40**	**313.2**	33.2	536	**68.4**	33.8	**107.2**	**65.6**
BP-SVM	199.6	167.2	43	416	32.8	536	69.4	33.8	115.2	75.2
BM-SVM	201.8	169.2	45	416	32.8	538	73.2	33.8	126	76.4

disorder, sonar, echo-cardiogram, Cleveland heart disease, tic-tac-toe and Haberman's survival. The goal was to learn the SVM of lowest total error rate, given a target detection rate. In all cases, leave one out cross validation was used to find the best cost estimate. We considered detection rates between 80% and 95%, with increments of 2.5%, and set C, C_1, C_{-1} and b (SVM threshold) for each method so as to achieve the smallest false positive rate on the validation set. The total error was computed for each detection rate, and the mean of these errors is reported in Table V.2. Results are reported for the proposed CS-SVM, the BM-SVM [45] and the BP-SVM [9, 51, 23, 113, 19]. While the table confirms the previous observation that the BP-SVM outperforms the BM-SVM [9, 51, 23, 113, 19], none of them matches the CS-SVM. This is most interesting given the fact that CS-SVM has the same computational complexity and number of tuning parameters as the BP-SVM. Overall, CS-SVM has the smallest error on 7 of the 10 datasets, sometimes by a very substantial margin. CS-SVM and BP-SVM have equal error on 2 datasets, and BP and BM-SVMs have a slight advantage on Wisconsin.

The second set of experiments was based on the German Credit data set [37, 69]. This dataset has 700 examples of good credit customers and 300 examples of bad credit customers. Each example is described by 24 attributes, and the goal is to identify bad costumers, to be denied credit. This data set is particularly interesting for cost-sensitive learning because it provides a cost matrix for the different types of errors. Classifying a good credit customer as bad (a false-positive) incurs a loss of 1. Classifying a bad credit customer as good (a miss) incurs a loss of 5. Hence, on this dataset, the leave one out cross validation of CS-SVM and BP-SVM parameters was subject to the constraint $\frac{C_1}{C_{-1}} = 5$. A cost insensitive SVM was also trained. Table V.1 presents the loss achieved by each method. Note that BP-SVM does not produce any improvement with respect to the cost insensitive SVM. On the other hand, the loss achieved with CS-SVM is 328$ smaller, i.e. a substantial reduction of cost by 37.36%.

V.F Summary and discussion

In this chapter, we have extended the probability elicitation view of loss function design introduced in Chapter II to the cost sensitive classification problem. This extension was applied to the SVM problem, so as to produce a cost-sensitive hinge loss function. A cost-sensitive SVM learning algorithm was then derived, as the minimizer of the associated risk. Unlike previous SVM algorithms, the one now proposed enforces cost sensitivity for both separable and non-separable training data, enforcing a larger margin for the preferred class, independent of the choice of slack penalty. It also offers guarantees of optimality, namely classifiers that implement the cost-sensitive Bayes decision rule and approximate the cost-sensitive Bayes risk. Empirical evidence confirms its superior performance, when compared to previous methods.

V.G Acknowledgments

The text of Chapter V, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “Risk minimization, probability elicitation, and cost-sensitive SVMs ” , in *International Conference on Machine Learning (ICML)*, 2010. The dissertation author was a primary researcher and an author of the cited material.

Chapter VI

Cost sensitive boosting

VI.A Introduction

Classification problems such as fraud detection [101], medical diagnosis [106], or object detection in computer vision [102, 97, 93, 81, 74, 88, 5, 79], are naturally cost sensitive [29]. In these problems the cost of missing a target is much higher than that of a false-positive, and classifiers that are optimal under symmetric costs (such as the popular zero-one loss) tend to under perform. The design of optimal classifiers with respect to losses that weigh certain types of errors more heavily than others is denoted as cost-sensitive learning [29]. Current research in this area falls into two main categories. The first aims for generic procedures that can make arbitrary classifiers cost sensitive, by resorting to Bayes risk theory or some other cost minimization strategy [115, 27]. The second attempts to extend particular algorithms, so as to produce cost-sensitive generalizations. Of interest to this work are classifiers obtained by thresholding a continuous function, here denoted as a *predictor*, and therefore similar to the Bayes decision rule (BDR) [107, 28], which is well known to be optimal for both cost-insensitive and cost-sensitive classification. In particular, we consider learning algorithms in the boosting family [33, 15, 35]. These are algorithms that 1) learn a predictor by combining weak classification rules (weak learners), and 2) use a sample re-weighting mechanism to emphasize points that are difficult to classify.

In this chapter, we consider the problem of how to extend loss functions used in boosting algorithms, based on the theory of cost sensitive loss function design from Chapter V, so as to achieve optimal *cost-sensitive* decision rules. The starting point is the observation, by Friedman et al. [35], that in the (asymptotic) limit of infinite training data the predictor which minimizes the exponential loss used by AdaBoost (and many other boosting algorithms) is the ratio of posterior distributions that also appears in the BDR. Convergence to this optimal predictor is, however, not guaranteed *everywhere* for finite training samples. It is, in fact, well known that, in this case, boosting does not produce calibrated estimates of class posterior probabilities [63, 62, 71, 35, 44]. This is due to the emphasis of sample reweighing on the classification boundary: while the boosted predictor converges to the optimal predictor in a small neighborhood of this boundary, it does not approximate the latter well away from it. This does not compromise *cost-insensitive* classification performance, which only requires the two predictors to have the same sign, but impairs *cost-sensitive* performance, which requires a good approximation of the optimal predictor throughout the feature space.

Two conditions are identified as necessary for optimal cost-sensitive boosting: 1) that the expected boosting loss is minimized by the optimal cost-sensitive decision rule, and 2) that empirical loss minimization emphasizes a neighborhood of the target cost-sensitive boundary, rather than that optimal in the cost-insensitive sense. We propose that this is best accomplished by modifying boosting's loss function, so that boosting-style gradient descent can satisfy the two necessary conditions above. This leads to a general framework for the cost-sensitive extension of boosting algorithms. We introduce cost-sensitive versions of the exponential and binomial losses, which underlie AdaBoost [33], RealBoost [35, 86], and LogitBoost [35]. Cost-sensitive extensions of the algorithms are derived, and shown to satisfy the necessary conditions for cost-sensitive optimality. The new algorithms are compared with various cost-sensitive extensions of boosting available in the literature, including AdaCost [30], CSB0, CSB1, CSB2 [94] asymmetric-AdaBoost [102] and AdaC1, AdaC2, AdaC3 [92]. All of these extensions are heuristic, achieving cost-sensitivity by manipulation of AdaBoost's weights and confidence parameters. In most cases it is not clear if, or how, these manipulations modify boosting's loss. This is unlike the framework now proposed, which inherits all properties of cost-insensitive boosting, simply shifting boosting's emphasis from the neighborhood of the cost-insensitive boundary to the neighborhood of the target cost-sensitive boundary.

The performance of the proposed cost-sensitive boosting algorithms is also evaluated through experiments on synthetic problems, and datasets from the

UCI repository [69] and computer vision face [105] and car [1] detection problems. These experiments show that the proposed algorithms do indeed possess cost sensitive optimality, and can meet target detection rates without (sub-optimal) weight manipulation. They are also shown to outperform the previously available cost-sensitive boosting methods, consistently achieving the best results in all experiments. The chapter is organized as follows. In Section VI.B we review the main principles of cost-sensitive classification. Section VI.C then presents a brief review of the standard boosting algorithms and previous attempts at cost-sensitive extensions, discussing their limitations for optimal cost-sensitive classification. The new framework for cost-sensitive boosting is introduced in Section VI.D, where the extensions of AdaBoost, RealBoost, and LogitBoost, are also derived. Finally, empirical evaluation is discussed in Section VI.E, and some conclusions are drawn in Section VI.F.

VI.B Cost-sensitive classification

We start with the fundamentals of cost-sensitive classification. Most concepts apply to multi-way classification, but here we only consider the problem of binary classification, or *detection*.

VI.B.1 Detection

A detector, or binary classifier, is a function $h : \mathcal{X} \rightarrow \{-1, 1\}$ that maps a feature vector $\mathbf{x} = (x_1, \ldots, x_N)^T \in \mathcal{X} \subset \mathbb{R}^N$ into a class label $y \in \{-1, 1\}$. This mapping is implemented as

$$h(\mathbf{x}) = \text{sgn}[f(\mathbf{x})] \tag{VI.1}$$

where $f : \mathcal{X} \rightarrow \mathbb{R}$ is a predictor, and $\text{sgn}[x] = 1$ if $x \geq 0$, and $\text{sgn}[x] = -1$ otherwise. Feature vectors are samples from a random process $\mathbf{X}$ that is described by a probability distribution $P_{\mathbf{X}}(\mathbf{x})$ on $\mathcal{X}$, and labels are samples from a random variable Y of probability distribution $P_Y(y)$, $y \in \{-1, 1\}$. The detector is optimal if it minimizes the risk $R = E_{\mathbf{X},Y}[L(\mathbf{x}, y)]$, where $L(\mathbf{x}, y)$ is a loss function. We consider losses of the form

$$L(\mathbf{x}, y) = \begin{cases} 0, & \text{if } h(\mathbf{x}) = y \\ C_2 & \text{if } y = -1 \text{ and } h(\mathbf{x}) = 1 \\ C_1 & \text{if } y = 1 \text{ and } h(\mathbf{x}) = -1 \end{cases}, \tag{VI.2}$$

with $C_i > 0$. When $C_1 = C_2$ the detector is cost-insensitive, otherwise it is cost-sensitive. The three scenarios accounted by $L(\mathbf{x}, y)$ are denoted as correct decisions ($h(\mathbf{x}) = y$), false positives ($y = -1$ and $h(\mathbf{x}) = 1$), and false-negatives or misses ($y = 1$ and $h(\mathbf{x}) = -1$).

For many cost-sensitive problems, the costs C_1 and C_2 are specified from domain knowledge. For example, in a fraud detection application, prior experience dictates that there is an average cost of C_2 dollars per false positive, while a false negative (miss) will cost $C_1 > C_2$ dollars, on average. In this case, the costs are simply C_2 and C_1. There are, nevertheless, problems in which it is more natural to specify target detection or false-positive rates than costs. The two types of problems can be addressed within a common optimal detection framework.

VI.B.2 Optimal detection

When C_1 and C_2 are specified, the optimal predictor is given by the BDR [107, 28], i.e.

$$f^* = \arg\min_f E_{\mathbf{X},Y}[L(\mathbf{x}, y)]$$

with

$$f^*(\mathbf{x}) = \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})C_1}{P_{Y|\mathbf{X}}(-1|\mathbf{x})C_2}. \tag{VI.3}$$

An alternative specification is in terms of error rates, where the goal is to minimize the false-positive rate of the classifier given a target detection rate. The optimal solution can be obtained with recourse to the Neyman-Pearson Lemma [70]: for any detection rate ξ, the optimal predictor is still (VI.3). However, for a given ξ, the constants (C_1, C_2) must be such that the specified detection rate is met, i.e.

$$\int_{\mathcal{H}} P(\mathbf{x}|y=1)d\mathbf{x} = \xi \tag{VI.4}$$

with

$$\mathcal{H} = \left\{ \mathbf{x} \left| \frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})} > \frac{C_2}{C_1} \right. \right\}.$$

The only difference is that, rather than specifying costs, one has to search for the costs that satisfy (VI.4). This can be done by cross-validation. Since all that matters is C_1/C_2, C_2 can be set to one and the search is one-dimensional. In any case, the optimal detector can be written as

$$h^*_T(\mathbf{x}) = \operatorname{sgn}\left[f^*_0(\mathbf{x}) - T\right] \tag{VI.5}$$

where

$$f^*_0(\mathbf{x}) = \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(-1|\mathbf{x})}, \tag{VI.6}$$

is the optimal cost-insensitive predictor and

$$T = \log \frac{C_2}{C_1}. \tag{VI.7}$$

Hence, for any cost structure (C_1, C_2), cost-sensitive optimality differs from cost-insensitive optimality only through the threshold T: all optimal cost-sensitive rules can be obtained from $f^*_0(\mathbf{x})$ by threshold manipulation. Furthermore, from (VI.4), different thresholds correspond to different detection rates, and threshold manipulation can produce the optimal decision rule at any detection (or false-positive) rate. This is the motivation for the widespread use of receiver operating curves (ROCs) [96, 39, 2], and the tuning of error rates by threshold manipulation.

VI.B.3 Practical detection

In practice, the probabilities of (VI.6) are unknown, and a learning algorithm is used to estimate the predictor $\hat{f}(\mathbf{x}) \approx f^*_0(\mathbf{x})$, producing an approximately optimal cost-sensitive rule

$$\hat{h}_T(\mathbf{x}) = \operatorname{sgn}[\hat{f}(\mathbf{x}) - T]. \tag{VI.8}$$

This, however, does not guarantee good cost-sensitive performance for the particular cost-structure (C_1, C_2) associated with T. In fact, there are no guarantees of the latter *even when the cost-insensitive detector is optimal*, i.e. when $\hat{h}_0(\mathbf{x}) = \text{sgn}[f_0^*(\mathbf{x})]$. While the necessary and sufficient conditions for cost-insensitive optimality are that

$$\begin{aligned} \hat{f}(\mathbf{x}) &= f_0^*(\mathbf{x}) = 0, \;\; \forall \mathbf{x} \in \mathcal{C} && \text{(VI.9)} \\ \text{sgn}[\hat{f}(\mathbf{x})] &= \text{sgn}[f_0^*(\mathbf{x})], \;\; \forall \mathbf{x} \notin \mathcal{C}, && \text{(VI.10)} \end{aligned}$$

where

$$\mathcal{C} = \left\{ \mathbf{x} \middle| \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(-1|\mathbf{x})} = 0 \right\}$$

is the optimal cost-insensitive classification boundary, the optimality of (VI.8) requires that

$$\begin{aligned} \hat{f}(\mathbf{x}) &= f_0^*(\mathbf{x}) = T, \;\; \forall \mathbf{x} \in \mathcal{C}_T && \text{(VI.11)} \\ \text{sgn}[\hat{f}(\mathbf{x}) - T] &= \text{sgn}[f_0^*(\mathbf{x}) - T], \;\; \forall \mathbf{x} \notin \mathcal{C}_T && \text{(VI.12)} \end{aligned}$$

with

$$\mathcal{C}_T = \left\{ \mathbf{x} \middle| \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(-1|\mathbf{x})} = T \right\}.$$

Hence, the necessary condition for cost-sensitive optimality of $\hat{f}$ at any point $\mathbf{x}$ in the boundary $\mathcal{C}_T$, $\hat{f}(\mathbf{x}) = f_0^*(\mathbf{x})$, is much tighter than the sufficient condition for cost-insensitive optimality of $\hat{f}$ at that point, $\text{sgn}[\hat{f}(\mathbf{x})] = \text{sgn}[f_0^*(\mathbf{x})]$.

