

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Algorithms for Interactive Machine Learning

Permalink

<https://escholarship.org/uc/item/0r6435qb>

Author

Poulis, Stefanos

Publication Date

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Algorithms for Interactive Machine Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Stefanos Poulis

Committee in charge:

Professor Sanjoy Dasgupta, Chair
Professor Ery Arias-Castro
Professor Kamalika Chaudhuri
Professor Virginia de Sa
Professor Lawrence Saul

2019

Copyright
Stefanos Poulis, 2019
All rights reserved.

The Dissertation of Stefanos Poulis is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

EPIGRAPH

None but ourselves can free our minds.

—Bob Marley

TABLE OF CONTENTS

Signature Page	iii
Epigraph.....	iv
Table of Contents	v
List of Figures	vii
List of Tables.....	x
Acknowledgements	xi
Vita.....	xiii
Abstract of the Dissertation	xiv
Chapter 1 Introduction.....	1
Part I Learning with feature feedback	8
Chapter 2 Introduction.....	9
Chapter 3 Theory on feature feedback	15
Chapter 4 Practical models of feature feedback	26
Part II Interactive topic modeling	34
Chapter 5 Introduction.....	35
Chapter 6 Constraint-based interactive topic modeling.....	44
Chapter 7 Interactive topic modeling with anchor anchors.....	53
Part III Interactive machine teaching	70
Chapter 8 Introduction.....	71
Chapter 9 Interactive machine teaching.....	77
Appendix A Supplementary material for Part I	86
Appendix B Supplementary material for Part II	105
Appendix C Supplementary material for Part III	114

Bibliography 127

LIST OF FIGURES

Figure 1.1.	Annotation with feature feedback	3
Figure 1.2.	Examination of a topic model by a user.	4
Figure 2.1.	Annotation with and without feature feedback.	10
Figure 2.2.	Vague feature feedback: selecting a word in x indirectly and noisily triggers a subset of the latent features z	12
Figure 4.1.	(a) to (c): Learning Curves at Different Values of α . (d): Number of Support Vectors.	31
Figure 5.1.	Topic models as interpreted by users: most probable words under each topic and how users interpret them.	36
Figure 5.2.	Anchor words are words that are very specific to a certain topic, thus are expected to have non-zero probability only under that topic. Anchor word-topic probabilities are the red-colored boxes to the right.	38
Figure 5.3.	Topic assignments are illustrated with different colors. Aggregating all topic assignments induces the topic proportions for this document. Picture taken by [11].	39
Figure 5.4.	Interactive topic modeling with anchor words. Anchor words that may be in the same topic are merged together to form the <i>idealized</i> topic that a user may desire, while other anchor words may be ignored.	40
Figure 5.5.	A view of our interactive anchor based system: a user is creating an “election hacking” topic by providing the words “computer”, “fbi”, “emails”, “messages” as anchors to the system.	42
Figure 6.1.	An interactive system that implements the constrained-based protocol. Here, the user has grouped words related to “presidential elections” in a bucket and words related to “terrorism” in another bucket	48
Figure 6.2.	Experiments on the 20ng data set. The first six panels in each figure show precision and recall curves in the various rounds. The last panel shows area under the precision and recall curve.	50
Figure 6.3.	Experiments on the Webkb data set. The first six panels in each figure show precision and recall curves in the various rounds. The last panel shows area under the precision and recall curve.	51
Figure 7.1.	The generic anchor words algorithm.	54

Figure 7.2.	Illustration of anchor facet shortcoming. Here the user combines anchor words ‘computer’ and ‘games’ which results in a point ‘computer-games’ somewhere in the middle of the simplex spanned by s_1 , s_2 and s_3 .	56
Figure 7.3.	Full interactive recovery algorithm	60
Figure 7.4.	Recovery of underlying topics using different forms of interaction. Subtopic average is the topic model created by averaging together all of the underlying subtopics.	62
Figure 7.5.	<i>Left</i> : Log-likelihood per token, coherence, % unique words, average entropy of topics. <i>Right</i> : Per-user performance on word intrusion task. Users were tested on all user-created topics they created.	66
Figure 8.1.	Left: A non-interactive teacher that provides examples in one shot. Right: An interactive teacher.	74
Figure 9.1.	The teacher’s algorithm. Here $m = \mathcal{X} $ and $N = \mathcal{H} $. For $S \subset \mathcal{X}$, we define $w(S) = \sum_{x \in S} w(x)$.	82
Figure 9.2.	Top: ‘Moon’ data with RBF kernel SVM; Middle: ‘Mixtures’ data with quadratic kernel; Bottom: MNIST (quadratic SVM) and Fashion MNIST (CNN).	85
Figure A.1.	Summary of the datasets and the number of topics used in the experiment	93
Figure A.2.	Top : Topic representation of a document with the class rec.motorcycles before and after feature feedback on bike and biker . Bottom: Descriptive words of the topics that are present in the document.	95
Figure A.3.	20ng	96
Figure A.4.	webkb	97
Figure A.5.	R8	98
Figure A.6.	R52	99
Figure A.7.	Cade	100
Figure A.8.	Ohsumed	101
Figure A.9.	Amount of Feature Feedback	102
Figure A.10.	Amount of Feature Feedback	103

Figure A.11.	Interface used in Human Experiment	104
Figure B.1.	The initial list of candidate anchor words that was presented to users. Users initialized topics that they wanted to create by dragging and dropping a candidate anchor in the dotted box labeled as ‘Merge words’.	107
Figure B.2.	(a) Merge anchor words view. (b) Complete groups view. (c) Merge and trash groups view. (d) Suggest anchor words view.	108
Figure C.1.	Moon-shaped dataset (separable), Linear kernel	117
Figure C.2.	Moon-shaped dataset (separable), Quadratic kernel	117
Figure C.3.	Moon-shaped dataset (separable), RBF kernel	118
Figure C.4.	Moon-shaped dataset (non-separable), Linear kernel	118
Figure C.5.	Moon-shaped dataset (non-separable), Quadratic kernel	119
Figure C.6.	Moon-shaped dataset (non-separable), RBF kernel	119
Figure C.7.	Circular dataset (separable), Linear kernel	120
Figure C.8.	Circular dataset (separable), Quadratic kernel	120
Figure C.9.	Circular dataset (separable), RBF kernel	121
Figure C.10.	Circular dataset (non-separable), Linear kernel	121
Figure C.11.	Circular dataset (non-separable), Quadratic kernel	122
Figure C.12.	Circular dataset (non-separable), RBF kernel	122
Figure C.13.	Mixtures of Gaussians dataset (separable), Linear kernel	123
Figure C.14.	Mixtures of Gaussians dataset (separable), Quadratic kernel	123
Figure C.15.	Mixtures of Gaussians dataset (separable), RBF kernel	124
Figure C.16.	Mixtures of Gaussians dataset (non-separable), Linear kernel	124
Figure C.17.	Mixtures of Gaussians dataset (non-separable), Quadratic kernel	125
Figure C.18.	(a) MNIST data set, quadratic kernel SVM (b) Fashion MNIST data set, convolutional neural network	126

LIST OF TABLES

Table 4.1.	Results of Human Experiment	32
Table 7.1.	Simulated anchors for each news group category	63
Table 7.2.	K-NN accuracy under various algorithms.	64
Table 7.3.	Examples of user anchor groupings.	66
Table 7.4.	Most probable words for the user created topics shown in Table 7.3.	67
Table B.1.	Topics created by user 1	109
Table B.2.	Topics created by user 2	110
Table B.3.	Topics created by user 3	111
Table B.4.	Topics created by user 4	112
Table B.5.	Topics created by user 5	113
Table C.1.	Number of SVs, TPs, and points that are both SVs and TPs on MNIST.	126

ACKNOWLEDGEMENTS

I am indebted to my advisor Sanjoy Dasgupta for guiding me with care through my years in grad school. I am greatly inspired by Sanjoy and I will carry the lessons I learned studying with him for the rest of my life.

I would also like to thank my friend and collaborator Christopher Tosh. His friendship and support has been invaluable in this journey.

Big thanks go to my friends in grad school Sharad Vikram, Zack Lipton, Akshay Balsubramani and Maximilian Metti for their friendship and support. It's been great fun working closely with them all.

I am grateful to my parents, grand mother, and brother. Their positivity is what kept me going through hard times.

Last but not least, I would like to thank my wife Alyssa for her patience and love throughout all these years. I certainly could not have done any of this without her.

Chapter 3 contains material as it appears in “Learning with feature feedback: from theory to practice.” S. Poulis and S. Dasgupta. International Conference on Artificial Intelligence and Statistics 2017. The dissertation author was the primary investigator.

Chapter 4 contains material as it appears in “Learning with feature feedback: from theory to practice.” S. Poulis and S. Dasgupta. International Conference on Artificial Intelligence and Statistics 2017. The dissertation author was the primary investigator.

Chapter 6 contains material that is currently being prepared for submission for publication of the material. S. Poulis, S. Dasgupta, C. Tosh. The dissertation author was the primary investigator.

Chapter 7 contains material as it appears in “Interactive topic modeling with anchor words.” International Conference on Machine Learning 2019. S. Poulis, S. Dasgupta, C. Tosh. The dissertation author was the primary investigator.

Chapter 9 contains material as it appears in “Teaching a black-box learner.” Inter-

national Conference on Machine Learning 2019. S. Dasgupta, D. Hsu, S. Poulis, X. Zhu.
The dissertation author was the primary investigator.

VITA

- 2005 Bachelor of Science, University of Piraeus
- 2008 Master of Science, University of Northern Colorado
- 2012 Master of Science, University of California, San Diego
- 2019 Doctor of Philosophy, University of California, San Diego

ABSTRACT OF THE DISSERTATION

Algorithms for Interactive Machine Learning

by

Stefanos Poulis

Doctor of Philosophy in Computer Science

University of California San Diego, 2019

Professor Sanjoy Dasgupta, Chair

In interactive machine learning, the learning machine is engaged in some fashion with an information source (e.g. a human or another machine). In this thesis, we study frameworks for interactive machine learning.

In the first part, we consider interaction in supervised learning. The typical model of interaction in supervised learning has been restricted to labels alone. We study a framework in which the learning machine can receive feedback that goes beyond labels of data points, to features that may be indicative of a particular label. We call this framework learning with feature feedback and study it formally in several settings.

In the second part, we study interaction in unsupervised learning, in particular,

topic modeling. Topic models are popular tools for analyzing large text corpora. However, the topics discovered by a topic model are often not meaningful to practitioners. We study two different interactive protocols for topic modeling that allow users to address deficiencies and build models that yield meaningful topics.

In the third part, we study interactive machine teaching. Different from traditional machine teaching, in which teachers do not interact with the learners, we study a framework in which interactive teachers can efficiently teach any concept to any learner.

Chapter 1

Introduction

The standard process of learning from data is typically done through a two-step process: first, a dataset of examples is collected; second, a learning machine is instructed to process the collected dataset and output a low-error hypothesis, as measured by closeness to some target. In supervised learning for instance, the target may be a linear separator or a decision tree and the machine is instructed to find the target by processing labeled examples. Likewise, in unsupervised learning the target may be some structure, such as a particular clustering of the dataset. This process of learning from data is well-understood by now: for supervised learning, a plethora of sample complexity bounds tell us how many labeled examples would suffice to learn various types of concept classes; for unsupervised learning, (eg. clustering, topic modeling) there are several algorithms with guarantees that tell us that the target structure will be provably recovered.

Despite the substantial progress several statistical and algorithmic challenges remain. For example, the number of examples that need to be labeled in order to learn a low-error classifier might be prohibitively large. Similarly, say a domain expert collects a dataset wherein certain patterns are expected. The expert might want to do some exploratory analysis and might decide to run a clustering algorithm on the dataset. How can the algorithm magically know what the patterns that the expert expects are?

In addition to such challenges and limitations, the standard process of learning

from data may not at all reflect how machine learning systems are deployed in the real world today. In contemporary applications of machine learning, e.g. virtual assistants, self-driving cars etc., learning machines are constantly interacting with some source of information. To deal with such situations, a rather different pipeline for learning is needed.

1.1 Interactive machine learning

In recent years, there has been substantial interest in *interactive machine learning*, wherein the learning machine is engaged adaptively with a source of information (e.g. a human or another machine). The hope in interactive machine learning is that interaction will make learning faster or even better. In this thesis, we will study several frameworks for interactive machine learning. We describe these frameworks below.

1.1.1 Learning with feature feedback

In supervised learning, perhaps the most well-studied area of interactive machine learning is *active learning* of classifiers, in which the learning machine requests only the labels of informative examples. It has been shown that active learning algorithms can learn a low-error classifier with substantially fewer labels than those needed by standard supervised learning algorithms.

In several settings however, the interaction between the learning machine and the source of information is much richer than that of active learning. When labeling a dataset for instance, a human can provide labels along with explanations for them. In a document labeling scenario say, the human labeler can highlight a few words that are indicative of the label of the document. This type of interaction that is complimentary to active learning can be called feature feedback.

In the first part of this thesis we study several models of feature feedback and give learning algorithms for each of them. We will see that, in certain cases, learning with feature feedback requires substantially fewer examples than standard supervised learning.

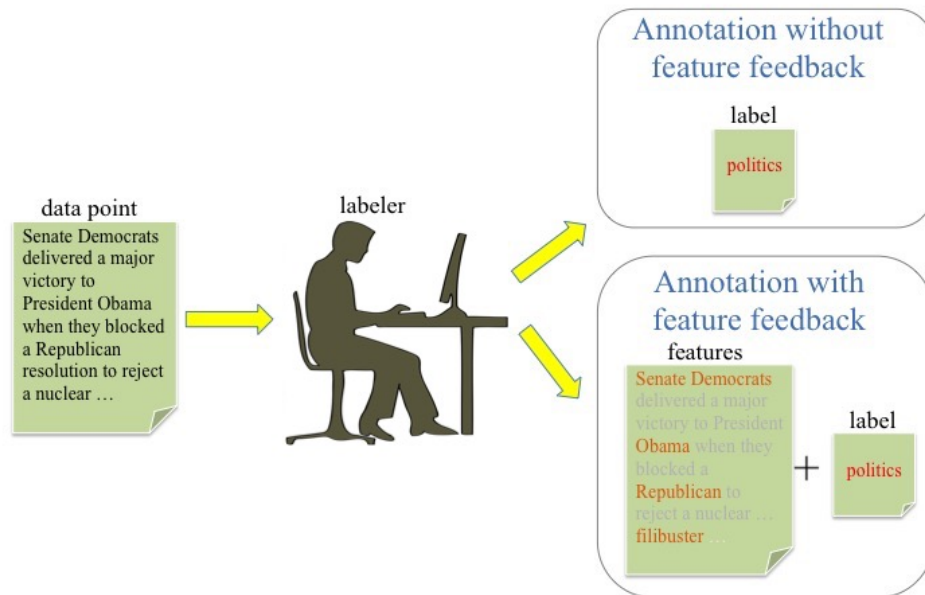


Figure 1.1. Annotation with feature feedback

1.1.2 Interactive topic modeling

Topic modeling is a popular method for learning thematic structure from large collections of documents, without any human supervision. The model is simple: documents are modeled as mixtures of topics, which are in turn modeled as distributions over a vocabulary of words.

The natural interpretation of topics, that they somehow represent the main themes of a corpus, has motivated their use by practitioners. The most common way to summarize a topic model is with a list of their most probable words, and topic models are then evaluated according to how well these lists align with a user's intuition, domain knowledge or understanding of the corpus. In this sense, a user expects to interpret and also evaluate a topic model via a small collection of words. However, traditional topic models may include poor quality topics or can be misaligned with the understanding of the corpus.

For instance, while examining the most probable words under learned topics a user may complain that two topics seem to be the same; or that they seem conflated; or that they seem random.