It follows that threshold manipulation can only produce optimal cost-sensitive detectors for all values of T if $\hat{f}(\mathbf{x}) = f_0^*(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$. Since this is a much more restrictive constraint than the necessary and sufficient conditions, (VI.9) and (VI.10), of cost-insensitive optimality there is, in general, no reason for a cost-insensitive learning algorithm to enforce it. This is, in fact, Vapnik's argument against generative solutions to the classification problem: that there is no point in attempting to learn the optimal predictor everywhere, when it is sufficient to do so on the classification boundary [98].

VI.C Boosting

This chapter addresses the cost-sensitive extension of boosting algorithms. Such algorithms learn a predictor $f(\mathbf{x})$ by linear combination of simple decision rules, or *weak learners* [84], $G_m(\mathbf{x})$

$$f(\mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}). \tag{VI.13}$$

Optimality is defined with respect to some risk, such as the expected exponential loss

$$E_{\mathbf{X},Y}[\exp(-yf(\mathbf{x}))], \tag{VI.14}$$

or the expected negative binomial log-likelihood

$$-E_{\mathbf{X},Y}[y' \log(p(\mathbf{x})) + (1-y') \log(1-p(\mathbf{x}))] \tag{VI.15}$$

where $y' = (y+1)/2 \in \{0,1\}$ is a re-parametrization of y and

$$p(\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{e^{f(\mathbf{x})} + e^{-f(\mathbf{x})}}. \tag{VI.16}$$

Learning is based on a finite sample $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, empirical loss estimates, and iterative selection of weak learners. At iteration m, a weight $w_i^{(m)}$ is assigned to example $(\mathbf{x}_i, y_i)$, reweighing $\mathcal{D}$ to amplify the importance of points that are poorly classified with the current predictor. We next review some popular algorithms in this family, whose cost-sensitive extensions will be later introduced. All of these can be interpreted as gradient descent on a functional space of linear combinations of weak learners, with respect to one of the losses above[60, 36, 117].

VI.C.1 AdaBoost

AdaBoost [33, 34] learns combinations of scaled binary classifiers

$$G_m^{Ada}(\mathbf{x}) = \alpha_m g_m(\mathbf{x}), \tag{VI.17}$$

where $\{\alpha_m\}_{m=1}^{M}$ is a weight sequence and $\{g_m(\mathbf{x})\}_{m=1}^{M}$ a sequence of binary rules, $g_m(\mathbf{x}) : \mathcal{X} \to \{-1, 1\}$, usually implemented with a *decision stump* $g_m(\mathbf{x}) =$

sgn$[\phi_m(\mathbf{x}) - t_m]$, where $\phi_m(\mathbf{x})$ is a feature response (projection of $\mathbf{x}$ along a basis function ϕ_m) and t_m a threshold. The ensemble predictor of (VI.13) is learned by gradient descent with respect to the exponential loss. The direction of largest descent at the m^{th} iteration is [41, 60]

$$g_m(\mathbf{x}) = \arg\min_g (err_{(m)}) \tag{VI.18}$$

where

$$err_{(m)} = \sum_{i=1}^{n} w_i^{(m)} [1 - I(y_i = g_m(\mathbf{x}_i))], \tag{VI.19}$$

is the total error of $g_m(\mathbf{x})$ and $I(\cdot)$ the indicator function

$$I(y = x) = \begin{cases} 1 & y = x \\ 0 & y \neq x. \end{cases} \tag{VI.20}$$

The optimal step size in the descent direction has closed-form

$$\alpha_m = \frac{1}{2} \log \left(\frac{1 - err_{(m)}}{err_{(m)}} \right), \tag{VI.21}$$

and the weights are updated according to

$$w_i^{(m+1)} = w_i^{(m)} e^{-y_i G_m^{Ada}(\mathbf{x}_i)}. \tag{VI.22}$$

VI.C.2 RealBoost

RealBoost [35, 86] is an extension of AdaBoost that produces better estimates of $f_0^*(\mathbf{x})$ by using real-valued weak learners in (VI.13) (in contrast with binary-valued weak learners.) In this case, the direction of greatest descent of the exponential loss is a (re-weighted) log-odds ratio

$$G_m^{real}(\mathbf{x}) = \frac{1}{2} \log \frac{P_{Y|\mathbf{X}}^{(w)}(1|\phi_m(\mathbf{x}))}{P_{Y|\mathbf{X}}^{(w)}(-1|\phi_m(\mathbf{x}))}, \tag{VI.23}$$

where, as before, $\phi_m(\mathbf{x})$ is a feature response to $\mathbf{x}$, and the superscript w indicates that the probability distribution is that of the re-weighted sample. Weights are updated according to

$$w_i^{(m+1)} = w_i^{(m)} e^{-y_i G_m^{real}(\mathbf{x}_i)}. \tag{VI.24}$$

VI.C.3 LogitBoost

LogitBoost is motivated by the following observation, initially made by Friedman et al. [35].

Lemma 44. (Statistical interpretation of boosting.)

The loss $E[\exp(-yf(\mathbf{x}))]$ *is minimized by the symmetric logistic transform of* $P_{Y|\mathbf{X}}(1|\mathbf{x})$*,*

$$f_0^*(x) = \frac{1}{2}\log\frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(-1|\mathbf{x})}. \tag{VI.25}$$

Proof. See [35]. ∎

This implies that both Ada and RealBoost are stage-wise procedures for fitting an additive logistic regression model. Friedman et al. argued that this is more naturally accomplished by stage-wise minimization of (VI.15). At the m^{th} boosting iteration, the optimal step is given by a weighted least squares regression for the weak learner $G_m^{logit}(\mathbf{x})$ that best fits a set of working responses

$$z_i^{(m)} = \frac{y_i' - p^{(m)}(\mathbf{x}_i)}{p^{(m)}(\mathbf{x}_i)(1-p^{(m)}(\mathbf{x}_i))},$$

where $p^{(m)}(\mathbf{x})$ is the probability of (VI.16) based on the predictor of (VI.13) after $m-1$ iterations. The weights are

$$w_i^{(m)} = p^{(m)}(\mathbf{x}_i)(1-p^{(m)}(\mathbf{x}_i)). \tag{VI.26}$$

VI.C.4 Limitations for cost-sensitive learning

We have already seen that the optimal cost-insensitive detector does not require the optimal predictor of (VI.25): it suffices that (VI.13) converges to any function satisfying (VI.9) and (VI.10). While Lemma 44 guarantees that the minimization of the exponential or binomial losses are sufficient to obtain (VI.25), these guarantees are asymptotic, and do not necessarily hold for finite samples. In fact, the large-margin classification theory suggests that good out-of-sample generalization requires a greater accuracy of the approximation inside a neighborhood of the optimal cost-insensitive boundary $\mathcal{C}$ than outside of it. For boosting, the emphasis on the boundary is accomplished through the example re-weighting of (VI.22), (VI.24), or (VI.26). This, however, usually implies that (VI.13) does not converge to the optimal predictor *everywhere*, and is not necessarily a good predictor for *cost-sensitive detection*.

To obtain some intuition, we consider a detection problem with a bounded optimal predictor $f_0^*(\mathbf{x})$. Assume a finite training sample $\mathcal{D}$ and that, as is common in the large-margin literature, sample points from the two classes are separable, i.e. the detector $\text{sgn}[f_0^*(\mathbf{x})]$ has zero classification error on $\mathcal{D}$ (Note that the classification error does not have to be zero in general, only for the particular sample $\mathcal{D}$.) Define the neighborhood $\mathcal{N}(\mathcal{C}) = \{\mathbf{x}; |f_0^*(\mathbf{x})| < \epsilon\}$, where $\epsilon > 0$ is such that $\mathcal{N}(\mathcal{C})$ contains at least one positive and one negative example. Let $\hat{f}^{(m)}(\mathbf{x})$ be the predictor learned by m iterations of boosting, and assume that

$$\hat{f}^{(m)}(\mathbf{x}) = \begin{cases} f_0^*(\mathbf{x}), & \forall \mathbf{x} \in \mathcal{N}(\mathcal{C}) \\ +\infty, & \text{if } f_0^*(\mathbf{x}) > 0 \text{ and } \mathbf{x} \notin \mathcal{N}(\mathcal{C}) \\ -\infty, & \text{if } f_0^*(\mathbf{x}) < 0 \text{ and } \mathbf{x} \notin \mathcal{N}(\mathcal{C}). \end{cases} \tag{VI.27}$$

For both Ada and RealBoost, a simple recursion shows that

$$\frac{w_i^{(m)}}{w_i^{(0)}} = e^{-y_i \sum_{k=1}^{m} G_k(\mathbf{x}_i)} = e^{-y_i \hat{f}^{(m)}(\mathbf{x}_i)}, \tag{VI.28}$$

where we have also used (VI.13). Let the initial weight distribution be uniform, $w_i^{(0)} = 1/n$, as is customary in boosting practice. Since $y_i \hat{f}^{(m)}(\mathbf{x}_i) \geq 0, \forall i \in \mathcal{D}$, it follows that

$$n w_i^{(m)} = e^{-|\hat{f}^{(m)}(\mathbf{x}_i)|}. \tag{VI.29}$$

Similarly, for LogitBoost,

$$\begin{aligned} w_i^{(m)}(\mathbf{x}_i) &= \left(e^{\hat{f}^{(m)}(\mathbf{x}_i)} + e^{-\hat{f}^{(m)}(\mathbf{x}_i)}\right)^{-2} \\ &\approx e^{-2\text{sgn}[\hat{f}^{(m)}(\mathbf{x}_i)]\hat{f}^{(m)}(\mathbf{x}_i)} = e^{-2|\hat{f}^{(m)}(\mathbf{x}_i)|}. \end{aligned} \tag{VI.30}$$

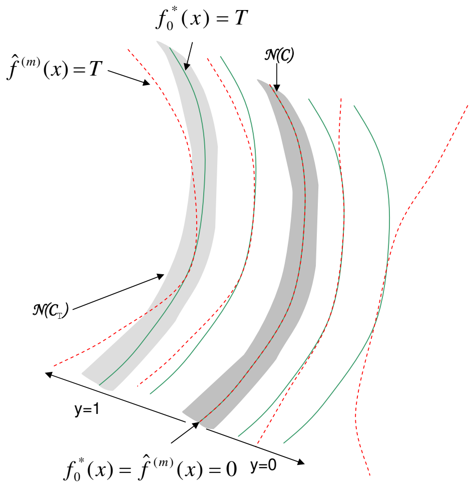

$f_0^*(x) = T$
$\hat{f}^{(m)}(x) = T$
$\mathcal{N}(\mathcal{C})$
$\mathcal{N}(\mathcal{C}_T)$
y=1
$f_0^*(x) = \hat{f}^{(m)}(x) = 0$
y=0

Figure VI.1 Example of a detection problem where boosting produces the optimal cost-insensitive detector but threshold manipulation does not lead to optimal cost-sensitive detectors. The figure presents level-sets of both the optimal predictor $f_0^*(\mathbf{x})$ (solid line) and the boosted predictor $\hat{f}^{(m)}(\mathbf{x})$ (dashed line). While boosting emphasizes the approximation of $f_0^*(\mathbf{x})$ in $\mathcal{N}(\mathcal{C})$, optimal cost-sensitive rules require a good approximation in other regions, e.g. $\mathcal{N}(\mathcal{C}_T)$.

In either case, $nw_i^{(m)}$ or $w_i^{(m)}$ can be seen as a measure of the importance of training point i (relative to the remainder of $\mathcal{D}$). Inside the neighborhood $\mathcal{N}(\mathcal{C})$ this importance is one for points along the *cost-insensitive* boundary $\mathcal{C}$ (where $\hat{f}^{(m)}(\mathbf{x}) = 0$), and decreases exponentially with the distance to it. Outside $\mathcal{N}(\mathcal{C})$ all points have zero importance (because $|\hat{f}^{(m)}(\mathbf{x})| = \infty$). Hence, despite the facts that 1) the predictor is already perfect in $\mathcal{N}(\mathcal{C})$ but 2) approximates $f_0^*(\mathbf{x})$ very poorly outside this neighborhood, all points outside $\mathcal{N}(\mathcal{C})$ are disregarded by subsequent boosting iterations. This implies that the predictor will not get any better in the sense of cost sensitive classification.

The example above turns out not to be a mathematical curiosity. Extensive empirical studies show that, when the span of the space of weak learners is rich enough to separate the training set into the two classes, and boosting is run for enough iterations, all boosting algorithms produce a distribution of posterior probabilities $P_{Y|\mathbf{X}}(y|\mathbf{x})$ highly concentrated around 0 or 1, independently of the true distribution [63, 62]. Note that this does not compromise cost-insensitive optimality: $\hat{f}^{(m)}(\mathbf{x}_i)$ simply grows to ∞ for positive, and to $-\infty$ for negative examples. But the boosted predictor has very poor *cost-sensitive* performance. This problem cannot be addressed by early stopping. In the iterations before class separation, boosting assigns exponentially decaying weight to points correctly classified by previous iterations, in the *cost-insensitive* sense. Hence, points far from $\mathcal{C}$ are exponentially discounted as boosting progresses, creating a soft neighborhood $\mathcal{N}(\mathcal{C})$ of nearby points that dominate the optimization. In result, boosting does not produce accurate posterior estimates, even in this regime [71, 63, 62]. This is, in fact, the reason for the popularity of post-processing boosting's predictions with probability calibration techniques, such as the method of Platt [75], or isotonic regression [116], when posterior accuracy is important [71].

The lack of everywhere convergence to the optimal predictor is illustrated in Fig. Figure VI.1, which depicts $f_0^*(\mathbf{x})$ and $\hat{f}^{(m)}(\mathbf{x})$. Because $f_0^*(\mathbf{x})$ increases (decreases) monotonically to the left (right) of $\mathcal{C}$, any $\hat{f}^{(m)}(\mathbf{x})$ with 1) $\mathcal{C}$ as a zero-level set, and 2) the same monotonicity, satisfies (VI.9)-(VI.10). The emphasis on $\mathcal{N}(\mathcal{C})$ guarantees that the zero-level set of $\hat{f}^{(m)}(\mathbf{x})$ closely approximates $\mathcal{C}$, assuring good cost-insensitive generalization. But the level sets of $\hat{f}^{(m)}(\mathbf{x})$ and $f_0^*(\mathbf{x})$ are not *identical* beyond $\mathcal{N}(\mathcal{C})$. In particular, the set $\hat{f}^{(m)}(\mathbf{x}) = T$ can differ significantly from $f_0^*(\mathbf{x}) = T$, the optimal cost-sensitive boundary $\mathcal{C}_T$ for the cost-structure of threshold T in (VI.5). Hence, threshold manipulation on $\hat{f}^{(m)}(\mathbf{x})$ *does not* produce the optimal cost-sensitive rule of (VI.5).

VI.C.5 Prior work on cost-sensitive boosting

This limitation is well known in the boosting literature, and motivated various cost-sensitive algorithms [30, 94, 102, 92]. Since, for cost-sensitive learning, the main problem is boosting's reweighing emphasis on $\mathcal{N}(\mathcal{C})$, instead of $\mathcal{N}(\mathcal{C}_T)$, it has long been noted that good cost-sensitive performance requires a different reweighing mechanism. This also complies with the intuition that cost-sensitive detection should weigh differently examples from different classes. A naive implementation of this intuition would be to modify the initial boosting weights, so as to represent the cost asymmetry. However, because boosting re-updates all weights at each iteration, it quickly destroys the initial asymmetry, and the predictor obtained after convergence is usually not different from that produced with symmetric initial conditions. A second natural heuristic is to modify the weight update equation. For example, the updated weight could be a mixture of (VI.22), (VI.24), or (VI.26), and the initial cost-sensitive weights. We refer to such heuristics as "weight manipulation". Previously proposed cost-sensitive boosting algorithms, such as AdaCost [30], CSB0, CSB1, CSB2 [94], Asymmetric-AdaBoost [102], AdaC1, AdaC2, or AdaC3 [92], fall in this class. For example, CSB2 [94] modifies the weight update rule of AdaBoost to

$$w_i^{(m+1)} = C_i \cdot w_i^{(m)} e^{-y_i G_m^{Ada}(\mathbf{x}_i)}, \tag{VI.31}$$

relying on (VI.21) for the computation of α_m. While various justifications are available for the different heuristic manipulations of the boosting equations, these manipulations provide no guarantees of asymptotic convergence to a good cost-sensitive decision rule. Furthermore, none of the cost-sensitive extensions can be easily applied to algorithms other than AdaBoost. We next introduce a framework for cost-sensitive boosting that addresses these two limitations.

VI.D Cost-sensitive boosting

The new framework is inspired by two observations. First, the different boosting algorithms are gradient descent methods [60, 36, 117] for empirical minimization of losses that are asymptotically minimized by the cost-insensitive predictor of (VI.25). Second, the main limitation, for cost-sensitive learning, is the emphasis of the empirical loss minimization on a neighborhood $\mathcal{N}(\mathcal{C})$ of the cost-insensitive boundary, as shown in Figure VI.1. These two properties are interconnected. While the limitation is due to the weight-update mechanism, simply modifying this mechanism (as discussed in the previous section) does not guarantee acceptable cost-sensitive performance. Instead, boosting involves a balance between weight updates and descent steps which must be components of the minimization of the *common* loss. For cost-sensitive optimality, this balance requires that the loss function satisfies two conditions, which we denote as the necessary conditions for cost-sensitive optimality.

1. The expected loss is minimized by the optimal cost-sensitive predictor $f^*(\mathbf{x})$ of (VI.3).
2. Empirical loss minimization leads to a weight-updating mechanism that emphasizes a neighborhood of $\mathcal{N}(\mathcal{C}_T)$.

This suggests an alternative strategy for cost-sensitive boosting: *to modify the loss functions so that these two conditions are met*. In what follows, we show how this can be accomplished for Ada, Real and LogitBoost. The framework could be used to derive cost-sensitive extensions of other boosting algorithms, e.g. GentleBoost [35] or AnyBoost [60]. We limit our attention to the ones referred for reasons of brevity, and their popularity.

VI.D.1 Cost-sensitive losses

We start by noting that the optimal cost-sensitive detector of (VI.5) can be re-written as $h^*_T = \text{sgn}[f^*(\mathbf{x})]$ with $f^*(\mathbf{x})$ as in (VI.3). Since the zero level-set of this predictor is the cost-sensitive boundary $\mathcal{C}_T$, boosting-style gradient descent on loss functions asymptotically minimized by $f^*(\mathbf{x})$ should satisfy the two necessary conditions for cost-sensitive optimality. The first is indeed met by the following extensions of the exponential and binomial losses.

Lemma 45. *The expected losses*

$$E_{\mathbf{X},Y}\left[I(y=1)e^{-y.C_1 f(\mathbf{x})} + I(y=-1)e^{-y.C_2 f(\mathbf{x})}\right], \tag{VI.32}$$

$$-E_{\mathbf{X},Y}[y' \log(p_c(\mathbf{x})) + (1-y')\log(1-p_c(\mathbf{x}))] \tag{VI.33}$$

where $I(\cdot)$ is the indicator function of (VI.20) and

$$p_c(\mathbf{x}) = \frac{e^{\gamma f(\mathbf{x})+\eta}}{e^{\gamma f(\mathbf{x})+\eta} + e^{-\gamma f(\mathbf{x})-\eta}}, \tag{VI.34}$$
$$\textit{with } \gamma = \frac{C_1 + C_2}{2}, \;\; \eta = \frac{1}{2}\log\frac{C_2}{C_1},$$

are minimized by the asymmetric logistic transform of $P_{Y|\mathbf{X}}(1|\mathbf{x})$,

$$f(\mathbf{x}) = \frac{1}{C_1 + C_2}\log\frac{P(y=1|\mathbf{x})C_1}{P(y=y''|\mathbf{x})C_2}, \tag{VI.35}$$

where $y'' = -1$ for (VI.32) and $y'' = 0$ for (VI.33).

Proof. See appendix VI.H.1. ■

Note that the cost sensitive extension of the exponential loss was also independently derived in Chapter V.

We next derive cost-sensitive boosting extensions, by gradient descent on empirical loss estimates, and later show that they shift the emphasis of boosting weights from $\mathcal{N}(\mathcal{C})$ to $\mathcal{N}(\mathcal{C}_T)$.

VI.D.2 Cost-sensitive AdaBoost

Theorem 46. (Cost-sensitive AdaBoost) *Consider the minimization of the empirical estimate of the expected loss of (VI.32), based on a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, by gradient descent on the space, $\mathcal{S}$, of functions of the form of (VI.13) and (VI.17), and define two sets*

$$\mathcal{I}_+ = \{i|y_i = 1\} \qquad\qquad \mathcal{I}_- = \{i|y_i = -1\}. \tag{VI.36}$$

The weak learner selected at iteration m consists of an optimal step α_m along the direction g_m of largest descent of the expected loss, and is given by

$$\begin{aligned}(\alpha_m, g_m) = \arg\min_{\alpha,g} &\sum_{i\in\mathcal{I}_+} w_i^{(m)} \exp(-C_1\alpha g(\mathbf{x}_i)) \\ &+ \sum_{i\in\mathcal{I}_-} w_i^{(m)} \exp(C_2\alpha g(\mathbf{x}_i))\end{aligned} \tag{VI.37}$$

with

$$w_i^{(m+1)} = \begin{cases} w_i^{(m)} e^{-C_1 \alpha_m g_m(\mathbf{x}_i)}, & i \in \mathcal{I}_+ \\ w_i^{(m)} e^{C_2 \alpha_m g_m(\mathbf{x}_i)}, & i \in \mathcal{I}_-. \end{cases} \tag{VI.38}$$

The optimal step $\alpha(g)$ along a direction g is the solution of

$$\begin{aligned} 2C_1 \cdot b \cdot \cosh(C_1\alpha) + 2C_2 \cdot d \cdot \cosh(C_2\alpha) = \\ C_1 \cdot \mathcal{T}_+ \cdot e^{-C_1\alpha} + C_2 \cdot \mathcal{T}_- \cdot e^{-C_2\alpha} \end{aligned} \tag{VI.39}$$

with

$$\mathcal{T}_+ = \sum_{i \in \mathcal{I}_+} w_i^{(m)} \qquad \mathcal{T}_- = \sum_{i \in \mathcal{I}_-} w_i^{(m)} \tag{VI.40}$$

$$b = \sum_{i \in \mathcal{I}_+} w_i^{(m)} [1 - I(y_i = g(\mathbf{x}_i))]$$

$$d = \sum_{i \in \mathcal{I}_-} w_i^{(m)} [1 - I(y_i = g(\mathbf{x}_i))] \tag{VI.41}$$

and the descent direction is given by

$$\begin{aligned} g_m = \arg\min_g \Big[(e^{C_1\alpha(g)} - e^{-C_1\alpha(g)}) \cdot b + e^{-C_1\alpha(g)} \mathcal{T}_+ \\ + (e^{C_2\alpha(g)} - e^{-C_2\alpha(g)}) \cdot d + e^{-C_2\alpha(g)} \mathcal{T}_- \Big] \end{aligned} \tag{VI.42}$$

Proof. See appendix VI.H.2. ■

For AdaBoost, possible descent directions are defined by a set of binary classifiers $\{g_k(\mathbf{x})\}_{k=1}^K$. The gradient descent iteration cycles through these, for each solving (VI.39). This can be done efficiently with standard scalar search procedures. In our experiments, the optimal α was found in an average of 6 iterations of bisection search. Given α, the loss associated with the binary classifier is computed and the best classifier selected by (VI.42). A summary of the cost-sensitive boosting algorithm is presented in Algorithm 4. It is worth mentioning that it is fully compatible with AdaBoost, in the sense that it reduces to the latter when $C_1 = C_2 = 1$.

Algorithm 4 Cost-sensitive AdaBoost

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y \in \{1, -1\}$ is the class label of example $\mathbf{x}$, costs $C_1, C_2$, set of binary classifiers $\{g_k(\mathbf{x})\}_{k=1}^K$, and number $M$ of weak learners in the final decision rule.
Initialization: Select uniformly distributed weights for each class
    $w_i = \frac{1}{2|\mathcal{I}_+|}, \forall i \in \mathcal{I}_+, \qquad w_i = \frac{1}{2|\mathcal{I}_-|}, \forall i \in \mathcal{I}_-.$
for $m = \{1, \ldots, M\}$ do
  for $k = \{1, \ldots, K\}$ do
    Compute (VI.40)-(VI.41) with $g(\mathbf{x}) = g_k(\mathbf{x})$ and solve (VI.39) with respect to $\alpha$.
    Use (VI.42) to compute the loss of the weak learner $(g_k(\mathbf{x}); \alpha_k)$ .
  end for
  select the weak learner $(g_m(\mathbf{x}), \alpha_m)$ of smallest loss.
  update weights $w_i$ according to (VI.38).
end for
Output: decision rule $h(x) = \text{sgn}[\sum_{m=1}^M \alpha_m g_m(x)]$.
```

VI.D.3 Cost-sensitive RealBoost

Theorem 47. (Cost-sensitive RealBoost) *Consider the minimization of the empirical estimate of the expected loss of (VI.32), based on a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, by gradient descent on the space, $\mathcal{S}^r$, of predictors of the form of (VI.13) where the weak learners $G_m(\mathbf{x})$ are real functions. Given features $\{\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x})\}$, the direction of largest descent at iteration m has the form*

$$G_m^{real}(\mathbf{x}) = G_{\phi_{k^*}}(\mathbf{x}) \tag{VI.43}$$

where the optimal feature is determined by

$$k^* = \arg\min_k \sum_{i \in \mathcal{I}_+} w_i^{(m)} \exp(-C_1 G_{\phi_k}(\mathbf{x}_i)) + \sum_{i \in \mathcal{I}_-} w_i^{(m)} \exp(C_2 G_{\phi_k}(\mathbf{x}_i)) \tag{VI.44}$$

with weights given by

$$w_i^{(m+1)} = \begin{cases} w_i^{(m)} e^{-C_1 G_m^{real}(\mathbf{x}_i)}, & i \in \mathcal{I}_+ \\ w_i^{(m)} e^{C_2 G_m^{real}(\mathbf{x}_i)}, & i \in \mathcal{I}_-, \end{cases} \tag{VI.45}$$

Algorithm 5 Cost-sensitive RealBoost

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where $y \in \{1, -1\}$ is the class label of example $\mathbf{x}$, costs $C_1, C_2$, and number $M$ of weak learners in the final decision rule.
Initialization: Select uniformly distributed weights for each class
    $w_i = \frac{1}{2|\mathcal{I}_+|}, \forall i \in \mathcal{I}_+,$    $w_i = \frac{1}{2|\mathcal{I}_-|}, \forall i \in \mathcal{I}_-.$
for $m = \{1, \dots, M\}$ do
  for $k = \{1, \dots, K\}$ do
    compute the gradient step $G_{\phi_k}(\mathbf{x})$ with (VI.46).
  end for
  select the optimal direction according to (VI.44) and set the weak learner $G_m^{real}(\mathbf{x})$ according to (VI.43).
  update weights $w_i$ according to (VI.45).
end for
Output: decision rule $h(\mathbf{x}) = \text{sgn}[\sum_{m=1}^{M} G_m^{real}(\mathbf{x})]$.
```

and where

$$G_\phi(\mathbf{x}) = \left\{ \frac{1}{C_1 + C_2} \log \frac{P^{(w)}_{Y|\mathbf{X}}(1|\phi(\mathbf{x}))C_1}{P^{(w)}_{Y|\mathbf{X}}(-1|\phi(\mathbf{x}))C_2} \right\}. \tag{VI.46}$$

$P^{(w)}_{Y|\mathbf{X}}(y|\phi(\mathbf{x})), y \in \{1, -1\}$ *are estimates of the posterior probabilities for the two classes, after the application of the feature transformation* $\phi(\mathbf{x})$ *to a sample re-weighted according to* $w_i^{(m)}$.