Interactive topic modeling aims to solve these problems by allowing a user to directly interact with the learned model and iteratively refine it.

In the second part of this thesis we study different frameworks for interactive topic modeling. We will see that in some cases, users can efficiently build customized and interpretable topic models, using our proposed frameworks.

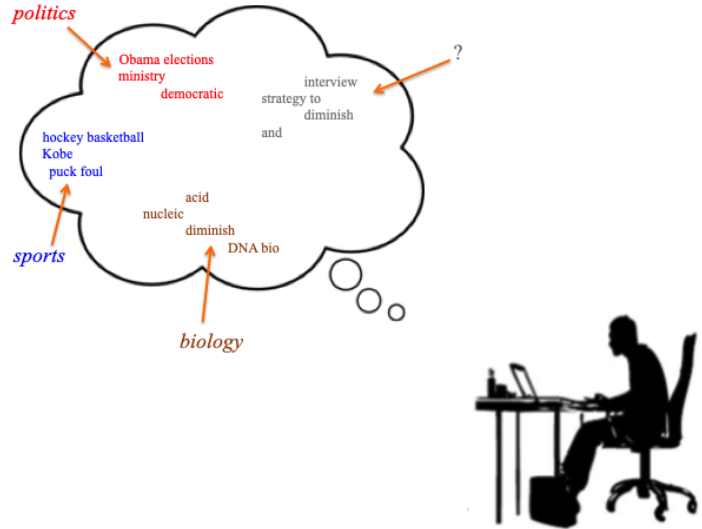


Figure 1.2. Examination of a topic model by a user.

1.1.3 Interactive machine teaching

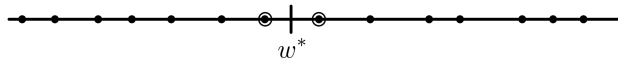
In machine teaching, the model of learning postulates that a “student” receive data from a “teacher”. In this setting, we have a student, who might be a machine learning algorithm, and a teacher that has some target concept h^* it needs to communicate to the student through training examples. The goal of the teacher is to help the learner (here, we use student and learner interchangeably) find the target hypothesis by providing as few

teaching examples as possible.

It can be shown that when learning from a teacher, the number of examples required may be significantly smaller, when compared to simply learning from random examples, i.e. *passive learning*. As an example, consider thresholds on the line. Let \mathcal{X} denote the instance space and $\mathcal{H} = \{h_w : w \in \mathbb{R}\}$ denote the hypothesis class, with

$$h_w(x) = \begin{cases} 1 & \text{if } x \geq w \\ 0 & \text{otherwise.} \end{cases}$$

In passive learning, $O(\frac{1}{\epsilon})$ training items are generally required to achieve an error within ϵ from the target threshold w^* . When the desired error is say, 0.001 the number of examples required in passive learning are in the order of 1000. But in the case of learning from a teacher, who in addition knows the target threshold w^* , only two points in \mathcal{X} are required, the ones nearest w^* , on either side of it:



This example illustrates the benefits of teaching over passive learning but also illustrates a significant issue with this particular notion of teaching: it requires the teacher to know \mathcal{H} , the learner’s hypothesis class. This can be unrealistic in many scenarios.

To put this into context, consider a geologist who may want to teach students to categorize rocks into igneous, sedimentary, metamorphic etc. and teaches by picking informative rock samples to show the students. There are two important points to make here. First, the geologist may know the target hypothesis but cannot “transmit” it into the students’ minds. Second, the geologist and the students may be using different models to categorize rocks, maybe even different representations. Thus, one could say that the students’ model is a *black-box* to the teacher.

In the third part of this thesis we study interactive machine teaching. We will see that when a teacher is allowed to interact with a learner who is a black-box, substantially

fewer teaching examples are needed, when compared to non-interactive teaching.

1.2 Summary of results

In Part I, we study learning with feature feedback. In Chapter 3 we develop some theory on learning with feature feedback. We study several models of feature feedback that deal with various levels of ambiguity and demonstrate their benefits in learning a concept. Then, in Chapter 4 we turn our attention to applications and develop practical algorithms that make use of feature feedback. Finally, we perform experiments illustrating the benefits of feature feedback, both in simulations on benchmark datasets, as well as in a study with real users.

In Part II, we study two protocols for interactive topic modeling. The first protocol, which is presented in Chapter 6, formalizes user interaction in the form of *constraints*. This protocol is studied specifically for Latent Dirichlet Allocation and yields an interactive algorithm that can be implemented efficiently. We show the benefits of this protocol in a series of simulation experiments. Then, in Chapter 7 we present our second interactive protocol for topic modeling, which makes use of *anchor words*: words that appear under only one topic. This protocol is efficient in terms of user interaction and allows users to build topic models that are interpretable and help them understand the main themes of the corpus. We illustrate the benefits of this protocol in simulations, as well as in experiments with real-users.

In Part III, we study interactive machine teaching. We are interested in whether an optimal teaching set exists when the teacher does not know the learner's hypothesis, that is, when the learner is a black-box to the teacher. In Chapter 9, we first illustrate through an example that a teacher who does not know the learner's hypothesis must, in general, provide labels on all the available data points. Then, we present an interactive protocol for black-box teaching. In this protocol, the teacher provides one teaching example at a

time, and in the interim is allowed to probe the predictions of the learner's current model, rather like giving the learner a quiz. We show that such a teacher can efficiently pick a teaching set that provably contains logarithmically as many examples when compared to a non-interactive teacher. We also demonstrate the efficacy of our interactive teaching protocol in a series of simulation experiments.

Part I

Learning with feature feedback

Chapter 2

Introduction

Annotating a data set is often a costly affair because a human is needed to scrutinize each data point and determine its label. One approach to reducing this effort and expense is *active learning*: the learner has access to a pool of unlabeled data points and adaptively decides which ones should be labeled. There are now several active learning algorithms that provably require only logarithmically as many labels as random querying, thus reducing the amount of labeling effort significantly [16, 18, 55] .

2.1 Feedback beyond the label

While scrutinizing a data point in order to provide its label, the human labeler can also provide some additional, richer feedback such as an explanation. This is complimentary to active learning and comes at essentially no extra cost. Here, we consider a strategy where this feedback is in the form of features: can the human, while examining the data point, provide not just the label but also the identity of one or more relevant features?

To put this into context, consider a document classification problem in which a labeler assigns each document x to a category y (“sports”, “politics”, and so on). While making this determination, the labeler might also be able to highlight a few words that are highly indicative of the label (e.g. “Congress”, “Obama”, “filibuster”). Figure 2.1 illustrates annotation with and without feature feedback. Some early work in

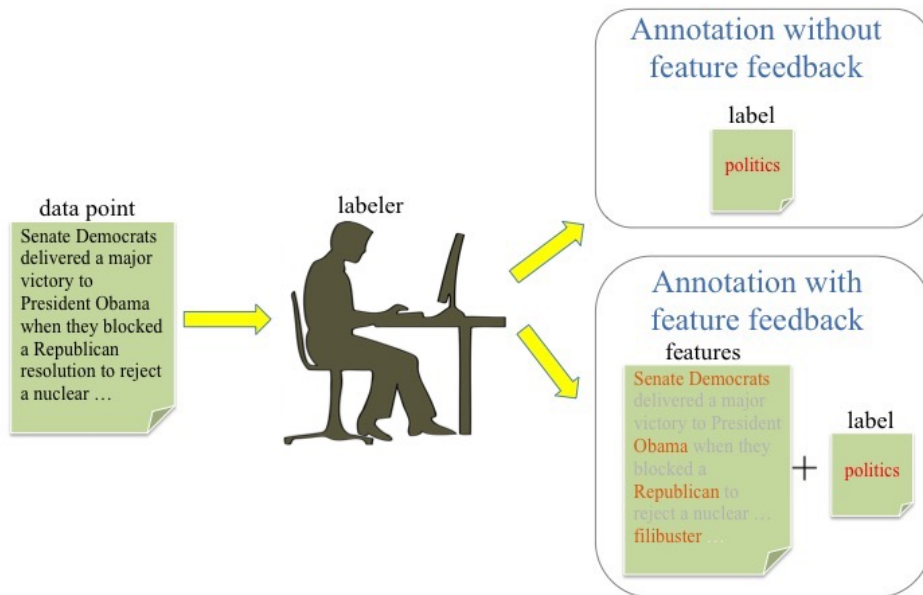


Figure 2.1. Annotation with and without feature feedback.

information retrieval that advocates this kind of auxiliary feedback is that of [17]. Since then, there have been several experimental studies of different methods for exploiting this feedback [45, 19, 22, 44, 50].

Alternatively, consider a computer vision system that is learning to recognize different animals. Whenever it makes a mistake – classifies a “zebra” as a “horse”, say – a human labeler corrects it. While doing this, the labeler can also highlight a part of the image (the stripes, for instance) that distinguishes the two animals. This feedback incurs little additional cost but is potentially very informative for classifier learning. Recent work on recognizing different species of birds, for instance, has used this effectively [12].

Feature feedback may at first seem intuitive but it is not trivial to model as it may vary according to the specific requirements of the learning problem. In the document example, the feedback yields *predictive features*: the presence of words like “Congress”, “Obama”, “filibuster” are predictors of the label “politics”. In contrast, in the vision

example the feedback (i.e. highlighting the stripes) yields a *discriminative feature*, whose presence distinguishes the class of zebras from the class of horses. Which feedback is appropriate for different kinds of learners?

Another difficulty with modeling feature feedback is that it can often be erroneous. Say the label of a document is “finance” and the labeler identifies the word “bank” as predictive. But “bank” has different meanings, some of which have nothing to do with “finance”. Thus, it is interesting to ask what assumptions can be made on the labeler. Is the labeler able to identify all the relevant features or just some of them?

As feature feedback has only been seen in specific applications and has not been studied in any formality such questions are yet to be answered. In this part of the thesis, we formalize feature feedback and study it rigorously. Next, we discuss how feature feedback can be modeled, in various scenarios.

2.1.1 Modeling feature feedback

Let’s return to the example of a document about “politics”, in which the labeler highlights a few specific words. How can a classifier use this? One idea is to somehow boost the importance of the provided words, say in the high-dimensional feature space of bags-of-words. But what happens when the labeler highlights a very rare word, like “filibuster”? This word is, indeed, predictive of the label, but it is also so specific that it might not apply to very many documents. Should then “filibuster” be treated as a proxy for a whole collection of words that co-occur with it, or possibly a proxy for an entire *topic*? This seems reasonable, but what is the right level of granularity for the topic, or the cluster of co-occurring words?

Similarly, in the computer vision example, suppose a labeler decides that a bird is a particular type of robin and provides additional feedback by clicking on its breast (whose color, for instance, might be a deciding factor). The learner may have some higher-level representation of the image, for instance a hierarchical parts decomposition, in which

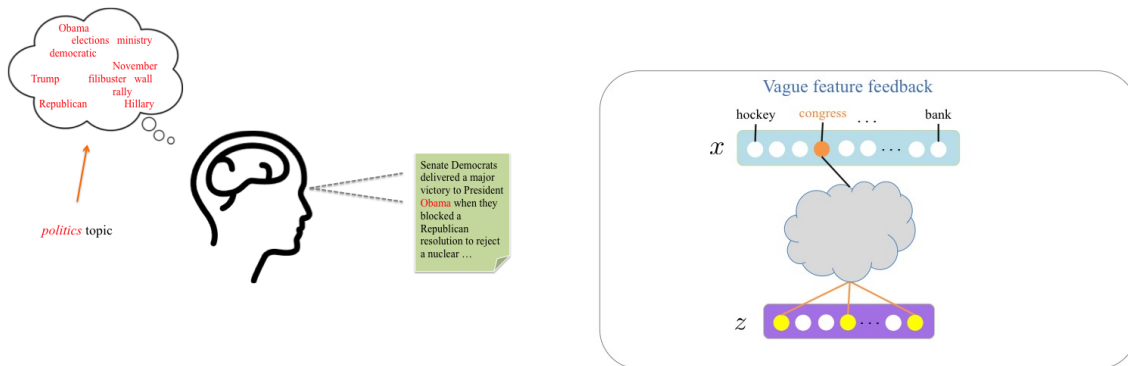


Figure 2.2. Vague feature feedback: selecting a word in x indirectly and noisily triggers a subset of the latent features z .

case it will in general be unclear which of these features the user is referring to—several features, at different scales, might be candidates.

In both the text classification and vision examples above, we see that labeler’s feedback can be quite ambiguous. In order to model this ambiguity, we will think of feature feedback as follows. We assume that there is a raw input x (document, image) and possibly an intermediate representation z (clusters of words, hierarchical parts decomposition) that the labeler cannot access directly. After deciding on the label y , the labeler may indicate one or more coordinates in x . In the absence of the intermediate features z , this feedback is *explicit*: the features that the labeler indicates at the x level will directly be used by the classifier. But when an intermediate representation z is available, the labeler’s feedback in x can also indirectly and noisily reference a subset of features in z , of which some might be relevant to y and some not. We call this type of feature feedback *vague*. Figure 2.2 illustrates vague feature feedback for the example of document classification.

2.1.2 Overview

In Part I we focus specifically on predictive feature feedback and present several models that can accommodate both explicit and vague feature feedback. The rest of Part I is organized as follows. First, in Section 2.1.3 we review previous work in feature feedback.

In Chapter 3 we study feature feedback for various abstract settings. We start with illustrating feedback feedback for the case of learning disjunctions in Section 3.1. Then, we move to models that are substantially more flexible and practical.

In Section 3.2.2 we study a model that is a probabilistic generalization of disjunctions. We define this model specifically in the document-topic setting, but it applies more generally to the x - z - y situation described above: the label y of each document x is assumed to be probabilistically generated from the unnamed intermediate-level features z . We call this the *probabilistic disjunction model* (PDM). We show that if we only had documents and labels, we could try to find a maximum-likelihood fit for the generative model, but we show that this is an NP-hard problem. On the other hand, feature feedback makes learning tractable. We give an efficient algorithm that exploits this feedback to learn a PDM.

In Section 3.3 we study learning linear separators with feature feedback. We suggest a straightforward approach to incorporating information that a particular feature is relevant: reducing the degree of regularization on that feature. This is algorithmically simple and we show that it leads to better generalization bounds.

In Chapter 4 we turn our attention to applications and develop two practical algorithms that make use of feature feedback.

In Section 4.1 we develop our first practical algorithm, which is a support vector machine. This algorithm is very simple and is derived directly from our regularization approach described above. We find that the regularization approach to feature feedback, despite its simplicity, has the drawback of not directly modeling vagueness in the labeler’s intent. To address this, the second algorithm that we develop is a bootstrapped PDM algorithm, in which a PDM is first fit to data, using a small amount of feature feedback, and is then used to label whichever documents it is confident about. This augmented training set is then used to train any other model of interest. The bootstrap PDM algorithm is presented in Section 4.2.

In Section 4.3 we present a series of simulation experiments that illustrate our

methods, along with a real user study.

In Section 4.4 we discuss directions for future work and conclude.

2.1.3 Related work

Incorporating domain knowledge into learning is not a new idea. Several works have considered using this knowledge to construct a preliminary classifier or to set Bayesian hyperparameters [48, 59, 19].

For predictive feature feedback more specifically, the feedback model closest in spirit to our approach is probably that of [21], whose *generalized expectation criteria* framework incorporates user-supplied feature-label relationships into the objective function for learning. Another line of work develops the idea of *annotator rationales* [62, 61, 20], in which the labeler highlights regions of the document that serve as explanations of the label; these are then used to generate *contrast examples* (same document, but with these regions removed) and the learning procedure asks for each document to be distinguished from its contrasting version. This framework involves denser annotation than we have in mind. A related form of “contrast example” is considered by [54], who incorporate this into an SVM framework and provide generalization bounds—though these are weaker and less general than our bounds, which have less requirements on the feedback and apply to any linear model. Later work by [53] developed the *constrained weight-space* SVM framework by allowing annotators to provide ranked features. One further research thread includes work developed in [40, 39, 46, 52], where active learning is used to incorporate feature feedback into learning. The framework there is to identify the *most informative features* to be shown to the human, when asked to label an example.

Discriminative feature feedback has only been studied in [49], where an elegant algorithm that solicits feedback that distinguishes true labels from mistaken predictions is presented. It is shown that the algorithm can provably learn whenever the target concept is a decision tree, or can be expressed as a particular type of multi-class DNF formula.

Chapter 3

Theory on feature feedback

In this chapter, we study feature feedback for different concepts. We first study the case where the target concept is a disjunction of boolean variables. Our analysis is under the mistake bound model of Littlestone [33], which we describe next.

3.0.1 The mistake bound model

Let \mathcal{X} be any finite instance space and \mathcal{Y} any label space. Also, let \mathcal{H} be any finite set of concepts on \mathcal{X} , so that each $h \in \mathcal{H}$ is of the form $h : \mathcal{X} \rightarrow \mathcal{Y}$. Let $h^* \in \mathcal{H}$ denote the target concept, that is h^* is the only concept in \mathcal{H} that is consistent with the labeled examples. Learning in the mistake bound model proceeds in rounds:

For $t = 1, 2, \dots$:

1. The learner receives a data point $x_t \in \mathcal{X}$.
2. The learner makes a prediction $h_t(x_t) = \hat{y}_t$.
3. The correct label y_t is revealed.
4. The learner updates its hypothesis to h_{t+1} .

Under the mistake bound model, the goal is to bound the total number of mistakes the learner commits, no matter how long the sequence.

3.1 Learning disjunctions with predictive feature feedback

For instance space $\mathcal{X} = \{0, 1\}^d$ and label space $\mathcal{Y} = \{0, 1\}$, let $\mathcal{H}_{d,k}$ denote the class of k -sparse monotone disjunctions, that is,

$$\mathcal{H}_{d,k} = \{x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_j} : 1 \leq i_1 < \cdots < i_j \leq d, 0 \leq j \leq k\}.$$

3.1.1 Learning without feature feedback

The Winnow algorithm of Littlestone [33] learns the target disjunction h^* with $O(k \log d)$ mistakes. In several domains however, d could be quite large and potentially infinite. Thus, it is of interest to remove the dependence on d . Can we achieve this with feature feedback? In the next section, we will see that we can.

3.1.2 Learning with feature feedback

In the simplest model of feature feedback, each label is accompanied by the index of a relevant feature, if appropriate. This is particularly easy to formalize in the case of learning disjunctions:

At round t :

1. If an instance x_t satisfies the target disjunction, then the learner receives a positive label as well as the index of a feature that is in the disjunction *and* is set in x_t .
2. If x_t does not satisfy the target disjunction, then the learner receives only a negative label.

Formally, for any $R \subset [d]$, write

$$h_R(x) = \bigvee_{i \in R} x_i,$$

and let R^* be the index set corresponding to the target disjunction h^* (that is, $h^* = h_{R^*}$). Then feature feedback on a positive instance x consists of any member of $R^* \cap \text{pos}(x)$.

As discussed in the introduction, in many scenarios the feature feedback is *vague*, that is, the features that the labeler identifies may not all be relevant. In order to model this we consider a weaker form of feature feedback: instead of getting the index of a specific relevant feature, the learner receives a small set of features of which at least one is relevant. That is, the learner is given a set $S \subset \text{pos}(x)$ such that $S \cap R^* \neq \emptyset$. When the size of this set is (at most) a constant c , we call this *c-vague feature feedback*. When all the features in S are relevant or when $c = 1$, the feedback is *explicit*.

Here's a simple online algorithm for learning k -sparse disjunctions with vague feature feedback. The algorithm, makes a prediction before seeing each label, and requires feature feedback only on mistakes.

```

Initialize  $R = \emptyset$ 
Repeat:
  See instance  $x$  and predict  $h_R(x)$ 
  Receive label  $y$ 
  If  $y = 0$  but  $h_R(x) = 1$ : (false positive)
    Set  $R = R \setminus \text{pos}(x)$ 
  If  $y = 1$  but  $h_R(x) = 0$ : (false negative)
    Receive a subset  $S \subset [d]$  and set  $R = R \cup S$ 

```

Lemma 3.1. *Suppose the labeler provides c -vague feature feedback, for some positive integer c . Then this method makes at most ck mistakes.*

Proof. A false negative occurs only when none of the target features in the current instance are in the set R . And when a target feature is added to R , it is never removed. Therefore, there are at most k false negatives; call this number f .

During these f false negatives, a total of at most cf variables are added to R ; at no other point does R grow. During a false positive, at least one variable is eliminated from R . Therefore, the number of false positives is at most $(c - 1)f$. Thus the total number of mistakes is at most $cf \leq ck$. \square

The class of disjunctions is interesting for theory but are not expressive enough for many practical situations. Thus we next develop a probabilistic generalization of disjunctions that is substantially more flexible. For concreteness, we define this model specifically in a document classification setting, but it applies more generally to the $x - z - y$ situation described in the introduction.

3.2 The Probabilistic Disjunction Model (PDM)

Let's define a stochastic model that generates the label $y \in \{1, 2, \dots, k\}$ of any document d . The model makes use of an intermediate-level representation that, for concreteness, we think of as referring to topics.

Suppose we have a set of T "topics" as well as a procedure for representing any document as a convex combination $\theta = (\theta_1, \dots, \theta_T)$ of these topics (so the θ_t are nonnegative and sum to 1). The details of how this is done are irrelevant. We will assume that every topic $t \in \{1, 2, \dots, T\}$ either has an associated label $\ell(t) \in \{1, 2, \dots, k\}$ or has $\ell(t) = ?$. In the former case, the topic is a strong predictor of the corresponding label. In the latter case, the topic is ambiguous, for instance, an overly general topic. We will denote the set of predictive topics as $P = \{t : \ell(t) \neq ?\}$ and we will assume that every document assigns non-zero probability to at least one predictive topic, that is, $\sum_{t \in P} \theta_t > 0$.

The *probabilistic disjunction model* is a generative process for the label of a document:

- Let $\theta = (\theta_1, \dots, \theta_T)$ be the topic representation of the document.
- Pick a predictive topic at random: choose $t \in P$ with probability proportional to θ_t .

- The label of the document is $\ell(t)$.

3.2.1 Learning without feature feedback

Suppose there is no feature feedback; that is, the learner has access only to a collection of (document, label) pairs. A reasonable objective, under the above stochastic model, is to find the assignment $\ell : \{1, 2, \dots, T\} \rightarrow \{1, 2, \dots, k, ?\}$ that maximizes the likelihood of the data. But we can show that merely finding an assignment with non-zero likelihood is NP-hard.

Theorem 3.2. *The following problem is NP-complete: Given a collection of labeled documents, where each document is represented as a distribution over topics, and where $k = 2$ (binary labels), find an assignment $\ell : [T] \rightarrow \{0, 1, ?\}$ with non-zero likelihood.*

(Proof in Appendix A.1.1.) Feature feedback makes this intractability go away, as we will see next.

3.2.2 Learning with feature feedback

The interactive labeling process works as follows:

Repeat until the budget for human interaction runs out.

1. The labeler gets a batch of (say) 10 documents.
2. For each document: he/she assigns it a label and chooses a predictive word (or maybe several words).

The goal of the learner is to identify the correct mapping $\ell : [T] \rightarrow \{1, 2, \dots, k, ?\}$. A scheme for doing this is shown in Algorithm 1. Roughly, when the user tags a document with label y and identifies relevant words w_1, \dots, w_c , the algorithm picks a set of topics $S \subseteq [T]$ triggered by these words and increments a counter n_{ty} for each $t \in S$. This n_{ty} counts how many times the user has suggested that topic t is predictive of label y .

The specific mechanism for choosing the set S based on the feedback, corresponding to the function `select-topics` in the pseudocode, is not relevant for the theoretical

results we establish below. In our experimental work, we use the following strategy: given feedback words w_1, \dots, w_c for document x , obtain topic distributions for each of these words in the context of document x ; call these p_1, \dots, p_c (distributions over T topics). Add topic t to the selected set S if the t th entry of $(p_1 + \dots + p_c)/c$ exceeds a predefined threshold.

Algorithm 1. Probabilistic Disjunction Model (PDM)

Input: Collection of unlabeled documents U

Initialize: $n_{ty} = 0, \forall t, y$

Labeled data set $L = \emptyset$

repeat

Draw next batch $B \subset U$ of documents at random

$U = U \setminus B$

for each document $x \in B$ **do**

Receive label y , relevant words w_1, \dots, w_c

Add (x, y) to L

$S = \text{select-topics}(x, w_1, \dots, w_c)$

for $t \in S$ **do**

$n_{ty} = n_{ty} + 1$

end for

end for

until budget runs out

Assigning a label to each topic. This is summarized in Algorithm 2. The total amount of feedback received for topic t is $n_t = \sum_y n_{ty}$. If this exceeds some fixed amount n_o , and moreover there is a specific label y for which $n_{ty} \geq \lambda n_t$, then we assign $\hat{\ell}(t) = y$. Here λ is a fixed fraction. In all other cases, we set $\hat{\ell}(t) = ?$.

Labeling a new document. This prediction rule is shown in Algorithm 3. Once topics are labeled, the estimated set of predictive topics is $\hat{P} = \{t : \hat{\ell}(t) \neq ?\}$. Let θ be the

Algorithm 2. Topic labeling assignment (TLA)

Input: $n_{ty} \forall t, y, \lambda, n_o$
for each topic t **do**
 $\hat{\ell}(t) = ?$
 $n_t = \sum_y n_{ty}$
 if $n_t \geq n_o$ **then**
 $y = \operatorname{argmax}_{y'} n_{ty'}$
 if $n_{ty} \geq \lambda n_t$ **then**
 $\hat{\ell}(t) = y$
 end if
 end if
end for

Algorithm 3. PDM prediction rule

Input: Topic representation $\theta \in [0, 1]^T$ of document d
Initialize: $\pi = 0^k$
Label topics according to TLA (Algorithm 2)
for each topic t **do**
 if $\hat{\ell}(t) \neq ?$ **then**
 $\pi(\hat{\ell}(t)) \leftarrow \pi(\hat{\ell}(t)) + \theta_t$
 end if
end for
Normalize π to sum to 1

topic distribution for the new document. The conditional probability that this document has label y is estimated as

$$\pi(y) = \frac{\sum_{t: \hat{\ell}(t)=y} \theta_t}{\sum_{t \in \hat{P}} \theta_t}.$$

3.2.3 Theoretical Guarantees

Correctness of topic labeling

In order to show that the topic labeling algorithm recovers the true labels $\ell(t)$ with high probability, we do not need the full strength of the PDM assumption. What we require is that the topics selected by the user are not systematically misleading. On each round, the machine associates a set of user-selected topics S with a label y . Some of these associations may be spurious, for instance, due to polysemy that the user inadvertently

overlooks. But the same spurious associations should not occur repeatedly.

To formalize this, first observe that the two sources of randomness in topic labeling are: (1) the random selection of documents for labeling, and (2) the possibly stochastic mechanism by which the human selects helpful words from a document.

Assumption 3.1. *For any topic t and any label $y \neq \ell(t)$, if we pick a document at random, ask the human for the label and for helpful words, and look at the induced set of selected topics,*

$$\Pr(\text{label} = y \mid \text{topic } t \text{ is selected}) \leq \lambda/2.$$

Meanwhile, for any predictive topic $t \in P$,

$$\Pr(\text{label} = \ell(t) \mid \text{topic } t \text{ is selected}) \geq 2\lambda.$$

Theorem 3.3. *Pick any $0 < \delta < 1$. Suppose Assumption 3.1 holds and that we set $n_o \geq (6/\lambda) \ln(Tk)/\delta$. Then with probability at least $1 - \delta$, for all $t \in [T]$ with $n_t \geq n_o$, we have $\widehat{\ell}(t) = \ell(t)$.*

(Proof in Appendix A.1.2.)

Label complexity

In order to quantify the amount of feedback needed to recover the true labels ℓ , we require that the user doesn't systematically avoid any informative topics, as follows.

Assumption 3.2. *There is an absolute constant c_o for which the following holds. Pick any t, y such that $\ell(t) = y$. Then for any document with topic distribution θ and label y , if we solicit feature feedback and look at the induced set of topics,*

$$\Pr(\text{topic } t \text{ is selected}) \geq c_o \frac{\theta_t}{\sum_{t': \ell(t')=y} \theta_{t'}}.$$

Let $\theta(x) = (\theta_1(x), \dots, \theta_T(x))$ be the topic distribution for any document x . We define the *prevalence* of a predictive topic $t \in P$ as

$$\gamma_t = \mathbb{E}_x \left[\frac{\theta_t(x)}{\sum_{t' \in P} \theta_{t'}(x)} \right],$$

where the expectation is over a uniform-random choice of x from the corpus. Roughly, γ_t tells us how common topic t is relative to other predictive topics, and thereby how easy it is to estimate $\ell(t)$.

Theorem 3.4. *Suppose documents are labeled according to the PDM process. Under Assumption 3.2, for any $t \in P$, the expected number of labels needed for $\ell(t)$ to be set is at most $n_o / (c_o \gamma_t)$.*

(Proof in Appendix A.1.3.) For fixed constants λ and δ , we need $n_o = O(\ln Tk)$. If all predictive topics are equally prevalent then they each have $\gamma_t = 1/|P|$. In this case, the number of rounds of interaction needed is $O(|P| \ln(Tk))$. This shows the benefit of feature feedback when only a small fraction of the topics are predictive (that is, $|P| \ll T$).

3.3 Learning linear thresholds with feature feedback

We now study feature feedback in the setting where the goal is to learn a linear classifier by minimizing a loss function and a regularization penalty. Given a data set $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \mathcal{Y}$, the optimization is:

$$\hat{w} = \operatorname{argmin}_w \frac{1}{n} \sum_{i=1}^n \ell(w \cdot x_i, y_i) + \lambda \|w\|^2,$$

where $\ell(\cdot)$ is a loss function and $\|\cdot\|$ is some norm. For SVMs, for instance, ℓ is the hinge loss and $\|\cdot\|$ is the 2-norm.

We propose a simple scheme for incorporating information about relevant features: reduce the regularization along those specific dimensions. To achieve this, we take the regularization norm $\|\cdot\|$ to be a *Mahalanobis* norm, given by a $p \times p$ positive definite matrix A :

$$\|x\|_A = \sqrt{x^T A x} = \|A^{1/2} x\|_2.$$

In the absence of feature feedback, A is the identity matrix I_p , giving the 2-norm. But if we find that features $R \subset [p]$ are relevant, we downweight the diagonal matrix in those dimensions: we set $A_{jj} = 1/c$ for relevant features j and $A_{jj} = 1$ otherwise, for some $c > 1$. In spirit, this regularization reweighting is analogous to increasing the prior on these features in a Bayesian model, as was done in [50].

We next study the statistical benefit of this estimator.