Proof. See appendix VI.H.3. ∎

The posterior probabilities $P^{(w)}_{Y|\mathbf{X}}(y|\phi_m(\mathbf{x})), y \in \{1, -1\}$ of (VI.46) can be estimated with standard techniques [28]. For example, using weighted histograms of feature responses if the $\phi_k(\mathbf{x})$ are scalar features. Histogram regularization should be used to avoid empty histogram bins. A summary of cost-sensitive RealBoost is presented in Algorithm 5. This is fully compatible with RealBoost, reducing to it when $C_1 = C_2 = 1$, and has identical computational complexity.

VI.D.4 Cost-sensitive LogitBoost

Finally, we consider LogitBoost.

Theorem 48. (Cost-sensitive LogitBoost) *Consider the minimization, by Newton's method, of the empirical estimate of the expected binomial loss of (VI.33), based on a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, on the space $\mathcal{S}^r$ of predictors of the form of (VI.13) with real-valued weak learners $G_m(\mathbf{x})$. Given a dictionary of features $\{\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x})\}$, and a predictor $\hat{f}^{(m)}(\mathbf{x})$, the Newton step at iteration m has the form*

$$G_m^{logit}(\mathbf{x}) = \frac{1}{2\gamma} G_{\phi_{k^*}}(\mathbf{x}) \tag{VI.47}$$

where $G_\phi(\mathbf{x}) = a_\phi \phi(\mathbf{x}) + b_\phi$ is the result of the weighted regression

$$(a_\phi, b_\phi) = \arg\min_{a_\phi, b_\phi} \sum_i w_i^{(m)} (z_i - a_\phi \phi(\mathbf{x}_i) - b_\phi)^2 \tag{VI.48}$$

with

$$\begin{aligned} z_i &= \frac{y_i' - p_c^{(m)}(\mathbf{x}_i)}{p_c^{(m)}(\mathbf{x}_i)(1 - p_c^{(m)}(\mathbf{x}_i))} && \text{(VI.49)} \\ w_i^{(m)} &= p^{(m)}(\mathbf{x}_i)(1 - p^{(m)}(\mathbf{x}_i)), && \text{(VI.50)} \end{aligned}$$

where $p_c^{(m)}(\mathbf{x})$ is the link function of (VI.34), and $p^{(m)}(\mathbf{x})$ that of (VI.16), with $f(\mathbf{x}) = \hat{f}^{(m)}(\mathbf{x})$. The optimal feature is determined by

$$k^* = \arg\min_k \sum_i w_i^{(m)} (z_i - a_{\phi_k} \phi_k(\mathbf{x}_i) - b_{\phi_k})^2. \tag{VI.51}$$

Proof. See appendix VI.H.4. ∎

A summary of cost-sensitive LogitBoost is presented in Algorithm 6. The algorithm is fully compatible with LogitBoost, in the sense that it reduces to the latter when $C_1 = C_2 = 1$ and has identical computational complexity. It is instructive to compare it with Platt's method for posterior probability calibration [75, 71, 50]. This procedure attempts to map the prediction $f(\mathbf{x}) \in [-\infty, +\infty]$ to a posterior probability $p(\mathbf{x}) \in [0, 1]$, using the link function of (VI.34). The γ

Algorithm 6 Cost-sensitive LogitBoost

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y'_1), \ldots, (\mathbf{x}_n, y'_n)\}$, where $y' \in \{0, 1\}$ is the class label of example $\mathbf{x}$, costs $C_1$, $C_2$, $\gamma = \frac{C_1+C_2}{2}$, $\eta = \frac{1}{2}\log\frac{C_2}{C_1}$, $\mathcal{I}_+$ the set of examples with label 1, $\mathcal{I}_-$ the set of examples with label 0, and number $M$ of weak learners in the final decision rule.
Initialization: Set uniformly distributed probabilities $p_c^{(1)}(\mathbf{x}_i) = p^{(1)}(\mathbf{x}_i) = \frac{1}{2}\ \forall \mathbf{x}_i$ and $\hat{f}^{(1)}(\mathbf{x}) = 0$.
for $m = \{1, \ldots, M\}$ do
  compute the working responses $z_i^{(m)}$ as in (VI.49) and weights $w_i^{(m)}$ as in (VI.50).
  for $k = \{1, \ldots, K\}$ do
    compute the solution to the least squares problem of (VI.48),
      $a_{\phi_k} = \frac{\langle 1 \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$   (VI.52)
      $b_{\phi_k} = \frac{\langle \phi_k(\mathbf{x}_i)^2 \rangle_w \cdot \langle z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$   (VI.53)
    where we have defined
      $\langle q(\mathbf{x}_i) \rangle_w \doteq \sum_i w_i^{(m)} q(\mathbf{x}_i).$
  end for
  select the optimal direction according to (VI.51) and set the weak learner $G_m^{logit}(\mathbf{x})$ according to (VI.47).
  set $\hat{f}^{(m+1)}(\mathbf{x}) = \hat{f}^{(m)}(\mathbf{x}) + G_m^{logit}(\mathbf{x})$.
end for
Output: decision rule $h(\mathbf{x}) = \text{sgn}[\sum_{m=1}^{M} G_m^{logit}(\mathbf{x})]$.
```

and η parameters are determined by gradient descent with respect to the binomial loss of (VI.33), also used in cost-sensitive LogitBoost. The difference is that, in Platt's method, cost-insensitive boosting is first used to learn the predictor $f(\mathbf{x})$ and maximum likelihood is then used to determine the parameters γ and η that best fit a cross-validation data set. On the other hand, cost-sensitive LogitBoost uses the calibrated link function throughout the boosting iterations. Note that, besides requiring an additional validation set, Platt's method does not solve the problem of Figure VI.1, since the emphasis of boosting remains on $\mathcal{N}(\mathcal{C})$, not on $\mathcal{N}(\mathcal{C}_T)$. We next show that all proposed cost-sensitive boosting algorithms solve this problem.

VI.D.5 Cost-sensitive weights

We have mentioned above that cost-sensitive boosting algorithms should

- converge asymptotically to the optimal predictor of (VI.3),
- emphasize a neighborhood of the cost-sensitive boundary $\mathcal{N}(\mathcal{C}_T)$, when learning from finite samples.

The first condition is guaranteed by the losses of (VI.32) and (VI.33). To investigate the second we consider the weight mechanisms of the three algorithms. Let $\hat{f}^{(m)}$ be the boosted predictor after m iterations. For both cost-sensitive Ada and RealBoost, a simple recursion shows that, for correctly classified points,

$$\frac{w_i^{(m)}}{w_i^{(0)}} = e^{-y_i Q_i \hat{f}^{(m)}(\mathbf{x}_i)} = e^{-Q_i |\hat{f}^{(m)}(\mathbf{x}_i)|},$$

where $Q_i = C_1$ if $i \in \mathcal{I}_+$ and $Q_i = C_2$ otherwise. For LogitBoost, the weight $w_i^{(m)}$ is a symmetric function of $p^{(m)}(\mathbf{x}_i)$, with maximum at $p^{(m)}(\mathbf{x}_i) = 1/2$ or, from (VI.16), at $\hat{f}^{(m)}(\mathbf{x}_i) = 0$. As in the cost-insensitive case,

$$w_i^{(m)}(\mathbf{x}) = \left(e^{\hat{f}^{(m)}(\mathbf{x}_i)} + e^{-\hat{f}^{(m)}(\mathbf{x}_i)}\right)^{-2} \approx e^{-2|\hat{f}^{(m)}(\mathbf{x}_i)|}.$$

These equations are qualitatively identical to (VI.29) and (VI.30). The only difference is that, as $\hat{f}^{(m)}(\mathbf{x})$ converges to (VI.35), its zero-level set is the cost-sensitive boundary $\mathcal{C}_T$. Hence, points along $\mathcal{C}_T$ have unitary importance, while the importance of the remaining points decreases exponentially with their distance to $\mathcal{C}_T$. This implies that all cost-sensitive boosting algorithms shift the boosting emphasis from $\mathcal{N}(\mathcal{C})$ to a soft neighborhood of the cost-sensitive boundary $\mathcal{N}(\mathcal{C}_T)$.

VI.E Experimental evaluation

To evaluate the proposed algorithms we started with a synthetic problem, of known BDR, which allows explicit comparison to the optimal cost-sensitive detector. Comparisons against previous methods were then performed with data from the UCI repository and a large face detection dataset. Finally, we compared cost-sensitive boosting and a number of state-of-the-art solutions to the computer vision problem of car detection. Unless otherwise noted, all boosting algorithms used decision stumps as weak learners, and all parameters were selected by cross-validation. The data was divided into train and test sets, and the training set split into five folds, four of which were used for training and one for validation. The latter served to tune parameters (cost parameters and classifier threshold) so as to minimize a classification cost. For car detection, this was the equal error rate (EER), the quantity usually reported for the dataset adopted (UIUC). Elsewhere, it was the number of false positives at a given detection rate. In this case, cross validation was repeated for detection rates between 80% and 95%, with increments of 2.5%. Cross validation was applied to all parameters of all methods. For example, support vector machines (SVMs) required validation of kernel bandwidth, margin/outliers trade-off parameter, and threshold.

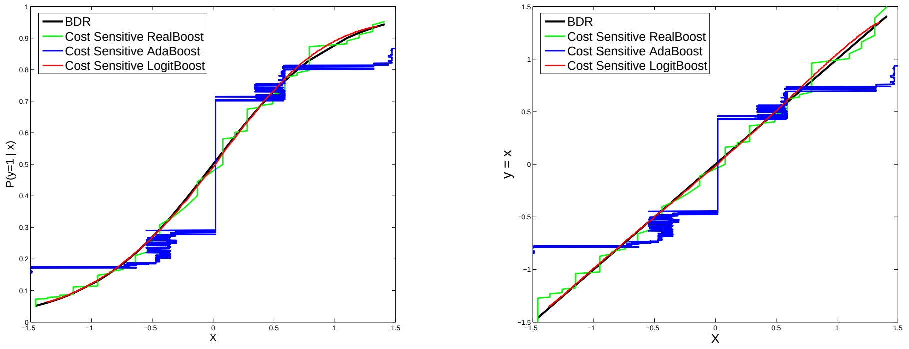
VI.E.1 Synthetic datasets

We start with a synthetic binary scalar problem, involving Gaussian classes of equal variance $\sigma^2 = 1$ and means $\mu_- = -1$ $(y = -1)$ and $\mu_+ = 1$ $(y = 1)$. Ten thousand examples were sampled per class, simulating the scenario where the class probabilities are uniform.

To test the accuracy of the cost-sensitive detectors we relied on the following observations. First, given a cost structure (C_1, C_2), a necessary condition for the optimality of the boosted detector is that the asymmetric logistic transform of (VI.35) holds along the cost-sensitive boundary, i.e. $x^* = f^{-1}(0)$ where $f(x)$ is the optimal predictor of (VI.35) and x^* the zero-crossing of the boosted predictor. Second, from (VI.35), this is equivalent to

$$P_{Y|X}(1|x^*) = \frac{C_2}{C_1 + C_2}. \tag{VI.54}$$

It follows that, given C_1, C_2 and x^*, it is possible to infer the true class posterior probabilities at x^*. This is equally valid for multivariate problems, where x^*



(a)

(b)

Figure VI.2 a) True posterior class probability $P_{Y|X}(y = 1|x)$, as a function of x, and estimates by cost-sensitive Ada, Logit and RealBoost. b) Comparison of the plots $(x^*, -\frac{T}{2})$ and (x^*, x^*).

becomes a level set. Hence, if the boosting algorithm produces truly optimal cost-sensitive detectors, the plots of $\frac{C_2}{C_1+C_2}$ and $P_{Y|X}(1|x^*)$, as functions of x^*, should be identical. For the Gaussian problem considered,

$$P_{Y|X}(1|x) = \frac{1}{1 + e^{-2x}}, \tag{VI.55}$$

and (VI.54) implies that $x^* = -T/2$, with T as in (VI.7). It is thus possible to evaluate the accuracy of the cost-sensitive detectors, for the entire range of (C_1, C_2), by either measuring the similarity between the plots $(x^*, \frac{C_2}{C_1+C_2})$ and $(x^*, \frac{1}{1+e^{-2x^*}})$ or the plots $(x^*, -\frac{T}{2})$ and (x^*, x^*). These are shown on Figure VI.2 for detectors learned with five iterations of cost-sensitive Ada, Real, and LogitBoost. In all cases $C_2 = 1$ and C_1 was varied over a range of values. Both Real and LogitBoost produce near optimal cost-sensitive detectors, but the restriction of the predictor to a combination of binary functions creates difficulties for AdaBoost.

VI.E.2 Real datasets

To evaluate performance on real data, various algorithms were compared on datasets from the UCI repository [69], and the face detection problem [105].

Table VI.1 Average number of errors for each classifier and UCI dataset, across five detection rates. The lowest average error achieved on each dataset is shown in boldface. Rank indicates the average ranking of the classifier across datasets, and #wins is the number of datasets on which a cost sensitive boosting algorithm achieved lower error than all previous boosting methods.

	pima	liver	wdbc	sonar	wpbc	Wisc	echo	heart	tic	survival	Rank	#w
CS-Ada	**205.6**	**143**	26.4	52.2	128.4	37.2	44	**61.4**	433.8	172.8	4.84	6
CS-Log	248.6	146.4	**25.8**	67	**85.6**	35	**40**	74.6	463.2	178.6	5.35	5
CS-Real	256.2	144	32.4	56.8	101.2	35.4	54	69.6	**110.4**	**96.6**	5.35	4
CSB0	241.2	161	43.8	66.6	140.2	40.8	46	89	329.2	101.8	8.2	
CSB1	384	175.8	30.8	65.8	121.8	89	65	100.8	415	188.6	10.95	
CSB2	223	143.5	31	42.6	118.8	45.8	61	88.8	317.4	145.2	6.45	
AdaC2	249.4	162.2	36	56	111.4	42.4	53	64.2	180	131.2	6.65	
AdaC3	250.4	169	29.6	48.2	113.8	39.6	57	102.6	258.6	205.2	8.4	
ADaCost	365	170	42.2	88	111	43.4	65	110	366	189	11.35	
SVM-L	415.2	153.2	32.2	74	111.4	33	43	66.8	550.2	181.4	7.75	
SVM-G	390	161.2	31	**35.8**	122	**30.6**	44	153.6	625	153.6	8.1	
Ada	244.2	168	28.4	57.4	132.8	37.6	48	73.8	465.6	174.6	8.1	
Real	263.8	154.6	32.4	67.2	104.8	35	47	67.6	119	152	6.4	
Logit	263	154	26	68	120.6	33.2	41	68.2	545.8	184.6	7.1	

UCI

Ten data sets were selected - Pima-diabetes, breast cancer diagnostic, breast cancer prognostic, original Wisconsin breast cancer, liver disorder, sonar, echo-cardiogram, Cleveland heart disease, tic-tac-toe, and Haberman's survival. In all cases, data points with missing values were ignored. The multi-class Cleveland heart disease data was converted to the problem of detecting presence (classes 1, 2, 3, 4) vs. absence (value 0) of disease. We compared the performance of the proposed cost-sensitive boosting algorithms (CS-Ada, CS-Real, and CS-Log), their previously available counterparts (CSB0, CSB1, CSB2, AdaC2, AdaC3, and AdaCost), and the combination of standard AdaBoost, RealBoost, or LogitBoost with Platt calibration [75]. Note that, because Asymmetric-AdaBoost [102] and CSB2 [94] are identical, we do not report results for the former. For completeness, we have also tested SVMs with linear and Gaussian kernels, and Platt calibration. In all cases, one point was first removed from the dataset and reserved for testing.

The classifier was trained on the remaining data so as to meet a target detection rate (all parameters cross-validated), and used to classify this test point. The process was iterated, each point taking a turn as test set, and the total number of classification errors recorded.

Table VI.1 presents the average number of errors for each classifier and dataset, across the five detection rates considered. To simplify the comparison, the table includes two overall statistics. The first is the number of datasets in which each cost-sensitive boosting algorithm achieved lower error than *all* prior cost-sensitive boosting algorithms. This is referred to as the number of *wins*. The second is the classifier ranking of [24]: the algorithms were first ranked on each dataset (rank one for lowest error) and the average rank of each classifier, across datasets, is reported. The three cost-sensitive boosting algorithms achieve the three smallest average ranks. From this point of view, only CSB2, AdaC2, and RealBoost with Platt calibration can be seen as competitive with CS-Ada, CS-Real, and CS-Logit. But the worse of the latter has an average rank 15% smaller than the best of the former.

The average ranks, across datasets, for the five detection rates considered, are presented in Table VI.2. While the overall conclusions are the same, note that AdaBoost, RealBoost, and LogitBoost tend to rank lower (relative to their cost-sensitive counterparts) as the detection rate increases. This follows from their cost-insensitivity (despite Platt calibration and threshold tuning). On the other hand, the ranks of CS-AdaBoost, CS-LogitBoost and CS-RealBoost improve relatively. For example, while the difference in rank between AdaBoost and CS-AdaBoost is $7.25 - 6.1 = 1.15$ at 85% detection rate, it grows to $9.5 - 5.2 = 4.3$ at 95%. This confirms our previous claim that threshold manipulation produces inferior results as the distance between cost-sensitive and insensitive boundaries increases.

To investigate the impact of the choice of weak learners in these conclusions, we performed the same experiments with decision trees [14] as weak learners. Following [35], we used four terminal node trees. To enable a comparison to the

Table VI.2 Average classifier rank, across ten UCI datasets, for five detection rates.

Det%	CSAda	Ada	CSLog	Log	CSReal	Real	CSB0	CSB1	CSB2	AdaC2	AdaC3	SVML	SVMG
85%	6.1	7.25	5.6	6.65	**5.3**	5.75	8.85	10.35	6.7	7.8	7.85	8.15	7.45
87.5%	**5.2**	7.2	5.9	6.45	5.5	6.25	8.5	10.7	6.25	6.9	8.7	8.05	7.75
90%	5.45	7.55	5.65	7.5	**4.3**	6.6	7.9	12.1	6.9	6.6	8.55	7.8	7.7
92.5%	5.2	7.9	5.8	7.55	**4.95**	6.6	8.0	11.6	6.25	6.15	8.3	7.8	8.05
95%	5.2	9.5	5.25	7.85	**5.05**	5.2	7.95	10.65	7.25	6.0	8.15	8.55	7.9

results achieved with decision stump methods, we limited the total number of features to 50. Since each tree contains three features, this implies $50/3 \approx 17$ weak learners per classifier. The implementations of CS-AdaBoost and CS-RealBoost relied on (VI.42) and (VI.44), respectively, as tree splitting criteria. All other aspects were identical to [35]. CS-Logit was not considered since it would require the implementation of regression trees, instead of classification trees that we have used. Table VI.3 and Table VI.4 compare the results obtained for the various cost sensitive boosting algorithms, datasets, and detection rates. For completeness, we also implemented a detector based on Random Forests [16] of 17 four terminal node trees and Platt calibration, which did not prove competitive with the proposed algorithms. There is no significant qualitative difference between the results of Table VI.1-Table VI.2, and Table VI.3-Table VI.4, suggesting that the proposed cost-sensitive boosting algorithms have superior performance independently of the weak learner adopted. In summary, with either decision stumps or trees, the proposed algorithms outperform the state-of-the-art in cost-sensitive boosting.

Face detection

UCI datasets are sometimes criticized as too small, or low-dimensional, to allow meaningful conclusions. We repeated the comparisons above on the real, large-scale, large-dimensional problem of face detection. This problem is also becoming an important area of application for cost-sensitive boosting, given the widespread use of boosting for the design of detector cascades [105]. We empha-