3.3.1 Improved Generalization Error Bounds

Let's start with a generalization bound for learning linear classifiers chosen from some set \mathcal{F} . Write the empirical loss function as

$$\widehat{\mathbb{L}}(w) = \frac{1}{n} \sum_{i=1}^n \ell(w \cdot x_i, y_i)$$

(regularization is incorporated by restricting \mathcal{F} to vectors of bounded norm). When the training data (x_i, y_i) comes i.i.d. from an (unknown) underlying distribution, the following seminal result shows the relation of $\widehat{\mathbb{L}}(w)$ to the true loss $\mathbb{L}(w) = \mathbb{E}_{x,y} \ell(w \cdot x, y)$:

Theorem 3.5. [9] *Suppose the loss function ℓ is Lipschitz in its first argument and is upper-bounded by a constant M_ℓ . Then for any $\delta > 0$, with probability $\geq 1 - \delta$ over the choice of data,*

$$\forall f \in \mathcal{F} : \quad \mathbb{L}(f) \leq \widehat{\mathbb{L}}(f) + 2R_n(\mathcal{F}) + M_\ell \sqrt{\frac{\log 1/\delta}{2n}},$$

where $R_n(\mathcal{F})$ is the Rademacher complexity of \mathcal{F} .

The key term here is $R_n(\mathcal{F})$. In our setup, let w^* be a sparse target classifier of interest and define a feature as being relevant if it is set in w^* . Using a powerful result of [29], we obtain the following.

Theorem 3.6. *Let $R = \{j \in [p] : w_i^* \neq 0\}$ denote the relevant features of w^* .*

- *We can write any x in terms of its relevant and other components, $x = (x_R, x_o)$.*
- *Let A be the diagonal matrix whose j th entry is $1/c$ if $j \in R$ and 1 otherwise.*

Then, for the family of linear separators $\mathcal{F} = \{w : \|w\|_A \leq \|w^\|_A\}$, we have*

$$R_n(\mathcal{F}) \leq \|w^*\|_2 \cdot \max_{x \in \mathcal{X}} \sqrt{\left(\frac{1}{c} \|x_o\|_2^2 + \|x_R\|_2^2\right)} \sqrt{\frac{2}{n}}.$$

(Proof in Section A.2.1.) In situations where the x_o (the irrelevant portion of the data) has significant norm, this downweighting by a factor of c substantially reduces the generalization error bound.

Chapter 3 contains material as it appears in “Learning with feature feedback: from theory to practice.” S. Poulis and S. Dasgupta. International Conference on Artificial Intelligence and Statistics 2017. The dissertation author was the primary investigator.

Chapter 4

Practical models of feature feedback

In the previous chapter we studied the benefits of feature feedback in abstract settings, for various concepts. In this chapter we turn our attention to applications.

4.1 Learning a support vector machine with feature feedback

Given training data $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \mathcal{Y}$ consider the SVM problem with our Mahalanobis regularizer:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{2} \|w\|_A^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & \xi_i \geq 0, \quad y_i(x_i^T w + b) \geq 1 - \xi_i, \quad \forall i. \end{aligned}$$

A straightforward derivation shows the following.

Lemma 4.1. *Pick any positive definite $p \times p$ matrix A . Then, learning a linear SVM on instances $\{(x_i, y_i)\}_{i=1}^n$ with Mahalanobis regularizer $\|w\|_A$ is equivalent to learning a linear SVM on modified instances $\{(A^{-1/2}x_i, y_i)\}_{i=1}^n$ with $\|w\|_2$ regularization.*

(Proof in Section A.3.) An SVM algorithm with feature feedback (SVM-FF) is given in Algorithm 4. For each supplied feature, the corresponding diagonal entries of A are set to a particular value $c < 1$ and every labeled and unlabeled example is weighted by

$A^{-1/2}$. Then, a standard linear SVM is trained on the weighted labeled instances.

Algorithm 4. SVM with feature feedback (SVM-FF)

Input: $c < 1$, unlabeled data set U

Initialize: $L = \emptyset, A = I_p$

repeat

Draw next batch $B \subset U$ of documents

$U = U \setminus B$

for each document $x \in B$ **do**

Receive label y , words s

Add (x, y) to L

for $j \in s$ **do**

$A_{jj} = c$

end for

Train linear SVM on $\{(A^{-1/2}x, y) : (x, y) \in L\}$

end for

until budget runs out

4.2 Bootstrapping PDM

The feedback in the regularization approach is explicit: the regularization will only be applied to features that the labeler selects. Let’s return to the “filibuster”–“politics” example in the introduction. Even though the word “filibuster” is a good predictor for “politics” it is a fairly uncommon word. Hence, not that many documents will be affected by reducing the regularization on it. On the other hand, vague feature feedback facilitated by the PDM is richer: feedback on “filibuster” propagates to other words in the same topic. To incorporate vague feedback into a linear classifier, we introduce the *bootstrapped* PDM (Algorithm 5). Given a labeled data set L and an unlabeled data set U , the algorithm fits a PDM to L and uses this PDM to predict on U . It then infers the labels of a set $I \subseteq U$

of data points for which it is confident. We say that the PDM is confident on an instance x if its prediction \hat{y} has estimated conditional probability $\pi(\hat{y}) \geq \tau_0$ (recall the notation of Algorithm 3), where τ_0 is a parameter to be set. One can then train any classifier on $L \cup I$. If the classifier of choice is a linear SVM, one can apply the mixed regularization, by multiplying every example by $A^{-1/2}$ and training a linear SVM on this weighted data set of labeled and inferred points.

Algorithm 5. Bootstrap PDM

Input: Unlabeled data set U, τ_0 (optionally, $c < 1$)

Initialize: $L = \emptyset$ (optionally, $A = I_p$)

repeat

Draw next batch $B \subset U$ of documents

$L = L \cup B; \quad U = U \setminus B$

Train PDM (Algorithm 1) on L

(optionally, update A as in Algorithm 4)

for each document $x \in U$ **do**

$I = \emptyset$ (documents with inferred labels)

Predict $\pi(\cdot)$ over labels according to Algorithm 3

Predict $\hat{y} = \operatorname{argmax}_{y' \in \{1, \dots, k\}} \pi(y')$

if $\pi(\hat{y}) \geq \tau_0$ **then**

Add (x, \hat{y}) to I

end if

end for

Train any classifier on $\{(x, y) : (x, y) \in L \cup I\}$

(optionally, train linear SVM as in Algorithm 4)

until budget runs out

4.3 Experiments

We conducted experiments on the following 6 benchmark text categorization data sets. **20 NewsGroups**: Set of approximately 20,000 documents, partitioned evenly across 20 newsgroups, containing postings about politics, sports, technology, religion, science etc. **Reuters-21578**: Another widely used collection for text categorization research. Documents with less than or with more than one label were eliminated, resulting in **R8** (8 classes) and **R52** (52 classes). **webkb**: Data set that contains web pages collected from computer science departments of various universities. **cade**: Web pages from the CADE Web Directory, which points to Brazilian web pages classified by human experts in 12 classes, including services, education, sciences, sports, culture etc. **ohsumed**: Medical abstracts from the MeSH (Medical Subject Headings) data set, belonging to 23 cardiovascular disease categories. For further details on the data sets, see section A.4.1 of the Appendix. The first five data sets were already processed [14]; we processed **ohsumed** in the same manner (stemming, removal of stop words and words shorter than two characters). As we are interested in single label documents, we only kept data points that had only one label. For each document we obtained its tf-idf and topic representations. For the latter we trained a Latent Dirichlet Allocation model using the collapsed Gibbs sampler [26]. The number of topics was 10 times the number of classes in each data set.

Oracle features

To simulate the labeler’s feedback, we first generated a list of *oracle* features for each class as follows. We first trained a logistic regression classifier with ℓ_1 regularization and took all the feature weights that were positive. We then looked at the level of correlation between these features and the class labels. Specifically, for various thresholds α , we considered feature j as feedback for class k if $P(k|j)$, the conditional probability of label k given the presence in the document of word j , was at least α . We then tested our models

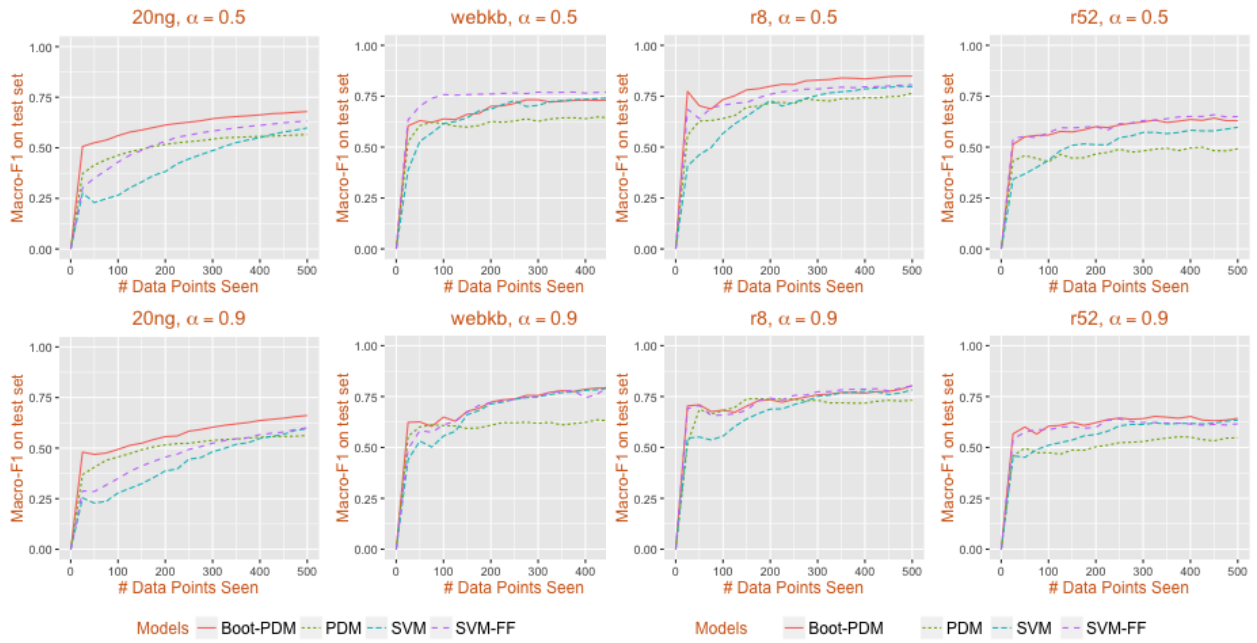
for various values of α . Feature feedback on a document applied if it contained any of the words in the list of its label. An example of feature feedback for the **20ng** dataset using the PDM is shown in figure A.2 in the appendix.

Experimental setup

We compared our models to a linear SVM without feedback. To choose the cost C of all SVM classifiers, we only tuned the SVM without feedback by optimizing the macro- F_1 score on the grid $\{1, 10, 100, 1000\}$. We then set C for the SVM-FF and bootstrap PDM models to that value. On the first few batch iterations we used 2-fold cross validation and continued with 5-fold in later iterations. We set the rest of the parameters for PDM, SVM-FF, and bootstrap PDM as follows: $\lambda = \frac{1}{10}$, $n_o = 2$, $c = \frac{1}{20}$ and $\tau_0 = .75$.

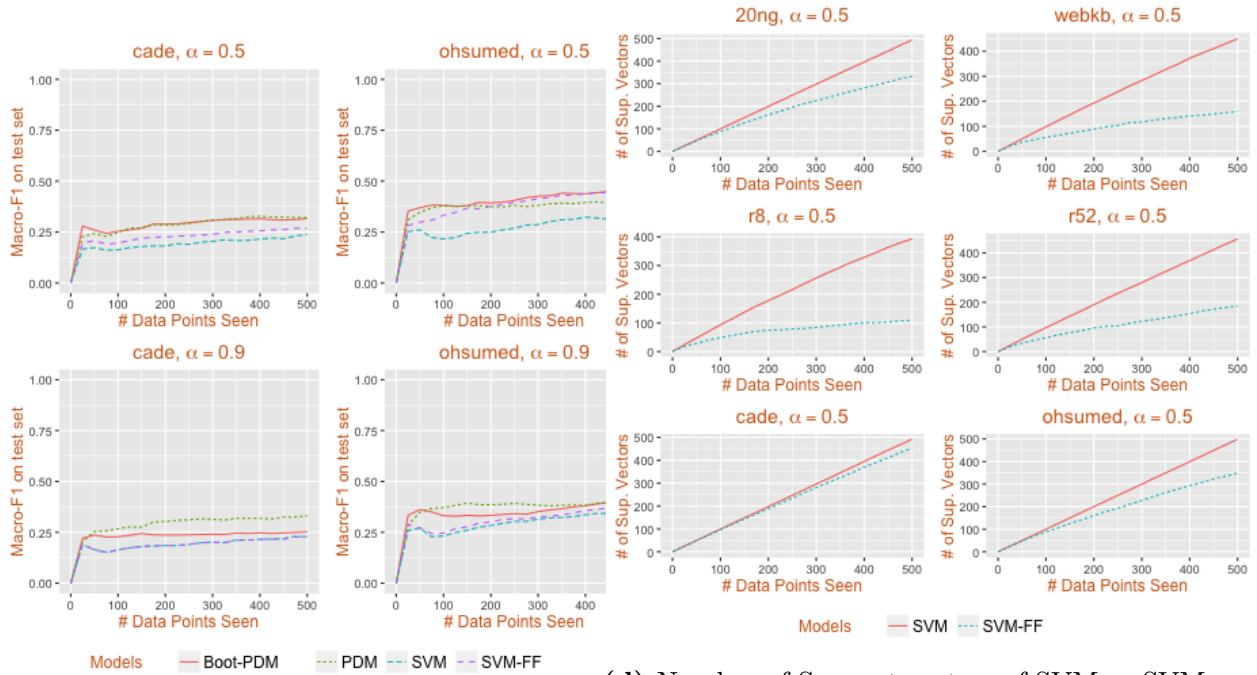
Discussion of simulation results

Figures 4.1 (a-c) show learning curves for the first 500 data points for each training data set, divided into 20 batches. For each batch iteration, we report macro- F_1 score on the test set. (See A.4.2 for a more detailed exposition of the experimental results.) Across the board, we find that feedback on a few predictive words helps significantly. To get a feel of the *amount* of feature feedback see figures A.9- A.10 in A.4.2. Vague feature feedback (PDM, bootstrap PDM) is particularly helpful when the labeled data set is small. Generous feature feedback (i.e. $\alpha \geq .5$) helps fast convergence when data are scarce but has a somewhat adverse effect when plenty of labeled samples are available. However, this improves for $\alpha \geq .9$. Interestingly, in addition to its superior performance, SVM-FF produces a solution that is much sparser than that of the SVM, as seen in figure 4.1d. This makes sense intuitively, as feature feedback helps the learning algorithm to focus on important dimensions.



(a) 20ng-Webkb

(b) R8-R52



(c) cade-ohsumed

(d) Number of Support vectors of SVM vs SVM-FF

Figure 4.1. (a) to (c): Learning Curves at Different Values of α . (d): Number of Support Vectors.

Small *vs* large data regimes

The simulation results illustrate that the benefits of feature feedback diminish asymptotically. We note that since we are learning a linear classifier, in the limit of enough labeled data, we can simply run SVM. Also, the degree of regularization in the SVM-FF can be adjusted so that $c \rightarrow 1$ as the sample grows. Hence, our methods are well suited to the fairly common situation where the amount of labeled data is limited.

Human experiment

To get a sense of the feature feedback that humans tend to provide and to quantify the difference in the benefits of a *selected* feature *vs* a *random* feature, we conducted a small human study involving 5 annotators. We considered a subset of the **20ng** data set that included points with classes *talk.politics.mideast*, *comp.graphics*, *sci.med*, *rec.autos* and *misc.forsale*. The annotators provided the labels of a randomly chosen set of 50 points along with a number of features via an interface. (See A.4.3 for details). For class k , call S_k , N_k the set of features that annotators selected and did not select, respectively. In table 4.1 we show $\bar{p}_{S_k} = \frac{1}{|S_k|} \sum_{j \in S_k} P(k|j)$ and $\bar{p}_{N_k} = \frac{1}{|N_k|} \sum_{j \in N_k} P(k|j)$, where the $P(k|j)$'s are the conditional probabilities described earlier.