Table VI.3 Average number of errors for each classifier and UCI dataset, across five detection rates using decision trees. The lowest average error achieved on each dataset is shown in boldface. Rank indicates the average ranking of the classifier across datasets, and #wins is the number of datasets on which a cost sensitive boosting algorithm achieved lower error than all previous boosting methods.

	pima	liver	wdbc	sonar	wpbc	Wisc	echo	heart	tic	survival	Rank	#w
CS-Ada	**230.6**	**129.4**	**42.2**	63	**95**	37	**46**	**49.4**	343.8	129.6	2.2	6
CS-Real	252.2	148	47.2	**62.6**	95.4	33.2	51	80	297.8	145	3.2	3
CSB0	252	178	42.6	91.6	123.6	46.6	57	74.4	238.2	**109**	4.4	
CSB1	313.4	176	50.4	88.6	112.8	40	62	138.4	490.2	161.6	6.4	
CSB2	299.8	162.8	57.8	83	117	32	50	103	342.2	131.2	4.7	
AdaC2	278.4	151.4	49.4	81	114.8	37.4	64	85.8	185.2	111.8	4.2	
AdaC3	272	163	43	82.6	118	**26.4**	47	82.4	**169.8**	121.8	3.4	
RForest	364.2	189	69.6	102.8	124.4	37.8	60	117.6	546	186	7.5	

size, however, that the goal here is not to compete with algorithms for cascade design, but simply compare cost-sensitive boosting algorithms. While cost-sensitive boosting can be used to design cascade nodes, the overall cascade design requires the solution of additional problems, such as determining the optimal cascade architecture (number of nodes and computation per node), whose solution is beyond the scope of this work. Furthermore, cascade (or face detector) design frequently involves steps, such as bootstrapping (automated collection of negative examples) or manual tuning of classifier parameters, that make objective comparisons of algorithms quite difficult. Our goal is simply to exploit the high-dimensionality of the face detection data ($50,000$ features) and the availability of a large dataset to compare cost-sensitive boosting algorithms in a realistic scenario.

These experiments were based on the experimental protocol of [105]: a face database of 9832 positive and 9832 negative examples, and weak learners based on a combination of decision stumps and Haar wavelet features. 6000 examples were used per class for training, and the remaining 3832 for testing, and all boosting algorithms were trained for 100 iterations. Given the computational complexity of these experiments, we restricted the comparison to CS-Ada and the previously

Table VI.4 Average classifier rank, across ten UCI datasets, for five detection rates using decision trees.

	CS-Ada	CS-Real	CSB0	CSB1	CSB2	AdaC2	AdaC3	RForest
85%	2.3	2.55	4.85	6.25	4.2	4.6	4.0	7.25
87.5%	2.4	3.05	4.7	6.35	3.95	4.3	4.05	7.2
90%	2.65	3.6	4.05	6.5	4.9	4.4	2.7	7.2
92.5%	1.85	3.55	4.3	6.05	5.1	4.4	3.25	7.5
95%	2.2	4.55	4.25	6.0	4.8	3.9	3.1	7.2

Table VI.5 Face detection rate and number of false positives at various cross-validation detection rates.

	85%		87.5%		90%		92.5%		95%	
Method	Det%	#FP	Det%	#FP	Det%	#FP	Det%	#FP	Det%	#FP
CS-Ada	85.2	**22**	87.44	**28**	90.37	**34**	92.64	**52**	95.25	**113**
CSB2	85.2	24	87.7	33	90.29	53	92.82	78	95.14	152
AdaC2	85.54	137	87.91	175	90.52	239	92.77	315	95.22	437
AdaC3	85.93	202	88.39	340	91.96	409	93.21	412	95.25	538
CSB0	86.01	276	88.12	325	90.63	418	92.95	592	97.57	933
CSB1	85.12	689	87.73	803	90.29	967	92.72	1142	95.12	1429

proposed cost-sensitive boosting algorithms (CSB0, CSB1, CSB2, AdaC2, AdaC3). All classifier parameters were tunned with the cross validation procedure described at the start of this section. The detection rate and number of false positives of each method are shown in Table VI.5, for each of the cross-validation detection rates. The number above each pair of columns is the target detection rate (used for cross-validation), while the detection rate and number of false positives measured on the test set are shown in the columns themselves. Note that all methods maintain a test detection rate very similar to the target, CS-Ada achieves the best performance, and only that of CSB2 is comparable. These results illustrate the importance of choosing the confidence α optimally, at each iteration. Methods that ignore α in the weight update rule (CSB0 and CSB1) have extremely poor performance. Methods that update α but are not asymptotically optimal (AdaC2, AdaC3) perform worse than CSB2, which relies on the α updates of AdaBoost.

Table VI.6 Performance on UIUC car dataset, single scale test set. Left side of the table presents methods that rigorously follow the experimental set up of [29] $\dagger$: Use variations of post-processing. $\diamond$: Use extended training set. N.R: Not Reported.

Method	EER	F-Measure	Det%	#FP	Method	EER	F-Measure	Det%	#FP
CS-AdaBoost	**93.5%**	**93.50%**	93.5%	13	Mutch† [68]	99.94%	N.R	N.R	N.R
Shotton [90]	92.8%	N.R	N.R	N.R	Wu$^\diamond$ [109]	97.5%	N.R	N.R	N.R
Bar-Hillel [10]	92.4%	N.R	N.R	N.R	Leibe+MDL$^{\dagger\diamond}$ [48]	97.5%	N.R	N.R	N.R
Leibe[48]	91%	N.R	N.R	N.R	Schneidermann$^\diamond$ [42]	97%	N.R	N.R	N.R
AdaBoost	90%	90.27%	90.5%	20	CS-AdaBoost†	95.5%	95.26%	95.5%	9
Fergus [32]	88.5%	N.R	N.R	N.R	Grabner$^{\dagger\diamond}$ [38]	93%	93.5%	N.R	N.R
Agarwal [1]	79%	77.08%	76.5%	44	AdaBoost†	92.5%	92.23%	92.5%	15

VI.E.3 Car detection

We finish by investigating how the simple application of the proposed cost-sensitive boosting algorithms fare against state-of-the-art object detection algorithms in computer vision. For this, we selected the problem of car detection on the popular UIUC Car dataset [1]. This is a dataset that precisely defines all variables of the experimental evaluation, e.g. a rigorous procedure for counting detections and false positives (which is not the case in [105]), and allows rigorous comparisons to a large literature. It is also a challenging data set, in the sense that only 500 positive and 500 negative examples are available for training. Unfortunately, not all results in the literature comply with the original protocol. For example classifiers are sometimes trained with much larger datasets, and significant variations in error rate can be achieved by optimizing the post-processing procedure (non-maximum suppression) to eliminate the false-positives that always occur in the neighborhood of a correct detection. Hence, even for this thoroughly standardized dataset, assessments of detector performance based on comparison of published results have to be taken with caution. We will discuss these problems in detail below.

We compared CS-Ada to both regular AdaBoost and a number of methods previously proposed in the literature.

All images were re-scaled to 20x50 pixels, and detection based on a pool of $162,000$ Haar features [105]. CS-Ada was used to learn 300 feature detectors, with the cross-validation procedure described at the start of this section. As is advised for this dataset, the resulting detectors were tested with the neighborhood suppression algorithm proposed in [1] and performance quantified by the EER. For completeness, we also indicate the maximum F-measure and corresponding detection and false-positive rates, although these statistics are not always reported in the literature. The F-measure is the weighted harmonic mean of precision and recall, summarizing the trade-off between these two statistics at each point of the ROC curve. The maximum F-measure, and the reported detection and false-positive rates, are those observed at the point where this trade-off is optimal. We limited the comparison to the single scale test set, with the results of Table VI.6.

The left side of the table presents results of methods that rigorously follow the experimental set up of [1]. Agarwal and AdaBoost classify rectangular image patches and can be seen as template classifiers. However, because they rely on highly localized features, they can also be seen as either learning a rough object segmentation (object outline within the patch), or a representation of the object as a spatial configuration of features. Both ideas have been explored in detail in the literature, with classifiers that *explicitly* segment the object to detect [48, 90, 89, 109, 108], learn configurations of its parts [32, 10] or both [90, 109]. Training such representations is manually intensive (e.g. requires precisely segmented examples) and the resulting decision rules have far more computation than those of the AdaBoost/Haar combination. Yet, at least when the protocol of [1] is followed precisely (left half of table), there is little evidence that they have benefits. On the contrary, simply replacing AdaBoost by CS-AdaBoost produces the best overall performance.

There are a number of ways in which performance can be improved by relaxing the experimental protocol. One popular modification is to improve the post-processing of the detector output, so as to eliminate spatially adjacent detections (non-maximum suppression). Methods that use variations of post-processing are identified in the right-side of the table with a †. These variations can lead to a dramatic performance increase. For example, Leibe et al. report an improvement from 91% to 97% EER by introducing their MDL procedure [48]. For the classifiers that we implemented, the simple extension of the suppression window from 71 to 140 pixels (similar to [68] which used 111 pixels for their detector) led to an improvement from 90% to 92.5% for Adaboost and from 93.5% to 95.5% for CS-Adaboost. We have not attempted to optimize performance any further in this way. Another popular performance enhancement strategy is to rely on an extended training set. Variations range from adopting completely different sets of positive and negative training examples [48], to extended sets of positives and negatives (the dataset of [1] plus additional data) [109], to the same set of positives but an extended set of negatives [38, 42]. Methods that rely on such extensions are identified by a $\diamond$ in the table. Given the reduced size of the UIUC car dataset, any of these extensions is likely to improve performance significantly. Unfortunately, they also make it virtually impossible to compare the underlying classification algorithms in an objective manner.

We emphasize that our claim here is not that the combination of CS-AdaBoost and Haar features is the ultimate solution for object detection. In fact, two of the top performing algorithms in each of the sides of Table VI.6 - Bar-Hillel [10] and Wu [109] - rely on the combination of boosting and other image representations (weak learners). It is likely that they could also benefit from the cost-sensitive extensions proposed in this work. What our results show is that 1) for object detection, CS-AdaBoost can lead to substantial performance improvements over AdaBoost, and 2) the combination of CS-AdaBoost and Haar wavelets is at least competitive with the state-of-the-art methods in the literature. This is not insignificant, since most of these competitors involve special purpose features, segmentation, or other vision operations which cost-sensitive boosting does not have access to, and are expensive. On the other hand, the architecture used with cost-sensitive boosting is completely generic, e.g. identical to that used by [105] for face detection.

VI.F Summary and discussion

We have presented a novel framework for the design of cost-sensitive boosting algorithms. The framework is based on the identification of two necessary conditions for the design of optimal cost-sensitive learning algorithms: that 1) expected losses must be minimized by optimal cost-sensitive decision rules, and 2) empirical loss minimization must emphasize the neighborhood of the target cost-sensitive boundary. These enable the derivation of cost-sensitive boosting losses which (similarly to the original cost-insensitive ones) can be minimized by gradient descent, in the functional space of convex combinations of weak learners, to produce boosting algorithms. The proposed framework was used to derive cost-sensitive extensions of AdaBoost, RealBoost and LogitBoost. Experimental evidence, derived from a synthetic problem, standard data sets, and the computer vision problems of face and car detection, was presented in support of the cost-sensitive optimality of the new algorithms. The performance of the latter was also compared to those of various previous cost-sensitive boosting proposals (CSB0, CSB1, CSB2, AdaC1, AdaC2, AdaC3 and AdaCost) as well as the popular combination of large margin classifiers and probability calibration. Cost-sensitive boosting was shown to consistently outperform all other methods tested.

VI.G Acknowledgments

The text of Chapter VI, in full, is based on the material as it appears in: Hamed Masnadi-Shirazi and Nuno Vasconcelos, “Cost-Sensitive Boosting” , in *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2010. The dissertation author was a primary researcher and an author of the cited material.

VI.H Appendix

VI.H.1 Proof of Lemma 45

To find the minimum of the cost-sensitive extension of the exponential loss of (VI.32) it suffices to search for the the function $f(\mathbf{x})$ of minimum expected loss conditioned on $\mathbf{x}$

$$\begin{aligned} l_e(\mathbf{x}) &= E_{Y|\mathbf{x}}\left[I(y=1)e^{-y.C_1 f(\mathbf{x})} + I(y=-1)e^{-y.C_2 f(\mathbf{x})}|\mathbf{x}\right] \\ &= P_{Y|\mathbf{x}}(1|\mathbf{x})e^{-C_1 f(\mathbf{x})} + P_{Y|\mathbf{x}}(-1|\mathbf{x})e^{C_2 f(\mathbf{x})}. \end{aligned}$$

Setting derivatives to zero

$$\begin{aligned} \frac{\partial l_e(\mathbf{x})}{\partial f(\mathbf{x})} &= -C_1 P_{Y|\mathbf{x}}(1|\mathbf{x})e^{-C_1 f(\mathbf{x})} + C_2 P_{Y|\mathbf{x}}(-1|\mathbf{x})e^{C_2 f(\mathbf{x})} \\ &= 0 \end{aligned} \tag{VI.56}$$

it follows that

$$\frac{C_1 P_{Y|\mathbf{x}}(1|\mathbf{x})}{C_2 P_{Y|\mathbf{x}}(-1|\mathbf{x})} = e^{(C_1+C_2)f(\mathbf{x})} \tag{VI.57}$$

and

$$f(\mathbf{x}) = \frac{1}{C_1+C_2}\log\frac{P_{Y|\mathbf{x}}(1|\mathbf{x})C_1}{P_{Y|\mathbf{x}}(-1|\mathbf{x})C_2}. \tag{VI.58}$$

It is straightforward to show that the second derivative is non-negative, from which the loss is minimized by $f(\mathbf{x})$.

To find the minimum of the cost sensitive extension of the binomial loss of (VI.33) it suffices to search for the the function $f(\mathbf{x})$ of minimum expected loss conditioned on $\mathbf{x}$

$$\begin{aligned} l_b(\mathbf{x}) &= -E_{Y|\mathbf{x}}[y'\log(p_c(\mathbf{x})) + (1-y')\log(1-p_c(\mathbf{x}))|\mathbf{x}] \\ &= -P_{Y|\mathbf{x}}(1|\mathbf{x})\log(p_c(\mathbf{x})) - P_{Y|\mathbf{x}}(0|\mathbf{x})\log(1-p_c(\mathbf{x})) \end{aligned}$$

with $p_c(\mathbf{x})$ given by (VI.34). For this, we first compute the minimum with respect to $p_c(\mathbf{x})$, which is given by

$$\frac{\partial l_b(\mathbf{x})}{\partial p_c(\mathbf{x})} = -P_{Y|\mathbf{x}}(1|\mathbf{x})\frac{1}{p_c(\mathbf{x})} + P_{Y|\mathbf{x}}(0|\mathbf{x})\frac{1}{1-p_c(\mathbf{x})} = 0 \tag{VI.59}$$

or

$$\log \frac{p_c(\mathbf{x})}{1 - p_c(\mathbf{x})} = \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(0|\mathbf{x})}.$$

Using (VI.34), this is equivalent to

$$2(\gamma f(\mathbf{x}) + \eta) = \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})}{P_{Y|\mathbf{X}}(0|\mathbf{x})},$$

or

$$f(\mathbf{x}) = \frac{1}{C_1 + C_2} \log \frac{P_{Y|\mathbf{X}}(1|\mathbf{x})C_1}{P_{Y|\mathbf{X}}(0|\mathbf{x})C_2}.$$

Since $\frac{\partial^2 l_b(\mathbf{x})}{\partial p_c(\mathbf{x})^2} \geq 0$ and $p_c(\mathbf{x})$ is monotonically increasing on $f(\mathbf{x})$ this is a minimum.

VI.H.2 Proof of Result 46

From (VI.32) the cost function can be written as

$$J[f] = E_{\mathbf{X},Y}[I(y = 1)\exp(-C_1 f(\mathbf{x})) + I(y = -1)\exp(C_2 f(\mathbf{x}))]$$

and the addition of the weak learner $G(\mathbf{x}) = \alpha g(\mathbf{x})$ to the predictor $f(\mathbf{x})$ results in

$$\begin{aligned} J[f + \alpha g] = E_{\mathbf{X},Y}[\, I(y = 1)w(\mathbf{x}, 1)\exp(-C_1 \alpha g(\mathbf{x})) + \\ I(y = -1)w(\mathbf{x}, -1)\exp(C_2 \alpha g(\mathbf{x}))] \end{aligned}$$

with

$$w(\mathbf{x}, 1) = \exp(-C_1 f(\mathbf{x})) \qquad w(\mathbf{x}, -1) = \exp(C_2 f(\mathbf{x})).$$

Since $J[f + \alpha g]$ is minimized if and only if the argument of the expectation is minimized for all $\mathbf{x}$, the direction of largest descent and optimal step size are the solution of

$$\begin{aligned} (\alpha_m, g_m(\mathbf{x})) \quad = & \\ & \arg \min_{\alpha, g(\mathbf{x})} E_{Y|\mathbf{X}} \left[I(y = 1)w(\mathbf{x}, 1)e^{-C_1 \alpha g(\mathbf{x})} \right. \\ & \left. +I(y = -1)w(\mathbf{x}, -1)e^{C_2 \alpha g(\mathbf{x})} |\mathbf{x} \right]. \end{aligned}$$

Noting that

$$
\begin{aligned}
& E_{Y|\mathbf{x}} \left[I(y=1)w(\mathbf{x},1)e^{-C_1\alpha g(\mathbf{x})} \right. \\
& \qquad \left. + I(y=-1)w(\mathbf{x},-1)e^{C_2\alpha g(\mathbf{x})}|\mathbf{x} \right] \\
& = E_{Y|\mathbf{x}} \left[I(y=1)I(g(\mathbf{x})=1)w(\mathbf{x},1)e^{-C_1\alpha} + \right. \\
& \qquad I(y=1)I(g(\mathbf{x})=-1)w(\mathbf{x},1)e^{C_1\alpha} + \\
& \qquad I(y=-1)I(g(\mathbf{x})=1)w(\mathbf{x},-1)e^{C_2\alpha} + \\
& \qquad \left. I(y=-1)I(g(\mathbf{x})=-1)w(\mathbf{x},-1)e^{-C_2\alpha}|\mathbf{x} \right] \\
& = E_{Y|\mathbf{x}} \left[I(y=1)I(g(\mathbf{x})=-1)w(\mathbf{x},1)(e^{C_1\alpha} - e^{-C_1\alpha}) \right. \\
& \qquad + I(y=1)w(\mathbf{x},1)e^{-C_1\alpha} + \\
& \qquad I(y=-1)I(g(\mathbf{x})=1)w(\mathbf{x},-1)(e^{C_2\alpha} - e^{-C_2\alpha}) \\
& \qquad \left. + I(y=-1)w(\mathbf{x},-1)e^{-C_2\alpha}|\mathbf{x} \right] \\
& = \quad P_{Y|\mathbf{x}}(1|\mathbf{x})w(\mathbf{x},1)I(g(\mathbf{x})=-1)(e^{C_1\alpha} - e^{-C_1\alpha}) \\
& \qquad + P_{Y|\mathbf{x}}(1|\mathbf{x})w(\mathbf{x},1)e^{-C_1\alpha} + \\
& \qquad P_{Y|\mathbf{x}}(-1|\mathbf{x})w(\mathbf{x},-1)I(g(\mathbf{x})=1)(e^{C_2\alpha} - e^{-C_2\alpha}) \\
& \qquad + P_{Y|\mathbf{x}}(-1|\mathbf{x})w(\mathbf{x},-1)e^{-C_2\alpha}
\end{aligned}
$$

it follows that

$$
\begin{aligned}
& (\alpha_m, g_m(\mathbf{x})) = \\
& \arg\min_{\alpha, g(\mathbf{x})} \left\{ P^{(w)}_{Y|\mathbf{x}}(1|\mathbf{x})I(g(\mathbf{x})=-1)(e^{C_1\alpha} - e^{-C_1\alpha}) \right. \\
& \qquad + P^{(w)}_{Y|\mathbf{x}}(1|\mathbf{x})e^{-C_1\alpha} \\
& \qquad + P^{(w)}_{Y|\mathbf{x}}(-1|\mathbf{x})I(g(\mathbf{x})=1)(e^{C_2\alpha} - e^{-C_2\alpha}) \\
& \qquad \left. + P^{(w)}_{Y|\mathbf{x}}(-1|\mathbf{x})e^{-C_2\alpha} \right\}
\end{aligned}
$$

where

$$
P^{(w)}_{Y|\mathbf{x}}(y|\mathbf{x}) = \frac{P_{Y|\mathbf{x}}(y|\mathbf{x})w(\mathbf{x},y)}{\sum_{y\in\{1,-1\}} P_{Y|\mathbf{x}}(y|\mathbf{x})w(\mathbf{x},y)}
$$

are the posterior estimates associated with a sample reweighed according to $w(\mathbf{x}, y)$. Hence, the weak learner of minimum cost is

$$\begin{aligned}
&(\alpha_m, g_m) = \\
&\arg\min_{\alpha,g} E_{\mathbf{X}} \Big\{ P^{(w)}_{Y|\mathbf{X}}(1|\mathbf{x}) I(g(\mathbf{x}) = -1)(e^{C_1\alpha} - e^{-C_1\alpha}) + \\
&\qquad P^{(w)}_{Y|\mathbf{X}}(1|\mathbf{x}) e^{-C_1\alpha} + \\
&\qquad P^{(w)}_{Y|\mathbf{X}}(-1|\mathbf{x}) I(g(\mathbf{x}) = 1)(e^{C_2\alpha} - e^{-C_2\alpha}) + \\
&\qquad P^{(w)}_{Y|\mathbf{X}}(-1|\mathbf{x}) e^{-C_2\alpha} \Big\}
\end{aligned}$$

and, replacing expectations by sample averages,

$$\begin{aligned}
(\alpha_m, g_m) = \arg\min_{\alpha,g} \big[&(e^{C_1\alpha} - e^{-C_1\alpha}) \cdot b + e^{-C_1\alpha} \cdot \mathcal{T}_+ \\
&+ (e^{C_2\alpha} - e^{-C_2\alpha}) \cdot d + e^{-C_2\alpha} \cdot \mathcal{T}_- \big],
\end{aligned}$$

with the empirical estimates $\mathcal{T}_+$, $\mathcal{T}_-$, b and d of (VI.40) - (VI.41). Given $g(\mathbf{x})$, and setting the derivative with respect to α to zero

$$\begin{aligned}
\frac{\partial}{\partial\alpha} &= C_1(e^{C_1\alpha} + e^{-C_1\alpha}) \cdot b - C_1 e^{-C_1\alpha} \cdot \mathcal{T}_+ + \\
&\quad C_2(e^{C_2\alpha} + e^{-C_2\alpha}) \cdot d - C_2 e^{-C_2\alpha} \cdot \mathcal{T}_- = 0
\end{aligned}$$

the optimal step size α is the solution of

$$\begin{aligned}
2C_1 \cdot b \cdot \cosh(C_1\alpha) + 2C_2 \cdot d \cdot \cosh(C_2\alpha) &= \\
C_1 \cdot \mathcal{T}_+ \cdot e^{-C_1\alpha} + C_2 \cdot \mathcal{T}_- \cdot e^{-C_2\alpha}&.
\end{aligned}$$

VI.H.3 Proof of Result 47

From (VI.32) the cost function can be written as

$$J[f] = E_{\mathbf{X},Y}[I(y = 1)\exp(-C_1 f(\mathbf{x})) + I(y = -1)\exp(C_2 f(\mathbf{x}))]$$

and the addition of the weak learner $G(\mathbf{x})$ to the predictor $f(\mathbf{x})$ results in

$$\begin{aligned}
J[f + G] = E_{\mathbf{X},Y}[&I(y = 1)w(\mathbf{x}, 1)\exp(-C_1 G(\mathbf{x})) + \\
&I(y = -1)w(\mathbf{x}, -1)\exp(C_2 G(\mathbf{x}))]
\end{aligned}$$

with

$$w(\mathbf{x}, 1) = \exp(-C_1 f(\mathbf{x})) \tag{VI.60}$$

and

$$w(\mathbf{x}, -1) = \exp(C_2 f(\mathbf{x})). \tag{VI.61}$$

Since $J[f + G]$ is minimized if and only if the argument of the expectation is minimized for all $\mathbf{x}$, and assuming that the weak learners depend on $\mathbf{x}$ only through some feature $\phi(\mathbf{x})$, the optimal weak learner is the solution of

$$\begin{aligned} G_\phi(\mathbf{x}) &= \arg\min_G E_{Y|\mathbf{X}}[I(y=1)w(\mathbf{x},1)\exp(-C_1 G(\mathbf{x})) \\ &\qquad + I(y=-1)w(\mathbf{x},-1)\exp(C_2 G(\mathbf{x}))|\mathbf{x}] \\ &= \arg\min_G P_{Y|\mathbf{X}}(1|\phi(\mathbf{x}))w(\mathbf{x},1)\exp(-C_1 G(\mathbf{x})) \\ &\qquad + P_{Y|\mathbf{X}}(-1|\phi(\mathbf{x}))w(\mathbf{x},-1)\exp(C_2 G(\mathbf{x})) \\ &= \arg\min_G P^{(w)}_{Y|\mathbf{X}}(1|\phi(\mathbf{x}))\exp(-C_1 G(\mathbf{x})) \\ &\qquad + P^{(w)}_{Y|\mathbf{X}}(-1|\phi(\mathbf{x}))\exp(C_2 G(\mathbf{x})) \end{aligned}$$

where

$$P^{(w)}_{Y|\mathbf{X}}(y|\phi(\mathbf{x})) = \frac{P_{Y|\mathbf{X}}(y|\phi(\mathbf{x}))w(\mathbf{x},y)}{\sum_{y\in\{1,-1\}} P_{Y|\mathbf{X}}(y|\phi(\mathbf{x}))w(\mathbf{x},y)}$$

are the posterior estimates associated with a sample reweighed according to $w(\mathbf{x}, y)$. Setting the derivatives of the cost to zero it follows that

$$G_\phi(\mathbf{x}) = \frac{1}{C_1 + C_2}\log\frac{P^{(w)}_{Y|\mathbf{X}}(1|\phi(\mathbf{x}))C_1}{P^{(w)}_{Y|\mathbf{X}}(-1|\phi(\mathbf{x}))C_2}.$$

The optimal feature ϕ^* is the one of smallest minimum cost

$$\begin{aligned} \phi^* &= \arg\min_\phi J[f + G_\phi] \\ &= \arg\min_\phi E_{\mathbf{X},Y}[I(y=1)w(\mathbf{x},1)\exp(-C_1 G_\phi(\mathbf{x})) + \\ &\qquad I(y=-1)w(\mathbf{x},-1)\exp(C_2 G_\phi(\mathbf{x}))] \\ &= \arg\min_\phi \left[\sum_{i\in\mathcal{I}_+} w(\mathbf{x}_i,1)\exp(-C_1 G_\phi(\mathbf{x}_i)) + \right. \\ &\qquad \left. \sum_{i\in\mathcal{I}_-} w(\mathbf{x}_i,-1)\exp(C_2 G_\phi(\mathbf{x}_i))\right]. \end{aligned}$$

Once $G_m^{real}(\mathbf{x})$ is found, the weights are updated so as to comply with (VI.60) and (VI.61), i.e.

$$w(\mathbf{x},1) \leftarrow w(\mathbf{x},1)\exp(-C_1 G_{\phi^*}(\mathbf{x}))$$

and

$$w(\mathbf{x},-1) \leftarrow w(\mathbf{x},-1)\exp(C_2 G_{\phi^*}(\mathbf{x})).$$

VI.H.4 Proof of Result 48

Rewriting the negative log-likelihood as

$$l_b[y', \hat{f}^{(m)}(\mathbf{x})] = -E_{\mathbf{X},Y}\left[y' \log \frac{p_c(\mathbf{x})}{1-p_c(\mathbf{x})} + \log(1-p_c(\mathbf{x}))\right]$$

and using (VI.34), it follows that

$$l_b[y', \hat{f}^{(m)}(\mathbf{x})] = -E_{\mathbf{X},Y}\left[2y'(\gamma \hat{f}^{(m)}(\mathbf{x}) + \eta) - \log\left[1 + e^{2(\gamma \hat{f}^{(m)}(\mathbf{x})+\eta)}\right]\right].$$

This loss is minimized by maximizing the conditional expectation

$$\begin{aligned} &-l_b[y', \hat{f}^{(m)}(\mathbf{x})|\mathbf{x}] = \\ &E_{Y|\mathbf{X}}\left[2y'(\gamma \hat{f}^{(m)}(\mathbf{x}) + \eta) - \log\left[1 + e^{2(\gamma \hat{f}^{(m)}(\mathbf{x})+\eta)}\right]\right] \\ &= 2E_{Y|\mathbf{X}}[y'|\mathbf{x}](\gamma \hat{f}^{(m)}(\mathbf{x}) + \eta) - \log\left[1 + e^{2(\gamma \hat{f}^{(m)}(\mathbf{x})+\eta)}\right] \end{aligned}$$

for all $\mathbf{x}$, i.e. by searching for the weak learner $G(\mathbf{x})$ that maximizes the cost

$$J[\hat{f}^{(m)}(\mathbf{x}) + G(\mathbf{x})] = -l_b[y', \hat{f}^{(m)}(\mathbf{x}) + G(\mathbf{x})|\mathbf{x}].$$

The maximization is done by Newton's method, which requires the computation of the gradient

$$\left.\frac{\partial J[\hat{f}^{(m)}(\mathbf{x}) + G(\mathbf{x})]}{\partial G(\mathbf{x})}\right|_{G(\mathbf{x})=0} = 2\gamma(E_{Y|\mathbf{X}}[y'|\mathbf{x}] - p_c(\mathbf{x}))$$

and Hessian

$$\left.\frac{\partial^2 J[\hat{f}^{(m)}(\mathbf{x}) + G(\mathbf{x})]}{\partial G(\mathbf{x})^2}\right|_{G(\mathbf{x})=0} = -4\gamma^2 p_c(\mathbf{x})(1-p_c(\mathbf{x}))$$

leading to a Newton update

$$G(\mathbf{x}) = \frac{1}{2\gamma} E_{Y|\mathbf{x}} \left[\frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} \right].$$

This is equivalent to solving the least squares problem

$$\min_{G(\mathbf{x})} E_{Y,\mathbf{x}} \left[\left(\frac{1}{2\gamma} \frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} - G(\mathbf{x}) \right)^2 \right],$$

and the optimal weak learner can, therefore, be computed with

$$\begin{aligned}
G^* &= \min_G \int P_{\mathbf{X}}(\mathbf{x}) \sum_{y'=0}^{1} P_{Y|\mathbf{X}}(y'|\mathbf{x}) \left(\frac{1}{2\gamma} \frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} - G(\mathbf{x}) \right)^2 d\mathbf{x} \\
&= \min_G \int P_{\mathbf{X}}(\mathbf{x}) \sum_{y'=0}^{1} \frac{P_{Y|\mathbf{X}}(y'|\mathbf{x}) w(\mathbf{x})}{\sum_{j=0}^{1} P_{Y|\mathbf{X}}(j|\mathbf{x}) w(\mathbf{x})} \left(\frac{1}{2\gamma} \frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} - G(\mathbf{x}) \right)^2 d\mathbf{x} \\
&= \min_G \int P_{\mathbf{X}}(\mathbf{x}) \sum_{y'=0}^{1} P^{(w)}_{Y|\mathbf{X}}(y'|\mathbf{x}) \left(\frac{1}{2\gamma} \frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} - G(\mathbf{x}) \right)^2 d\mathbf{x} \\
&= \min_G E^{(w)}_{Y,\mathbf{X}} \left[\left(\frac{1}{2\gamma} \frac{y' - p_c(\mathbf{x})}{p_c(\mathbf{x})(1 - p_c(\mathbf{x}))} - G(\mathbf{x}) \right)^2 \right]
\end{aligned}$$

which is the weighted least squares regression of z_i to $\mathbf{x}_i$ using weights w_i, as given by (VI.49) and (VI.50). The optimal feature is the one of smallest regression error.

Chapter VII

Conclusions

In this thesis we have presented a new framework for the design of Bayes consistent loss functions and developed a generative method for deriving such loss functions. This has allowed us to effectively design a large number of loss functions with certain novel shapes and properties that are custom tailored for certain classification problems. We have also provided a full analysis and taxonomy of such loss functions. This was achieved by studying and relating the two fields of risk minimization in machine learning and probability elicitation in statistics. Specifically, The class of Bayes consistent loss functions were partitioned into different varieties based on their convexity properties. The convexity properties of the loss and associated risk of Bayes consistent loss functions were also studied in detail which, for the first time, enabled the derivation of non convex Bayes consistent loss functions.

We also developed a fully constructive method for the derivation of novel canonical loss functions. This was due to a simple connection between the associated minimum conditional risk and optimal link functions. The added insight allowed us to derive 1) variable margin extensions of existing losses, 2) new losses from the minimum risks associated with existing non-canonical losses, and 3) new losses from cumulative distribution functions with explicit margin control. We then established a common boosting framework, canonical gradientBoost, for building boosting classifiers from all canonical losses. A number of experiments were conducted to study the effect of margin-control on the classification accuracy of the proposed variable-margin losses.

Next, we extended the probability elicitation view of loss function design to the problem of designing robust loss functions for classification. The robust Savage loss and corresponding SavageBoost algorithm was derived and shown to outperform other boosting algorithms on a set of experiments designed to test the robustness of the algorithms to outliers in the training data. We also argued that a robust loss should penalizes both large positive and large negative margins. The Tangent loss was derived with the desired robust properties. We then derived the associated TangentBoost classifier. This classification algorithm was shown to outperform other boosting algorithms on a variety of test sets involving various computer vision problems, including scene classification, object tracking, recognition, and MIL problems. Empirical evidence confirmed the importance of using robust Bayes consistent loss functions when dealing with noise, outliers and class ambiguity within the data.

We also extended the probability elicitation view of loss function design to the cost sensitive classification problem. A general framework for the derivation of Bayes consistent cost sensitive loss functions was developed. This was then used to derive a novel cost sensitive hinge loss function. A cost-sensitive SVM learning algorithm was then derived, as the minimizer of the associated risk. Unlike previous SVM algorithms, the one now proposed was shown to enforce cost sensitivity for both separable and non-separable training data, enforcing a larger margin for the preferred class, independent of the choice of slack penalty.