Table 4.1. Results of Human Experiment

	\bar{p}_{S_k}	\bar{p}_{N_k}
<i>misc.forsale</i>	0.63	0.76
<i>rec.autos</i>	0.95	0.82
<i>sci.med</i>	0.96	0.78
<i>comp.graphics</i>	0.83	0.66
<i>talk.politics.mideast</i>	0.98	0.74

Note that \bar{p}_{S_k} is smaller than \bar{p}_{N_k} only for the class *misc.forsale* because some annotators confused documents about items for sale with documents with class *comp.graphics*

and *rec.autos*. This is not a surprising effect and we expect to diminish with more labeled data and with a larger pool of annotators. Across the board, we find that humans tend to provide words that are highly predictive of the label.

4.4 Conclusions and future work

In this part of the thesis, we formalized feature feedback, a problem that has been largely studied empirically. We established models of feature feedback that dealt with ambiguity in the intent of the labeler and in several cases were able to quantify its benefits. Our experiments demonstrated that feature feedback can be very useful when labeled data is not abundant or when it is difficult to obtain. There are several directions for future work.

One potential direction is to develop models of feature feedback that operate in the *active learning* setting, where the learner is able to solicit feedback for labels and features actively and adaptively, by making requests only when needed. Thus, it would be interesting to explore whether the logarithmic improvements of active learning can be pushed even further.

Another interesting direction is to extend the work done in [49] for discriminative feature feedback and study further applications. One such application is to study models of discriminative feature feedback in a setting where data points may have multiple labels, such as images with several objects in them.

Finally, it may be of interest to extend our framework of learning linear thresholds with feature feedback. Under our framework, all relevant features were disclosed to the learner in advance. In the future, we envision a setting in which relevant features are gradually disclosed during rounds of interaction.

Chapter 4 contains material as it appears in “Learning with feature feedback: from theory to practice.” S. Poulis and S. Dasgupta. International Conference on Artificial Intelligence and Statistics 2017. The dissertation author was the primary investigator.

Part II

Interactive topic modeling

Chapter 5

Introduction

Topic models [11, 26, 32, 10] are an unsupervised approach to modeling textual data. Given a corpus of documents, topic modeling seeks a small number of probability distributions over the vocabulary, called topics, so that each document is well-summarized as a mixture of topics.

Topic models are most easily described by their generative process, the imaginary random process by which the model assumes the corpus of documents arose. The most common formulation for the document generating process is the following: each word is generated by first, selecting a topic from a document-specific distribution, and then by selecting a specific word from that topic-specific distribution.

There are two main algorithmic methods in fitting a topic model. The first method seeks to find the latent topic assignment for every word-document pair that maximizes some likelihood objective. This is done via *approximate inference* methods, such as variational techniques [11] or Markov Chain Monte Carlo (MCMC) [26]. The second and most recent method treats the topic model fitting problem as one of *statistical recovery*: recover the parameters that generated the corpus with a reasonable amount of samples; several algorithms that assume data are generated by a collection of topics and aim to provably recover these topics have been proposed [4, 5].

Regardless of the specific algorithmic method used to fit a topic model, the natural

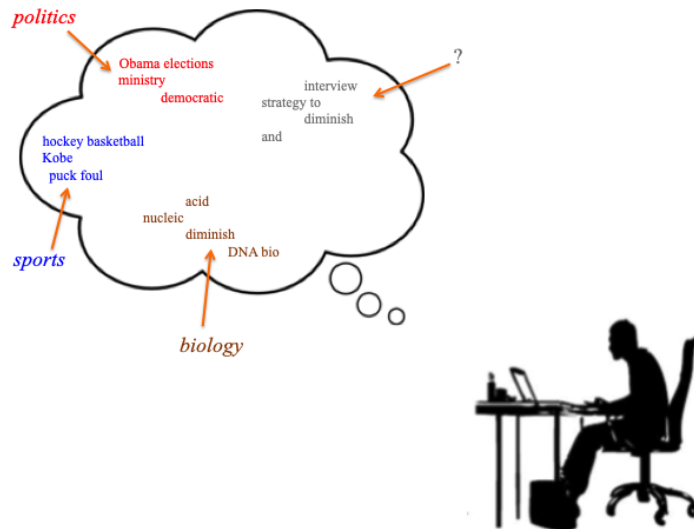


Figure 5.1. Topic models as interpreted by users: most probable words under each topic and how users interpret them.

interpretation of topics, that they represent the main themes of a corpus, has perhaps most motivated their use by practitioners [41, 30, 38]. Indeed, the most common way to summarize topics is with a short list of their most probable words, and topic models are judged according to how well these lists align with a user’s intuition [15]. In this sense, a user expects to interpret a topic model via a small collection of words. Figure 5.1 shows an example of how topics may be interpreted by a user.

Model fit and interpretability form two, sometimes opposing, objectives in topic modeling, and it can be difficult to strike a balance between the two. Consider the challenge of granularity: should there be different topics for “football”, “Olympics” and “basketball” or is a single topic over “sports” sufficient? Obviously more topics will be able to describe the corpus more easily, but a particular user may not care to make the distinction between three sports-related topics. Clearly no unsupervised method can be expected to always make the correct choice here.

To deal with such ambiguities, researchers have considered methods to introduce

interaction into topic modeling algorithms. There are two main approaches in introducing user interaction into a topic model. Naturally, each approach corresponds to one of the two algorithmic methods for fitting a topic model described above.

The first approach, which corresponds to the approximate inference methods for fitting a topic model, has been to encode positive and negative word correlations into prior distributions in the form of constraints. The idea is that by biasing models to group words a user knows should be together and separate words a user knows should remain apart, an algorithm can converge on a topic model that better reflects a user’s preferences. This approach which can be called *constraint-based* has been studied in several works [2, 28, 43].

The second approach, which corresponds to the statistical recovery method to fitting a topic model is to introduce interaction through *anchor words* — words which only occur with significant probability in a single topic [4, 5]. See Figure 5.2 for an illustration of anchor words. Because anchor words occur only in a single topic, users can treat them as proxies for entire topics, allowing large changes in a topic model with only a few interactions. This approach which may be called *anchor word-based* was first proposed by [38].

In this part of the thesis we will develop interactive topic modeling frameworks under both the constraint-based and the anchor word-based approaches. We start by motivating these frameworks below.

5.0.1 Our constraint-based framework

Let’s focus our attention on Figure 5.3 where we show a scientific article. Here we have highlighted different words used in the article with different colors. For example words about “data analysis”, such as “computer” and “prediction” are highlighted in blue; words about “evolutionary biology”, such as “life” and “organism” are highlighted in pink; words about “genetics”, such as “sequenced” and “genes” are highlighted in yellow. If we took the time to highlight every word in the article (other than stop words like “and”,

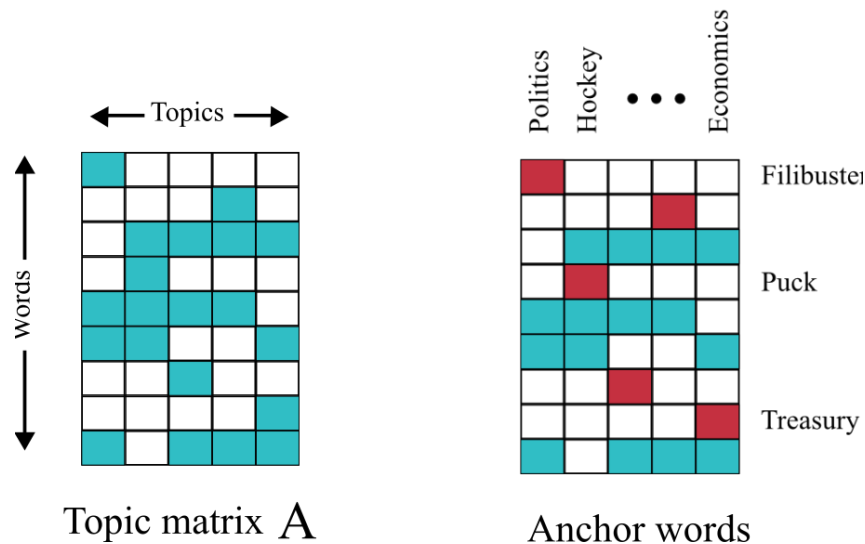


Figure 5.2. Anchor words are words that are very specific to a certain topic, thus are expected to have non-zero probability only under that topic. Anchor word-topic probabilities are the red-colored boxes to the right.

“but” , “if” etc., which contain little topical content) we can obtain the admixtures of topics for this article. Fitting a topic model is very much like this *highlighting* process: the inferencing algorithm will find the most likely topic assignment z_w for each word w .

Now suppose that the inferencing algorithm mistakenly assigns the word “genome” to the blue topic which is about “data analysis” instead of assigning it to the yellow topic, which is about “genetics”. How can a user intervene and correct this? Naturally, the user can step in and highlight “genome” in yellow. The algorithm could then incorporate this feedback as a constraint and in the next round the word “genome” will be assigned to the yellow “genetics” topic. In theory, our user could correct every mistaken assignment, just by highlighting each word with the appropriate color.

Our constraint-based interactive protocol formalizes the above intuition. We think of a topic model as a K -clustering of the words: we assume that for each word w there is a target assignment $z_w^* \in \{1, \dots, K\}$. A topic modeling algorithm will produce an estimate z_w and each time $z_w^* \neq z_w$ a user can intervene and provide feedback in the form of constraints. The algorithm will then incorporate this feedback and will output a new

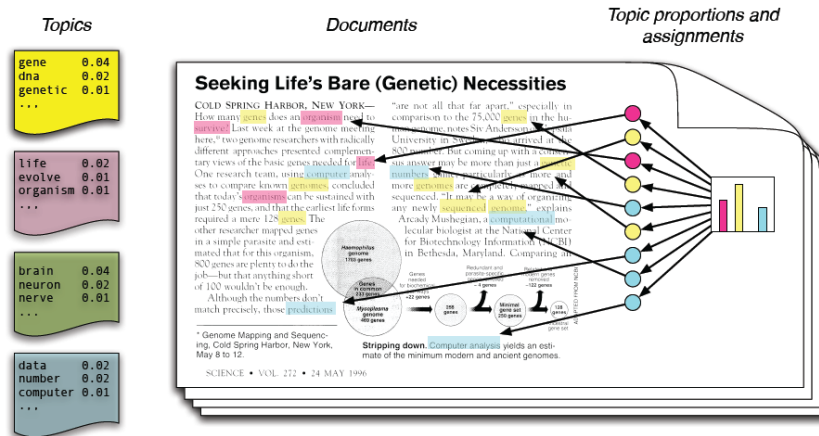


Figure 5.3. Topic assignments are illustrated with different colors. Aggregating all topic assignments induces the topic proportions for this document. Picture taken by [11].

topic assignment that obeys the user provided constraints.

5.0.2 Our anchor word-based framework

An *anchor word* for a topic A_j is a word that has positive probability under topic A_j and 0 probability under any other topic. Given the assumption that every topic has an associated anchor word, there is a natural algorithm to recover the topic matrix [5]. The algorithm proceeds in two steps: first, it selects anchor words for each topic; and second, in the recovery step, it reconstructs topic distributions given those anchor words. The input for the algorithm is the second-order moment matrix of word-word co-occurrences.

Anchor words have the leverage to trigger the large changes in a topic model that a user may be hoping for. Moreover, they may allow a user to address specific deficiencies in a topic model. To see this, recall our earlier example about the “football”, “Olympics” and “basketball” topics. Each of these topics will be associated with an anchor word, say “goal” for the football topic, “medal” for the Olympics topic and “jumpball” for the basketball topic. Now a user might be satisfied with just a single “sports” topic but the corpus itself will not look like it has the *ideal* sports topic that the user wants. What can be done in this case? Naturally, the user can group all three anchor words together, thus

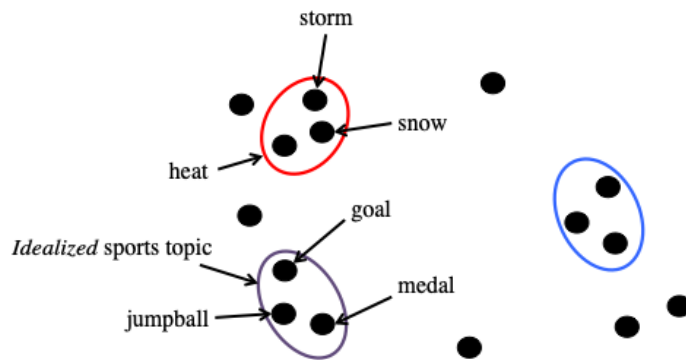


Figure 5.4. Interactive topic modeling with anchor words. Anchor words that may be in the same topic are merged together to form the *idealized* topic that a user may desire, while other anchor words may be ignored.

creating the ideal sports topic.

Our anchor word-based framework for interactive topic modeling is based on this idea. We present an anchor word-based interactive protocol wherein users are shown anchor words and are given the opportunity to create the *idealized* topics that they may desire, by grouping anchors if they should belong to the same topic, while removing others that are uninteresting. Figure 5.4 illustrates this. A topic is then created for each group. We have designed the interaction to be efficient in its use of human feedback by reducing the number of anchor words a user must examine to create a group. Figure 5.5 shows an example of our interactive system.

5.0.3 Overview

The rest of Part II is organized as follows. In Section 5.0.4 we review previous work on interactive approaches to traditionally unsupervised tasks.

In Chapter 6 we present our constraint-based approach to interactive topic modeling. Our interactive protocol, which is developed in Section 6.2, views the topic modeling problem as one of clustering: each word w in the corpus must be assigned to a target cluster $z_w^* \in \{1, \dots, K\}$. Interaction is then designed to solicit user feedback in the form

of “must-link” and “cannot-link” constraints so that the target cluster for each word is respected. We develop this protocol specifically for Latent Dirichlet Allocation [11] and present a version of the Gibbs sampler that incorporates the user-provided constraints and returns a topic model that obeys them. Finally, in Section 6.3 we conduct a series of simulation experiments that show that our interactive approach yields topic models that better aligned with a *target*, when compared to a non-interactive approach.

In Chapter 7 we present our anchor word-based approach to interactive topic modeling. In our approach, we require that the user is allowed to only interact with anchor words and *not* with arbitrary words that may seem interesting or descriptive. In Section 7.2.1, we show that this requirement is crucial by illustrating the potential pitfalls of a previous proposal for anchor word-based topic modeling that allows users to interact with arbitrary words.

In Section 7.2.2, we argue that the assumption that documents are generated by a small number of topics that are succinct, descriptive, and interpretable by a user is often unrealistic. To model the mismatch between the idealized view a user has in mind and the actual data generating process, we introduce a new model of data generation. We call this model the *subtopics model*: for each “ideal” topic there is a number of “subtopics”; documents are then generated as admixtures of these subtopics. We show that under this model, it is difficult to recover the idealized topics and continue by presenting an interactive protocol based on anchor words that is able to recover them.

In Section 7.3 we present a series of experiments. We first demonstrate the efficacy of our anchor-word based approach with *simulated* user interaction and then present a real user study on an interactive system that implements our protocol.

5.0.4 Related work

The observation that unsupervised learning objectives rarely align completely with a user’s intentions is not a new one. Nor is the solution of introducing human feedback



Figure 5.5. A view of our interactive anchor based system: a user is creating an “election hacking” topic by providing the words “computer”, “fbi”, “emails”, “messages” as anchors to the system.

to mitigate this issue. The approaches that have been studied thus far can be generally broken into two categories: constraint-based and higher-order.

In constraint-based interactive learning, a structure is found by optimizing some cost function subject to certain constraints. In flat clustering, for example, these constraints are pairs of data points which either must belong to the same cluster (*must-link*) or cannot belong to the same cluster (*cannot-link*) [58, 6]. For hierarchical clustering, these constraints take the form of triplets of data points $(\{x, y\}, z)$ wherein x and y must be closer to each other in tree-distance than either is to z [57].