Finally, we presented a novel framework for the design of cost-sensitive boosting algorithms. The proposed framework was used to derive cost-sensitive extensions of AdaBoost, RealBoost and LogitBoost. Experimental evidence, over a synthetic problem, standard data sets, and the computer vision problems of face and car detection, was presented in support of the cost-sensitive optimality of the new algorithms and cost-sensitive boosting was shown to consistently outperform all other methods tested.

Bibliography

[1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.

[2] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, "Generalization bounds for the area under the roc curve," *The Journal of Machine Learning Research*, vol. 6, pp. 393–425, 2005.

[3] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *European Conference on Machine Learning (ECML)*, 2004, pp. 39–50.

[4] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[5] Y. Amit and D. Geman, "Shape Quantization and Recognition with Randomized Trees," *Neural Computation*, vol. 9, pp. 1545–1588, 1997.

[6] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2003, pp. 561–568.

[7] S. Avidan, "Ensemble tracking," *IEEE PAMI*, vol. 29, no. 2, pp. 261–271, 2007.

[8] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009.

[9] F. R. Bach, D. Heckerman, and E. Horvitz, "Considering cost asymmetry in learning classifiers," *The Journal of Machine Learning Research*, vol. 7, pp. 1713–1741, 2006.

[10] A. Bar-Hillel and D. Weinshall, "Efficient learning of relational object class models," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 175–198, 2008.

[11] P. Bartlett, M. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *JASA*, 2006.

[12] C. M. Bishop, *Neural networks for pattern recognition.* New York: Oxford University Press Inc., 2004.

[13] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Int. J. Comput. Vision*, 1996.

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Belmont, CA: Wadsworth, 1984.

[15] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[16] ——, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," *(Technical Report) University of Pennsylvania*, 2005.

[18] R. Caruana, A. Niculescu-mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *ICML*, 2004, pp. 137–144.

[19] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[21] M. Collins, "Discriminative reranking for natural language parsing," in *In Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[23] M. Davenport, R. Baraniuk, and C. Scott, "Controlling false alarms with support vector machines," in *ICASSP*, 2006.

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[25] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, 2000.

[26] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1-2, pp. 31–71, 1997.

[27] P. Domingos, “Metacost: a general method for making classifiers cost-sensitive,” in *Knowledge Discovery and Data Mining*, 1999, pp. 155–164.

[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* New York: John Wiley Sons Inc, 2001.

[29] C. Elkan, “The foundations of cost-sensitive learning,” in *Joint Conference on Artificial Intelligence*, 2001.

[30] W. Fan, S. Stolfo, J. Zhang, and P. Chan, “Adacost: Misclassification cost-sensitive boosting,” in *ICML*, 1999.

[31] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *CVPR*, 2008.

[32] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, p. 264.

[33] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, 1997.

[34] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *International Conference on Machine Learning*, 1996, pp. 148–156.

[35] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *Annals of Statistics*, 2000.

[36] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[37] P. Geibel, U. Brefeld, and F. Wysotzki, “Perceptron and svm learning with generalized cost models,” *Intelligent Data Analysis*, vol. 8, pp. 439–455, 2004.

[38] H. Grabner, C. Beleznai, and H. Bischof, “Improving adaboost detection rate by wobble and mean shift,” in *Proceedings Computer Vision Winter Workshop*, 2005, pp. 23–32.

[39] D. Green and J. Swets, *Signal detection theory and psychophysics.* New York: John Wiley and Sons Inc., 1966.

[40] M. Haruno, S. Shirai, and Y. Ooyama, “Using decision trees to construct a practical parser,” *Machine Learning*, vol. 34, p. 131149, 1999.

[41] Hastie, Tibshirani, and Friedman, *The Elements of Statistical Learning.* New York: Springer-Verlag Inc, 2001.

[42] H.Schneiderman, "Feature-centric evaluation for efficient cascaded object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[43] P. J. Huber and E. M. Ronchetti, *Robust Statistics.* New York: John Wiley Sons Inc, 2009.

[44] W. Jiang, "Process consistency for adaboost," *The Annals of Statistics*, vol. 32, pp. 13–29, 2004.

[45] G. Karakoulas and J. Shawe-Taylor, "optimizing classifiers for imbalanced training sets," in *NIPS*, 1999.

[46] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *ICML*, 1997, pp. 179–186.

[47] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006, pp. 2169–2178.

[48] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *European Conference on Computer Vision, Workshop on Statistical Learning in Computer Vision*, May 2004, pp. 17–32.

[49] C. Leistner, A. Saffari, P. M. Roth, and H. Bischof, "On robustness of online boosting - a competitive study," in *IEEE ICCV Workshop on On-line Computer Vision*, 2009.

[50] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[51] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, pp. 191–202, 2002.

[52] J. Liu and M. Shah, "Scene modeling using co-clustering," in *ICCV*, 2007.

[53] R. Maclin and D. Opitz, "An empirical evaluation of bagging and boosting," in *In Proceedings of the Fourteenth National Conference on Artificial Intelligence.* AAAI Press, 1997, pp. 546–551.

[54] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *CVPR*, 2009.

[55] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *ICML*, 1998, pp. 341–349.

[56] H. Masnadi-Shirazi and N. Vasconcelos, "Asymmetric boosting," in *ICML*, 2007.

[57] ——, “On the design of loss functions for classification: theory, robustness to outliers, and savageboost,” in *NIPS*, 2008, pp. 1049–1056.

[58] ——, “Cost-sensitive boosting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, p. 294, 2010.

[59] ——, “Risk minimization, probability elicitation, and cost-sensitive svms,” in *International Conference on Machine Learning (ICML)*, 2010, pp. 759–766.

[60] L. Mason, J. Baxter, P. Bartlett, and M. Frean, “Boosting Algorithms as Gradient Descent,” in *NIPS*, 2000.

[61] R. McDonald, D. Hand, and I. Eckley, “An empirical comparison of three boosting algorithms on real data sets with artificial class noise,” in *International Workshop on Multiple Classifier Systems*, 2003.

[62] D. Mease and A. J. Wyner, “Evidence contrary to the statistical view of boosting,” *JMLR*, 2008.

[63] D. Mease, A. J. Wyner, and A. Buja, “Boosted classification trees and class probability/quantile estimation,” *JMLR*, 2007.

[64] P. Meer, C. V. Stewart, and D. E. Tyler, “Robust computer vision: an interdisciplinary challenge,” *Comput. Vis. Image Underst.*, 2000.

[65] S. Merler, C. Furlanello, B. Larcher, and A. Sboner, “Tuning costsensitive boosting and its application to melanoma diagnosis,” in *In Multiple Classifier Systems: Proceedings of the 2nd International Workshop*, 2001, p. 3242.

[66] R. M.Gray, A. Buzo, and Y. Matsuyama, “Distortion measures for speech processing,” *IEEE Trans. Acoustic, Speech and Sig. Proc*, vol. 28, pp. 367–376, 1980.

[67] J. M. Moguerza and A. Munoz, “Support vector machines with applications,” *Statistical Science*, vol. 21, p. 322, 2006.

[68] J. Mutch and D. G. Lowe, “Object class recognition and localization using sparse features with limited receptive fields,” *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, 2008.

[69] D. Newman, S. Hettich, C. Blake, and C. Merz, “UCI repository of machine learning databases,” 1998. [Online]. Available: http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html

[70] J. Neyman and E. S. Pearson, “On the problem of the most efficient tests of statistical hypotheses.” *Philosophical Transactions of the Royal Society of London*, vol. 231, pp. 289–337, 1933.

[71] A. Niculescu-Mizil and R. Caruana, “Obtaining calibrated probabilities from boosting,” in *Uncertainty in Artificial Intelligence*, 2005.

[72] W. S. Noble, “Support vector machine applications in computational biology,” *Kernel Methods in Computational Biology. B. Schoelkopf, K. Tsuda and J.-P. Vert, ed. MIT Press*, pp. 71–92, 2004.

[73] ——, “What is a support vector machine?” *Nature Biotechnology*, vol. 24(12), pp. 1565–1567, 2006.

[74] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, “Pedestrian Detection Using Wavelet Templates,” in *IEEE Conference in Pattern Recognition and Computer Vision*, 1997.

[75] J. Platt, “Probabilistic outputs for support vector machines and comparison to regularized likelihood methods.” in *Adv. in Large Margin Classifiers*, 2000.

[76] J. Ramon and L. De Raedt, “Multi instance neural networks,” in *ICML*, 2000.

[77] N. Rasiwasia and N. Vasconcelos, “Scene classification with low-dimensional semantic spaces and weak supervision,” in *CVPR*, 2008.

[78] ——, “Holistic context modeling using semantic co-occurrences,” in *CVPR*, 2009.

[79] D. Roth, M. Yang, and N. Ahuja, “Learning to Recognize Three-Dimensional Objects,” *Neural Computation*, vol. 14, pp. 1071–1103, 2002.

[80] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. New York: John Wiley Sons Inc, 2003.

[81] H. A. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–38, 1998.

[82] L. J. Savage, “The elicitation of personal probabilities and expectations,” *JASA*, vol. 66, pp. 783–801, 1971.

[83] H. Sawhney and S. Ayer, “Compact representations of videos through dominant and multiple motion estimation,” *(IEEE) PAMI*, 1996.

[84] R. E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, pp. 197–227, 1990.

[85] ——, “The boosting approach to machine learning: An overview,” *In D.D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, Nonlinear Estimation and Classification. Springer*, 2003.

[86] R. E. Schapire and Y. Singer, "Improved boosting using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[87] R. E. Schapire, Y. Singer, and A. Singhal, "Boosting and rocchio applied to text filtering," *In Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval*, 1998.

[88] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[89] E. Seemann, B. Leibe, and B. Schiele, "Multi-aspect detection of articulated objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1582–1588.

[90] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *IEEE international Conference on Computer Vision*, vol. 1, 2005, pp. 503–510.

[91] D. R. Smith, *Variational Methods in Optimization.* New Jersey: Prentice-Hall Inc, 1974.

[92] Y. Sun, A. K. C. Wong, and Y. Wang, "Parameter inference of cost-sensitive boosting algorithms," in *Machine Learning and Data Mining in Pattern Recognition*, 2005.

[93] K. Sung and T. Poggio, "Example Based Learning for View-Based Human Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, January 1998.

[94] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *ICML*, 2000.

[95] A. Torralba, "Contextual priming for object detection," *Int. J. Comput. Vision*, 2003.

[96] H. L. V. Tree, *Detection, Estimation and Modulation Theory.* New York: John Wiley and Sons Inc, 1968.

[97] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, 1991.

[98] V. N. Vapnik, *Statistical Learning Theory.* John Wiley Sons Inc, 1998.

[99] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *ICCV*, 2007.

[100] S. Viaene, R. A. Derrig, and G. Dedene, "A case study of applying boosting naive bayes to claim fraud diagnosis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 5, pp. 612–620, 2004.

[101] ——, “Cost-sensitive learning and decision making for massachusetts pip claim fraud data,” *International Journal of Intelligent Systems*, vol. 19, pp. 1197–1215, 2004.

[102] P. Viola and M. Jones, “Fast and robust classification using asymmetric adaboost and a detector cascade,” in *NIPS*, 2002.

[103] P. Viola, M. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” in *Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, p. 734.

[104] P. Viola, J. C. Platt, and C. Zhang, “Multiple instance boosting for object detection,” in *NIPS*, 2006.

[105] P. A. Viola and M. J. Jones, “Robust real-time face detection,” *IJCV*, 2004.

[106] A. Vlahou, J. O. Schorge, B. W. Gregory, and R. L. Coleman, “Diagnosis of ovarian cancer using decision tree classification of mass spectral data,” *Journal of Biomedicine and Biotechnology*, vol. 2003, no. 5, p. 308314, 2003.

[107] A. Wald, “Contributions to the theory of statistical estimation and testing hypotheses.” *The Annals of Mathematical Statistics*, vol. 10, pp. 299–326, 1939.

[108] J. Winn and J. Shotton, “The layout consistent random field for recognizing and segmenting partially occluded objects,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 37–44.

[109] B. Wu and R. Nevatia, “Simultaneous object detection and segmentation by boosting local shape feature based classifier,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 18–23.

[110] G. Wu and E. Chang, “Adaptive feature-space conformal transformation for imbalanced data learning,” in *ICML*, 2003, pp. 816–823.

[111] ——, “Kba: kernel boundary alignment considering imbalanced data distribution,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 786–795, 2005.

[112] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” *The Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[113] X. Wu and R. Srihari, “New ν-support vector machines and their sequential minimal optimization,” in *ICML*, 2003.

[114] Y. Wu and Y. Liu, “Robust truncated-hinge-loss support vector machines,” *JASA*, 2007.

[115] B. Zadrozny and C. Elkan, "Learning and making decisions when costs and probabilities are both unknown." in *7th International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 203–213.

[116] ——, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 609–616.

[117] R. S. Zemel and T. Pitassi, "A gradient-based boosting algorithm for regression problems," in *Advances in Neural Information Processing Systems*, 2000, pp. 696–702.

[118] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *NIPS*, 2001, pp. 1073–1080.

[119] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Annals of Statistics*, 2004.

[120] T. Zhang and B. Yu, "Boosting with early stopping: Convergence and consistency," *Annals of Statistics*, 2005.