In the context of topic modeling, constraint-based interaction has typically focused on probabilistic models where constraints are either down-weighted or eliminated. Whether these constraints are introduced all at once [2, 43] or in interactive rounds [28], the focus of these methods has been on modifying the prior distribution over topics so that they favor certain word correlations. Thus, the user feedback in such methods is translated into *soft constraints*.

In contrast, our approach differs in that we allow for *hard constraints*. We think of a topic model as a clustering of the elements of the corpus. We have designed the interaction so that the user can directly affect the model in a way that respects a target clustering. In principal, our interactive approach allows a user to completely specify a target clustering.

Higher-order feedback seeks to effect large changes in a model by modifying aspects

of the model directly. As such, the types of feedback considered are highly dependent on the task at hand. In clustering, for example, researchers have considered split and merge requests in which a user indicates that a certain cluster ought to be broken up into smaller clusters (a split request) or that several clusters should be grouped together into a single cluster (a merge request). Given certain assumptions on the target clustering, upper bounds can be given on the number of rounds of interaction needed to find the target clustering [8, 7].

Perhaps the most convincing use of higher-order feedback in topic modeling is via anchor words. Because each anchor word has a unique topic associated with it, actions performed on anchor words have the potential to effect large changes in the topic model. [38] proposed a protocol in which a user creates a group of words that they feel are representative of a topic and these words are aggregated into a single pseudo-anchor word. These pseudo-anchor words are then used to create a topic model in the same way that actual anchor words would be used.

The anchor word-based interactive protocol considered in this work is similar to that considered by [38] in its reliance on anchor words. However, our method differs considerably both in the types of words a user can interact with (we only allow a user to interact with geometrically-meaningful anchor words) and in the way we utilize the user-created groups (we sidestep the creation of pseudo-anchor words).

Chapter 6

Constraint-based interactive topic modeling

In this chapter we present our constrained-based approach to interactive topic modeling. As we study this model specifically for Latent Dirichlet Allocation (LDA), we start by specifying the LDA model.

6.1 Latent Dirichlet Allocation

A *corpus* is a collection of documents d_1, \dots, d_m , each of which is represented in the bag-of-words representation as a vector in \mathbb{Z}_+^V , where V is the size of the vocabulary. In LDA the stochastic model that generates the corpus is the following:

For each document d_j :

1. Draw a document-topic Dirichlet distribution with parameter α , $\theta_j \sim \text{Dir}(\alpha)$.
2. Draw a topic-word Dirichlet distribution with parameter β , $\phi_k \sim \text{Dir}(\beta)$.
3. For each word w in position i of document d_j :
 - (a) Draw a topic $z_{ij} \sim \text{Multinomial}(\theta_j)$
 - (b) Draw a word $w_{ij} \sim \text{Multinomial}(\phi_{z_{ij}})$

We can now represent the corpus as N pairs (w_i, d_j) and can think of a topic model as a vector \mathbf{z} of N random variables taking values in $\{1, \dots, K\}$. Each value of \mathbf{z}

corresponds to the topic of (w_i, d_j) . We can also pack all the elements of the corpus into a vector \mathbf{w} . We are then interested in estimating the posterior distribution $P(\mathbf{z}|\mathbf{w})$. It can be shown that

$$Pr(\mathbf{z}|\mathbf{w}) \propto \prod_{k=1}^K \frac{\left(\prod_w \Gamma(n_k^{(w)}(z) + \beta)\right) \left(\prod_d \Gamma(n_k^{(d)}(z) + \alpha)\right)}{\Gamma(n_k(z) + W\beta)}$$

where $\Gamma(\cdot)$ is the gamma function and

$$\begin{aligned} n_k(z) &= |\{i, j : z_{ij} = k\}| \\ n_k^{(w)}(z) &= |\{i, j : z_{ij} = k, w_i = w\}| \\ n_k^{(d)}(z) &= |\{i, j : z_{ij} = k, d_j = d\}|. \end{aligned}$$

An algorithm for sampling from the LDA posterior distribution is the Gibbs sampler [26]. We can also show that the updates for the Gibbs sampler are

$$Pr(z_{ij} = k | z_{-ij}) \propto \frac{(n_k^{(w_i)}(z_{-ij}) + \beta)(n_k^{(d_j)}(z_{-ij}) + \alpha)}{n_k(z_{-ij}) + W\beta},$$

where $n_k(z_{-ij})$ is a count the does not include the assignment of z_{-ij} .

6.2 An interactive protocol

Let's return to our highlighting experiment from the introduction and suppose that we have a helpful user who is able to provide corrective feedback to the topic model. To formalize this we will assume that there is a ground truth topic vector \mathbf{z}^* . The user does not know the values of \mathbf{z}^* but is able to provide feedback in the form of constraints: after seeing a pair $(w_i, d_j), (w_p, d_q)$ the user says “must-link” if $z_{ij}^* = z_{pq}^*$ or “cannot-link” if $z_{ij}^* \neq z_{pq}^*$. Here is the interactive protocol that we consider.

Initialize a set of constraints \mathcal{C} .

Repeat:

1. The user is presented with a pair $(w_i, d_j), (w_p, d_q)$ along with labels z_{ij}, z_{pq} .
2. If $z_{ij}^* \neq z_{pq}^*$ but $z_{ij} = z_{pq}$, the user provides a cannot-link constraint that is added to \mathcal{C} .
3. If $z_{ij}^* = z_{pq}^*$ but $z_{ij} \neq z_{pq}$ the user provides a must-link constraint that is added \mathcal{C} .
4. The algorithm (approximately) samples a topic model which satisfies the provided constraints \mathcal{C} .

6.2.1 An interactive Gibbs sampler

Given feedback in the form of must-link and cannot-link constraints \mathcal{C} , what form does the updated posterior take? Observe that if \mathcal{C} contains a must-link constraint for (w_i, d_j) and (w_p, d_q) , then

$$Pr(z_{ij} = k | z_{-ij}, \mathcal{C}) = \begin{cases} 1 & \text{if } k = z_{pq} \\ 0 & \text{else.} \end{cases}$$

While easy to implement, the above suffers from two issues: (1) we will need to keep track of $O(N^2)$ quantities where N is the size of the corpus and (2) there is no way to explicitly incorporate hard constraints into a Gibbs sampler: any Markov chain with

such behavior will not be irreducible and therefore will not converge to the stationary distribution. To overcome these two difficulties, we will propose a modification to the Gibbs sampler and will instead compute the conditional probabilities for the connected components induced by \mathcal{C} , which we denote by $\text{CC}(\mathcal{C}) = \{s_1, \dots, s_m\}$.

For $s \in \text{CC}(\mathcal{C})$ let $N(s) \subset \mathcal{C}$ denote the set of cannot-link neighbors of s , i.e. $s' \in N(s)$ if and only if there exists a cannot-link constraint in \mathcal{C} for some $(w_i, d_j) \in s$ and some $(w_p, d_q) \in s'$. Then it is not too hard to observe

$$\Pr(z_s = k | z_{-s}, \mathcal{C}) \propto \begin{cases} 0 & \text{if } \exists s' \in N(s) \text{ s.t. } k = z_{s'} \\ \Pr(z_s = k | z_{-s}) & \text{else} \end{cases}$$

The following lemma shows that the conditional probabilities of the constrained posterior distribution can be easily computed.

Lemma 6.1. *For a given connected component s and topic k , if it is the case that there are no cannot link edges between s and any other component with topic assignment k , then*

$$\Pr(z_s = k | x, z_{-s}, \mathcal{C}) \propto \begin{cases} 0 & \text{if } \exists s' \in N(s) \text{ s.t. } k = z_{s'} \\ \frac{p_k^{(w)}(s, z_{-s}) p_k^{(d)}(s, z_{-s})}{p_k(s, z_{-s})} & \text{else} \end{cases}$$

where

$$\begin{aligned} p_k^{(w)}(s, z) &= \prod_{w \in s} \frac{\Gamma(n_k^{(w)}(z) + n^{(w)}(s) + \beta)}{\Gamma(n_k^{(w)}(z) + \beta)} \\ p_k^{(d)}(s, z) &= \prod_{d \in s} \frac{\Gamma(n_k^{(d)}(z) + n^{(d)}(s) + \alpha)}{\Gamma(n_k^{(d)}(z) + \beta)} \\ p_k(s, z) &= \frac{\Gamma(n_k(z) + n_k(s) + W\beta)}{\Gamma(n_k(z) + W\beta)} \end{aligned}$$

(Proof in Section 6.1 of Appendix B.)

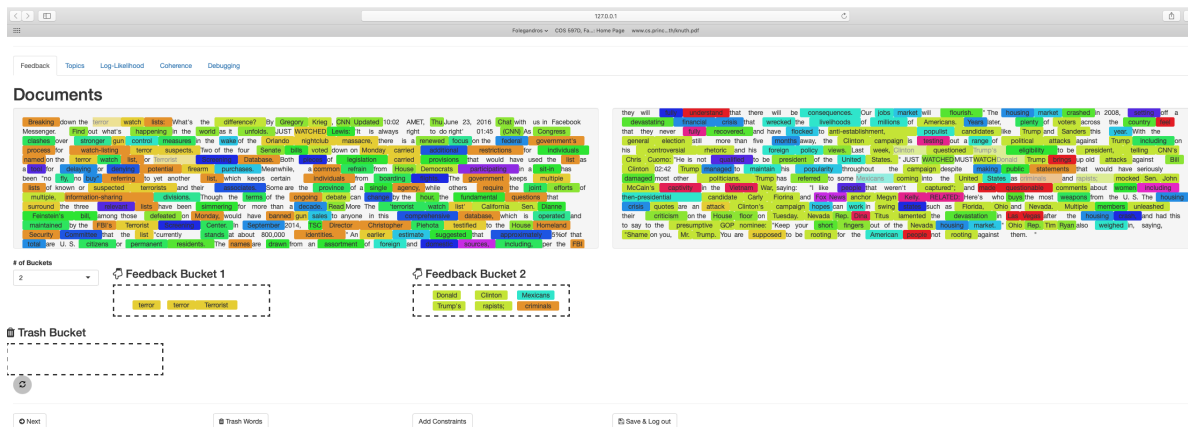


Figure 6.1. An interactive system that implements the constrained-based protocol. Here, the user has grouped words related to “presidential elections” in a bucket and words related to “terrorism” in another bucket

6.2.2 An interactive system

How can our interactive protocol be implemented? An example of a system that implements the protocol is shown in Figure 6.1. The system works follows:

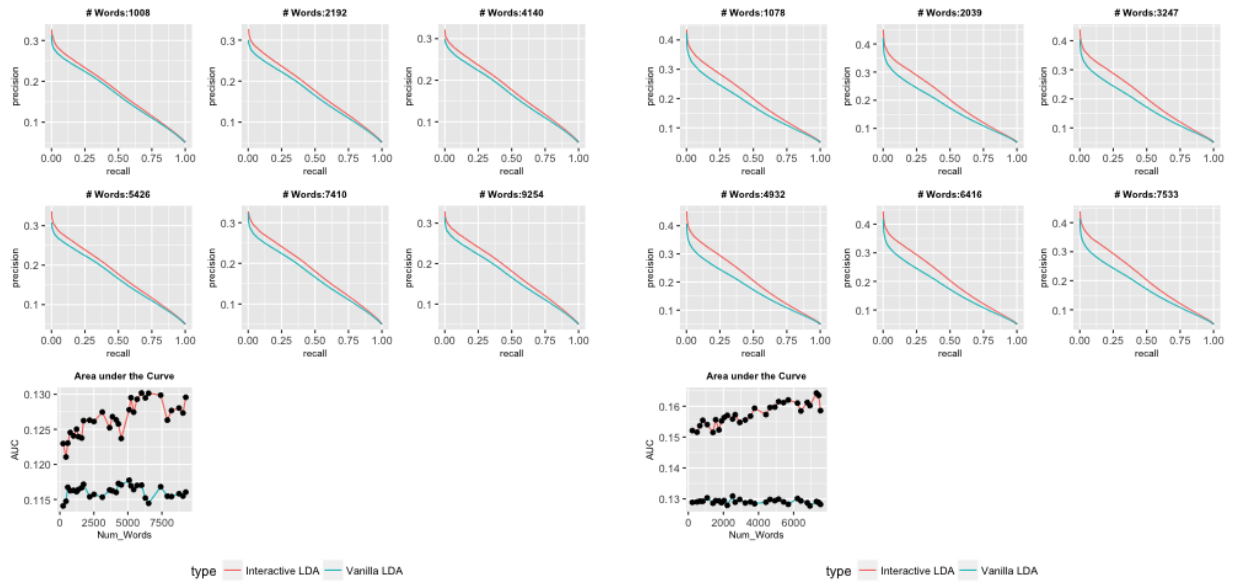
- Initialize a topic model vector $\mathbf{z}^{(0)}$
- For $t = 1, 2, \dots$:
 1. The user is shown a pair of documents. Topic labels according to $\mathbf{z}^{(t-1)}$ can displayed by the different colors.
 2. The user puts must-link words in the same “bucket” and cannot-link words in different buckets. This induces a constraint set \mathcal{C} .
 3. A topic model vector \mathbf{z} is sampled according to Lemma 6.1.
 4. $\mathbf{z}^{(t)} = \mathbf{z}$

6.3 Experiments with an oracle user

In this section, we compare the performance of our interactive LDA model with that of vanilla LDA in the setting of document retrieval. Here, a test document is considered as the *query* and used to retrieve similar documents by performing k -NN classification. We assume that user interaction is aimed towards producing document representations θ that will put documents with the same label close in some distance and documents with different labels further apart. Using labeled documents, we create an “oracle” user that provides sets of constraints that respect the labels of documents. To generate must-link constraints, we consider words whose level of correlation with the label is high. Specifically, for various thresholds α , we considered word j as feedback for class k if $P(k|j)$, the conditional probability of label k given the presence in the document of word j , is at least α . So, every time a new pair of documents is seen, the oracle will select word j from a document with label k , such that $P(k|j) \geq \alpha$. This method was employed in 7.3. If the labels in the pair are different, we create two *buckets* of must-link constraints. Then we generate cannot-link constraints between words in the two different buckets. Now if both labels in the pair are the same, we only create one bucket of must link constraints. We experimented with the **20ng** and **webkb** corpora that were described in 7.3 using 10, 20, and 50 topics. The level of correlation α was set to .5. Results are shown in figures 6.3- 6.2

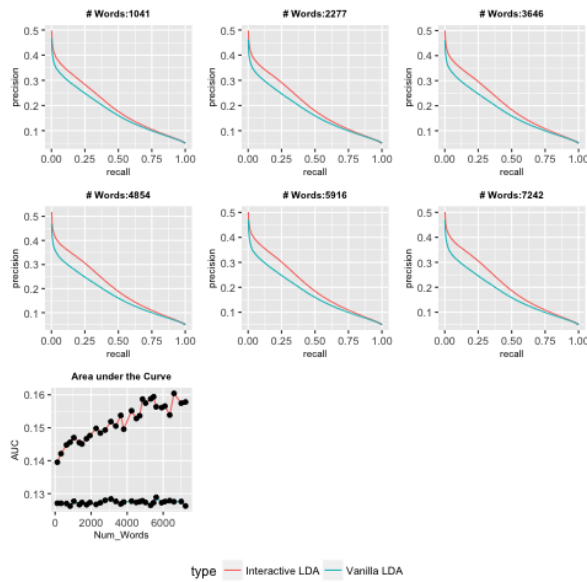
6.3.1 Discussion of simulation results

In Figures 6.3 and 6.2 we display precision and recall curves (first 6 panels in each figure) at various stages of the simulation. Also, in the last panel of each figure we display results throughout all the rounds of the simulation, in terms of the area under the precision-recall curve. As it can be seen, our interactive protocol outperforms Vanilla LDA in all our experiments.



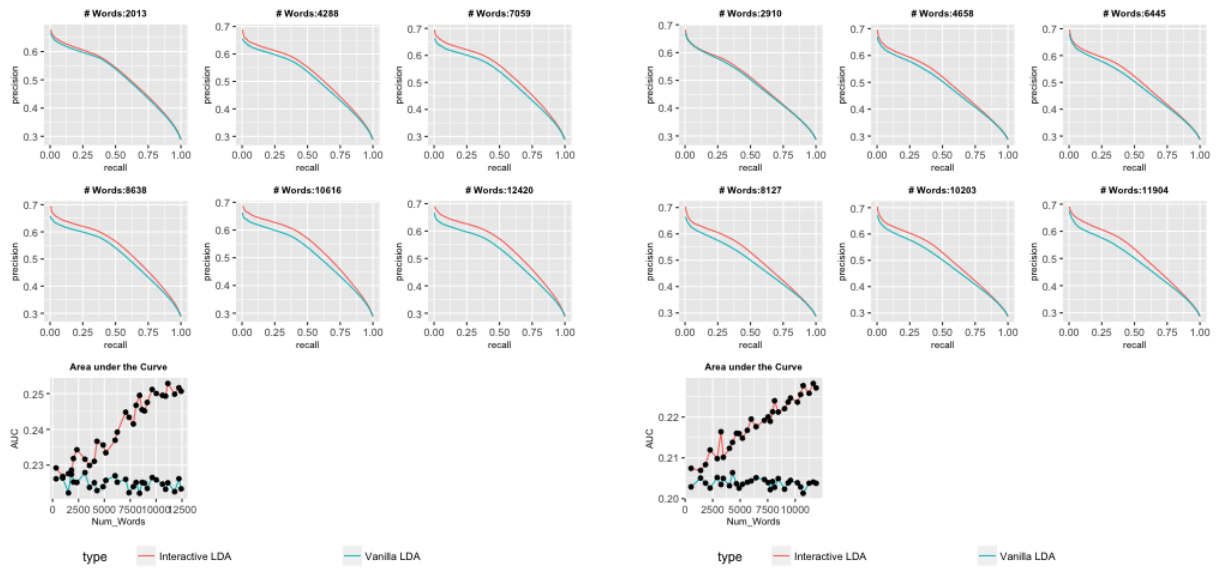
(a) 10 topics

(b) 20 topics



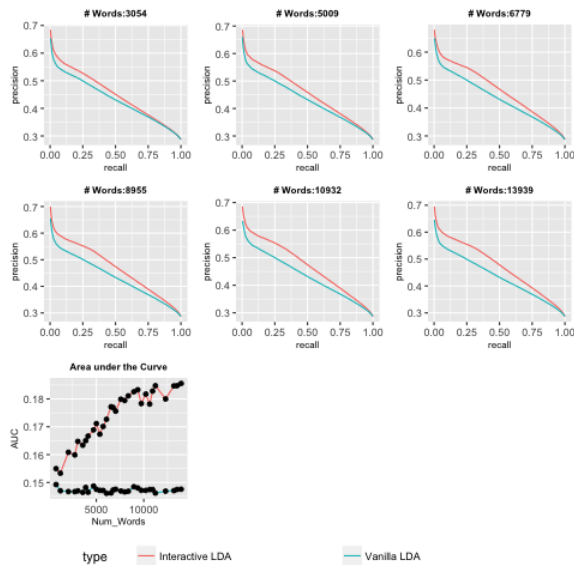
(c) 50 topics

Figure 6.2. Experiments on the 20ng data set. The first six panels in each figure show precision and recall curves in the various rounds. The last panel shows area under the precision and recall curve.



(a) 10 topics

(b) 20 topics



(c) 50 topics

Figure 6.3. Experiments on the Webkb data set. The first six panels in each figure show precision and recall curves in the various rounds. The last panel shows area under the precision and recall curve.

One downside to our interactive approach is the amount of feedback that is required. Although our protocol allows a user to directly affect the latent topic assignments, it may not be practical for real-world applications. In the next section, we present our anchor word-based interactive protocol, which is substantially more practical.

Chapter 6 contains material that is currently being prepared for submission for publication of the material. S. Poulis, S. Dasgupta, C. Tosh. The dissertation author was the primary investigator.

Chapter 7

Interactive topic modeling with anchor anchors

In this chapter we will present an approach that overcomes the limitations of our previous approach using anchor words. We start with some preliminaries in below.

7.1 Preliminaries

A *corpus* is a collection of documents d_1, \dots, d_m , each of which is represented in the bag-of-words representation as a vector in \mathbb{Z}_+^V , where V is the size of the vocabulary. A *word-topic matrix* is a $V \times K$ matrix A such that each column A_i corresponds to a topic and is represented as an element of Δ_V , the V -dimensional probability simplex.

Given a word-topic matrix A and a prior distribution $\tau \in \Delta_K$, here is the generative model for a corpus:

- For each document $d = 1, 2, \dots$:
 - Draw a topic distribution $p_d \sim \tau$
 - For word i in document d , draw its topic $z_i \sim p_d$ and then draw the vocabulary word $w_i \sim A_{z_i}$.

Together, the matrix A and distribution τ induce a word co-occurrence matrix

$Q \in \mathbb{R}^{V \times V}$ and topic co-occurrence matrix $R \in \mathbb{R}^{K \times K}$ satisfying

$$Q_{i,j} = \Pr(w_1 = i, w_2 = j) \quad \text{and}$$

$$R_{k,k'} = \Pr(z_1 = k, z_2 = k')$$

for a randomly generated document with words w_1 and w_2 with associated topics z_1 and z_2 .

We say that a word i is an *anchor word* for topic k if $A_{i,k} \gg 0$ and $A_{i,k'} = 0$ for all other topics $k' \neq k$. Further, we say that the topic matrix is separable if each topic k has an associated anchor word s_k .

Given such a corpus, several algorithms have been designed to provably recover the anchor words of a topic model and the topics associated with them [4, 47, 5]. The general approach is given in Figure 7.1. In this work, we will assume that we have access to such procedures and their subroutines.

1. **Compute normalized word co-occurrences.** Form the $V \times V$ matrix \bar{Q} , where $\bar{Q}_{ij} = \Pr(w_2 = j | w_1 = i)$. The rows of \bar{Q} lie in Δ_V .
2. **Identify the anchor words.** Find K rows of \bar{Q} , say s_1, \dots, s_K , such that the rest of the rows lie approximately in the convex hull of the \bar{Q}_{s_i} . These are the anchor words.
3. **Express all rows as convex combinations of anchor rows.** For each word i , find positive weights $C_{i,1}, \dots, C_{i,K}$ summing to 1 such that $\bar{Q}_i \approx C_{i,1}\bar{Q}_{s_1} + \dots + C_{i,K}\bar{Q}_{s_K}$. Then $C_{i,k} \approx \Pr(z = k | w = i)$.
4. **Recover the topic distribution.** By Bayes' rule: $A_{i,k} = \Pr(w = i | z = k) \propto C_{i,k}\Pr(w = i)$.

Figure 7.1. The generic anchor words algorithm.

7.2 An anchor word based interactive protocol

As pointed in the introduction, there are many difficulties associated with topic modeling as a purely unsupervised task. These include the identification of the correct number of topics, filtering out noise, and dealing with the inherent ambiguities of language. Moreover, different users may have different desiderata in a topic model that may not be possible to satisfy simultaneously.

To address these issues, several methods have been considered for injecting human knowledge into topic modeling. The approach with the closest resemblance to our own is the recently proposed *anchor facet* approach [38]. In this method, a user synthesizes pseudo-anchor words by averaging together subsets of words the user chooses. As we will see, these pseudo-anchors disregard the underlying geometry of the data in ways that can lead to problems in topic recovery.

The remainder of this section is organized as follows. We first give an example where the anchor facet approach leads to identifiability issues. Next, we present a generative model for which standard unsupervised techniques cannot recover the desired topics, even in the infinite data limit. Finally, we present our interactive protocol which can, in fact, find good estimates of the desired topics.

7.2.1 An anchor facet problem

In the anchor facet model, a user chooses a set of words \mathcal{G} from the vocabulary that they feel should represent a topic. For instance, they might choose **games** and **computer** to indicate a ‘computer games’ topic. The corresponding word co-occurrence vectors (rows of \bar{Q}) are then aggregated to form a *pseudo-anchor* g , by taking their harmonic mean (among other options), and this g is added to the set of anchor words. After the user has created the pseudo-anchors, a topic model is recovered using steps 3-4 of Figure 7.1.

This approach is intuitively appealing but hard to justify geometrically. The

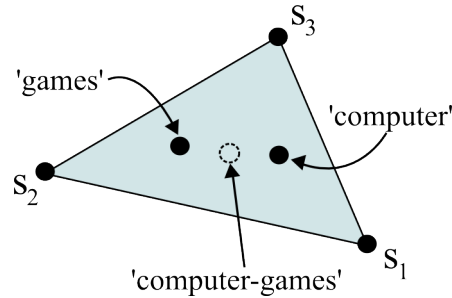


Figure 7.2. Illustration of anchor facet shortcoming. Here the user combines anchor words ‘computer’ and ‘games’ which results in a point ‘computer-games’ somewhere in the middle of the simplex spanned by s_1 , s_2 and s_3 .

correctness of the anchor words algorithm depends on the anchors being at the corners of the simplex containing all the word vectors. Pseudo-anchors violate this in two ways: (1) they don’t have a clear meaning in terms of co-occurrence probabilities (if, as suggested, the harmonic mean is used for aggregation) and (2) they may well lie near the center of the simplex. For instance, it could easily happen that a large fraction of the remaining words are not well-approximated as convex combinations of pseudo-anchors; in which case, these words will be assigned to topics in a fairly arbitrary manner.

7.2.2 A subtopic view of document generation

The topic modeling view of data generation, that a corpus is generated by a relatively small number of topics that are easily interpretable by a human, is often an oversimplification. In reality, documents on similar subjects can vary wildly in their choice of language due to authorship, the times they were written, etc. A topic model that accurately fits a real corpus must necessarily contain many topics.

To see this, imagine a corpus of news documents collected over the course of a year, in which a small but significant percentage of articles deal with weather. A user wishing to analyze this corpus via topic modeling might be satisfied with a single weather topic. However, the corpus itself will not look like it only has a single weather topic. Indeed, the distribution of words in a weather article written in September during hurricane season

will look significantly different from the distribution of words in a weather article written in January during blizzard season, which in turn will look significantly different from the distribution of words in a weather article written in July during drought season. Thus, accurately modeling the weather-related aspects of the corpus requires several topics. And that is just the weather! Conceivably, other aspects of the corpus for which a user might imagine a single topic sufficing can in turn be broken into components that actually model the data.

On the other hand, a model with hundreds or thousands overlapping and highly correlated topics is not easy to work with. Many users would prefer a significantly simpler model that may not perfectly describe the data but summarizes the core subjects well.

To model the mismatch between the idealized view a user has in mind and the actual data generating process, we introduce the *subtopics view* of data generation. It is described by the following generative model.

- There are several ‘ideal’ topics M_1, \dots, M_K along with some topic-topic co-occurrence matrix $R^M \in \mathbb{R}^{K \times K}$.
- For each ideal topic M_k , some number of ‘subtopics’ indexed by the set \mathcal{G}_k are drawn i.i.d. from a distribution satisfying $\mathbb{E}[A_t] = M_k$ for each $t \in \mathcal{G}_k$.
- The corpus is generated according to the new topic matrix A and some topic-topic co-occurrence matrix R^A satisfying $\sum_{t \in \mathcal{G}_k} \sum_{t' \in \mathcal{G}'_k} R^A_{tt'} = R^M_{kk'}$.

Here we call the topic model induced by M and R^M as the *idealized model* and the topic model induced by A and R^A the *subtopics model*. Intuitively, the idealized model is the model that would have generated the corpus in an ideal world, e.g. an ideal weather topic. However, the corpus is actually generated by the subtopics model with a larger number of more specific topics, e.g. hurricane, blizzard, and drought topics. As

the following lemma shows, the co-occurrence matrix induced by a subtopic model is intrinsically biased away from the idealized model in expectation.

Lemma 7.1. *If Q^M is the co-occurrence matrix induced by the idealized model and Q^A is the co-occurrence matrix induced by the subtopics model, then*

$$\mathbb{E}_A[Q^A] = Q^M + \sum_k R_{k,k}^M \Sigma^{(k)}$$

where $\Sigma^{(k)}$ is the covariance matrix of the subtopic distributions generated under ideal topic k .

Proof. Fix words i, j and subtopic matrix A . Then

$$\begin{aligned} Q_{i,j}^A &= \Pr(w_1 = i, w_2 = j) \\ &= \sum_{t,t'} \Pr(w_1 = i | z_1 = t) \cdot \Pr(w_2 = j | z_2 = t') \\ &\quad \cdot \Pr(z_1 = t, z_2 = t') \\ &= \sum_{t,t'} R_{t,t'}^A A_{i,t} A_{j,t'} \end{aligned}$$

Taking expectations of this with respect to the A 's and noting that (i) A_t and $A_{t'}$ are independent for $t \neq t'$ and (ii) $\mathbb{E}[A_t A_t^T] = \Sigma^{(k)} + M_k M_k^T$ for all $t \in \mathcal{G}_k$, we have

$$\begin{aligned} \mathbb{E}[Q_{i,j}^A] &= \sum_{k,k'} \sum_{t \in \mathcal{G}_k} \sum_{t' \in \mathcal{G}_{k'}} \mathbb{E}[R_{t,t'}^A A_{i,t} A_{j,t'}] \\ &= \sum_{k \neq k'} R_{k,k'}^M M_{i,k} M_{j,k'} + \sum_k \sum_{t,t'} \mathbb{E}[R_{t,t'}^A A_{i,t} A_{j,t'}] \\ &= \sum_k \sum_{t \neq t'} R_{t,t'}^A M_{i,k} M_{j,k} + \sum_k \sum_t R_{t,t}^A \left(M_{i,k} M_{j,k} + \Sigma_{i,j}^{(k)} \right) \\ &\quad + \sum_{k \neq k'} R_{k,k'}^M M_{i,k} M_{j,k'} \end{aligned}$$

$$= \sum_{k,k'} R_{k,k'}^M M_{i,k} M_{j,k'} + \sum_k R_{k,k}^M \Sigma_{i,j}^{(k)} \quad \square$$

The above lemma shows that in general the co-occurrence matrix generated by the subtopics model is biased away from the co-occurrence matrix that would be generated by the idealized model. Indeed, in the special case where $\Sigma^{(k)} = \Sigma$ for $k = 1, \dots, K$, the above reduces to

$$\mathbb{E}_A[Q^A] = Q^M + \text{tr}(R^M) \cdot \Sigma.$$

Thus, directly fitting a topic model based on these statistics should not in general recover the ideal topics. Rather, some other approach is needed.

7.2.3 An interactive protocol

How do we recover the idealized topics M ? Returning to our weather example, we could start by fitting a model with say, 500 topics. Next, we could ask a user to peruse these, form a group of some good weather subtopics e.g. hurricane, blizzard, drought, etc., and then average subtopics in the group to get an estimate of an ideal weather topic. But the way the topics are displayed presents a challenge: perusing 500 topics and finding their salient groupings might place an overwhelming cognitive load on a user. Indeed, even if each subtopic is uniquely identified by its top 10 words (which often is not the case), a prospective user would have to wade through 5000 words! What is needed, then, is a way to ensure we have a unique representation for each topic and to present these to the user as succinctly as possible.

Our approach is to utilize anchor words. Assuming each subtopic is associated with an anchor word, we find an ‘overcomplete’ list of anchor words s_1, \dots, s_T and present these to a user as proxies for entire topics. The user can quickly sort through this list and easily identify subtopics by their component anchor words. After a few rounds, the user will form K groupings of selected anchor word indices $\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_K \subset \{1, \dots, T\}$.

It is possible that there are anchor words that a user simply does not recognize as significant, perhaps because they are uninteresting background topics. Thus, we do not require a complete partition of the anchor words from the user.

Given a corpus generated by the subtopic model, our interactive protocol for estimating the M_k 's is relatively simple and it is given in Figure 7.3. It is not hard to see that if our estimates of the subtopics A are unbiased and the user correctly identifies each true subtopic group \mathcal{G}_k , then each estimate \widehat{M}_k will be close to the ideal topic M_k .

Issues arise when, due to undersampling, the set of candidate anchor words contains words that are not true anchor words. These ‘spurious’ anchor words disrupt our ability to estimate the subtopics, leading to errors in our estimates of the ideal topics. To counter this issue, we consider an alternative procedure that replaces step (c) with the following:

- (c') Using only the anchor words selected by the user, estimate the topic vectors \widehat{A}_j for each $j \in \mathcal{G}_1 \cup \dots \cup \mathcal{G}_K$ by running a topic recovery algorithm.

We call the algorithm that uses step (c') *partial interactive recovery* to distinguish it from the *full interactive recovery* algorithm that uses step (c).

- (a) Identify the ‘candidate’ anchor words s_1, \dots, s_T via a standard anchor-finding algorithm.
- (b) Present these to the user and receive K groupings of selected anchor word indices $\widehat{\mathcal{G}}_1, \dots, \widehat{\mathcal{G}}_K \subset \{1, \dots, T\}$.
- (c) Using all the anchor words, estimate the topic vectors $\widehat{A}_1, \dots, \widehat{A}_T$ by running a topic recovery algorithm.
- (d) For each group $\widehat{\mathcal{G}}_k$, average the associated topic vectors $\widehat{M}_k = \frac{1}{|\widehat{\mathcal{G}}_k|} \sum_{j \in \widehat{\mathcal{G}}_k} \widehat{A}_j$.

Figure 7.3. Full interactive recovery algorithm

7.3 Experiments

In this section we study real and simulated users in a variety of experiments. First, we simulate a user looking to recover the *ideal* topics from a synthetic dataset of documents generated by our subtopic model. Next, we look at a real dataset and simulate a user seeking to produce a topic model that results in meaningful document representations, that is, documents that share similar subjects should have similar representations. In our final experiment we explore if real users, equipped with our interactive tools, can understand the main aspects of the corpus that they are analyzing and can create topic models that are interpretable.

7.3.1 Topic recovery in simulated subtopic model

We considered the problem of recovering an ideal topic model, given data generated by a subtopic model. To do this, we generated K ideal topics ϕ_1, \dots, ϕ_K from a symmetric Dirichlet(α) distribution. For each topic ϕ_i , we generated m subtopics, drawn from the non-symmetric Dirichlet(ϕ_i/σ) distribution. Finally, we generated D documents using these subtopics and a symmetric Dirichlet(β) distribution over the document-topic distributions.

We compared the anchor group approach of this paper against the non-interactive anchor word approach of [5], the anchor facet approach of [38], and the constraint-based approach of [28]. For the anchor group and anchor facet approach, we generated $m \cdot K$ anchor words and grouped together anchor words whose resulting topics are closest to each of the corresponding underlying topics. For the constraint-based approach, we created the SPLIT and MERGE constraints based on the highest probability words of each of the K underlying topics. For the experiments involving the constraint-based approach, we ran the tree-structured Gibbs sampler between 100-200 iterations as in [28].

Figure 7.4 displays the average errors of the resulting models. For both ℓ_1 and ℓ_2 error, the anchor group approach of this paper performed the best.

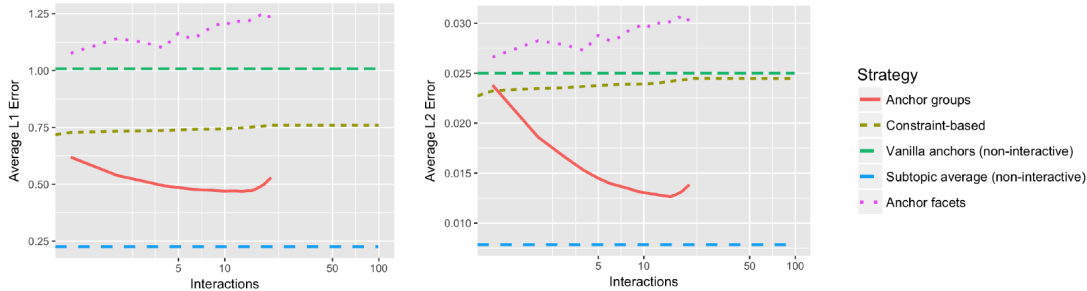


Figure 7.4. Recovery of underlying topics using different forms of interaction. Subtopic average is the topic model created by averaging together all of the underlying subtopics.

7.3.2 Document representation experiment

To compare the quality of the topic models produced by the various algorithms, we conducted an experiment on the inferred document representations produced by these models. We used the **20 Newsgroups** dataset,¹ which consists of $\approx 18\text{K}$ documents each belonging to one of 20 categories.

We again compared our anchor group approach against the anchor facet approach of Lund et al. [38] and the constraint-based approach of Hu et al. [28]. We ran the anchor finding algorithm of Arora et al. [5] to generate 500 candidate anchor words. For the interactive anchor-based approaches, we calculated

$$g(a, c) = \frac{\# \text{ times } a \text{ occurs in document with label } c}{\# \text{ times } a \text{ occurs in corpus}}$$

for each anchor word a and each news group category c ; and for each category c , we selected the 10 anchor words a with the highest $g(a, c)$ value. Table 7.1 shows the anchors for each news group. For topic recovery, we used RECOVERL2 of [5].

For the constraint-based approach, we calculated $g(a, c)$ for all words a , not just anchor words, and selected the 10 words a with the highest $g(a, c)$ value for each category c . For the resulting grouping, we generated all of the corresponding SPLIT and MERGE constraints.

¹<http://qwone.com/~jason/20Newsgroups>

Table 7.1. Simulated anchors for each news group category

alt.atheism:	cco amus rice tek contradict wisc philosoph islam satan sincer
graphics:	viewer svga hidden bitmap vga gif render pointer transform routin
os.ms-windows.misc:	cica desktop challeng diamond beta swap zip icon ati brett
sys.ibm.pc.hardware:	gatewai motherboard isa jumper bio ati cach interrupt viru slower
sys.mac.hardware:	centri quadra horizont slot simm ethernet newer iii soni connector
windows.x:	sparc motif pointer xterm client bitmap compil patch athena widget
misc.forsale:	forsal stereo dual speaker super genesi soni bag gold sam
rec.autos:	valv transmiss bird truck turbo honda steer ecn cylind tight
rec.motoreycles:	rider honda helmet drink steer shaft shoulder chain dog infant
rec.sport.baseball:	hitter philli giant era baltimor bond morri relief plate talent
rec.sport.hockey	goalie bruin penguin nhl quebec winnipeg jersei leaf ranger tie
sci.crypt:	sternlight ggp den cellular lobbi eff colost transmit graham perri
sci.electronics:	amp motorola audio isol nois batteri uga transform filter acid
sci.med:	geb physician cure diet sensit skin aka infect russel nose
sci.space:	zoo alaska spacecraft digex flight solar astronomi uxa apollo jpl
soc.religion.christian:	gospel revel resurrect uga hebrew prayer vers prai inspir soc
talk.politics.guns:	dividian cdt handgun cnn reno packet boulder cult bullet cco
talk.politics.mideast:	melkonian serdar propaganda holocaust jerusalem slaughter hatr carter bosnia bosnian
talk.politics.misc:	cramer partner reform libertarian decad incom sexual ncr acc reno
talk.religion.misc:	sandvik albert mormon cult inspir miracl gospel contradict promis arizona

To evaluate the quality of the competing topic models we looked at the local neighborhood structure of the resulting document representations using a k-nearest neighbor (k-NN) classifier. For a given topic model with m topics, we embedded the documents into the m -dimensional probability simplex using LDA [26]. We then computed the leave-one-out cross-validation (LOOCV) accuracy of the k-NN classifier over a sample of 2K embedded documents. Table 7.2 presents the performances of the resulting k-NN’s for varying values of k on several interactive and non-interactive methods. All interactive methods had 20 topics (one for each news group category), whereas the number of topics varied for the non-interactive ones.

We observed that for all values of k, our interactive algorithms (FULL and PARTIAL) outperformed all other interactive and non-interactive approaches.

Table 7.2. K-NN accuracy under various algorithms.

Model	k = 10	k = 20	k = 50	k = 100
FULL 20	0.330	0.324	0.309	0.273
PARTIAL 20	0.337	0.337	0.321	0.287
LUND ET AL. 20	0.236	0.223	0.197	0.173
HU ET AL. 20	0.221	0.212	0.196	0.178
ALL 500	0.218	0.193	0.155	0.126
SELECT 200	0.228	0.199	0.158	0.130
VANILLA 20	0.144	0.140	0.133	0.121

7.3.3 User study

We conducted a small-scale user study to evaluate the anchor group interactive algorithm. Five users were asked to create their own topic model based on a corpus of recent news articles. All users were doctoral students in computer science, three of whom had past experience with topic modeling.

Data collection and preprocessing We used a collection of news articles crawled from the CNN website as its corpus; it was provided to us by a commercial search engine. The corpus contained about 10K articles, starting from around April 2016 and spanning about year. The articles covered a diverse range of subjects including politics, economy, sports, technology, science, and law. It also spanned several notable events such as the 2016 U.S. presidential debates and election, the 2016 Olympics games, and the Brexit referendum. It is also worth noting that since the dataset was created by a crawler, some articles contain boilerplate content such as advertisements and links to other irrelevant articles, which we did not take any steps to remove. We also did not perform any stemming. We only removed stop words and kept words that occurred in at least 10 documents. The final vocabulary contained about 17K words. After running an anchor word algorithm [47], we had a list of about 500 anchor words as the basis of our interactive interface.

Interactive process User feedback was collected via a web-based interface. At the beginning, users were prompted to select an element from the list of anchor words. After a word was selected, the user was taken to a separate screen where they created a topic by grouping words they felt were similar enough to the originally selected word. Figure 5.5 shows an instance of a user that has chosen to create a topic by merging the words ‘hackers’, ‘computer’, ‘fbi’ and ‘messages’. The box to the right displays a suggestion of 10 anchor words that are closest in ℓ_1 distance to the group of anchor words already in the topic.² This component of the interface made topic creation more efficient by reducing the number of anchor words a user scanned to create a group.

Perusing a list of 500 words many times can be taxing on a user. To help users better traverse the space of anchor words, the interface had four additional features.

- *Complete topic*: After merging anchors into a topic, the user could complete the topic with anchors suggested by the system. The suggested words were sorted by ℓ_1 distance.
- *Merge topics*: The user was given the option to merge two or more created topics into one.
- *Delete topic*: The user was given the option to delete a grouping they had created.
- *Suggest topics*: When creating a new topic the user was given the option to hit a button that suggested new anchors. The system highlights words that are further away in ℓ_2 distance from the space spanned by the words already selected by the user.

Appendix B contains a step-by-step instance of the interactive procedure, including the starting list of anchor words and each of the above functions.

²For a group of words S already in the topic, we sort each word $w \notin S$ according to their distance from the set S : $d(w, S) = \min\{\|w - w'\|_1 : w' \in S\}$.

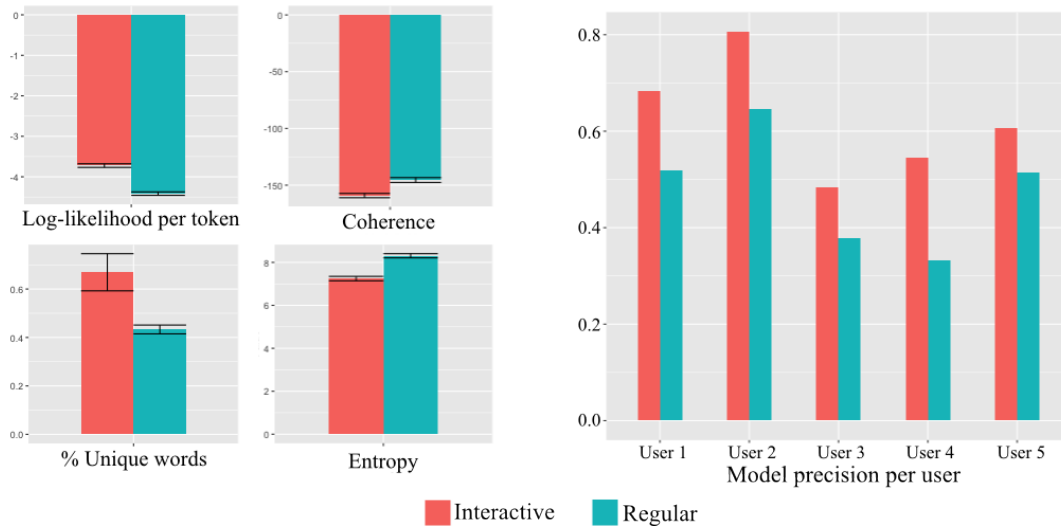


Figure 7.5. *Left:* Log-likelihood per token, coherence, % unique words, average entropy of topics. *Right:* Per-user performance on word intrusion task. Users were tested on all user-created topics they created.

Results Before starting the process, users were given some brief information about the dataset and then asked to create topics that would best summarize it. Using the interactive process described above, users created groupings of anchor words. Some examples of these groupings are given in Table 7.3.

Table 7.3. Examples of user anchor groupings.

1	russian putin intelligence agencies
2	olympics rio olympic athletes brazil sport winner
3	hollywood movie entertainment star film character original awards controversy
4	joe politics vice rubio cruz kasich ballot campaigns convention voting poll delegates elections pennsylvania
5	israel peace region council terrorist terror isis suicide iraqi falluja iraq troops syrian syria aleppo refugees turkey

After collecting the feedback that users provided, we used the partial interactive recovery algorithm of Section 7.2.3 with the RECOVERL2 of [5] to learn a topic model for each user. We call models created by user interactions **Interactive**. For each user, we also learned a topic model with the same number of topics without any interaction using

Algorithm 1 from [5]. We call these models **Regular**.

Qualitative assessment of topics Shaded rows of Table 7.4 give the most probable words under topics learned using the user feedback of Table 7.3. Unshaded rows show the most probable words under the topic of the regular method that was closest in ℓ_1 distance to the one above it.

Table 7.4. Most probable words for the user created topics shown in Table 7.3.

1	Interactive	russian putin russia intelligence obama
	Regular	obama president trump clinton visits
2	Interactive	rio olympic olympics games athletes
	Regular	minister prime company million published
3	Interactive	film star show awards disney
	Regular	trump comedy show company million
4	Interactive	cruz kasich president clinton convention
	Regular	trump clinton donald campaign trumps
5	Interactive	falluja isis battle syrian forces
	Regular	attacks brussels terror airport police

Across the board, the interactive method resulted in better quality topics that seemed to align with the intentions of the user that created them. Moreover interactive topics seemed more easily interpretable and more general than the topics of the regular method. For example, looking at topics 1 and 2 in Table 7.4, one can see that the interactive method yielded topics that matched what the user was trying to achieve. (See groupings 1 and 2 in Table 7.3.) We observe a similar situation for topic 5, for which the interactive method yielded a topic related to current events in the Middle East, while the regular method yielded a very specific topic about the Brussels terror attack. Tables B.1-B.5 in Appendix B show a complete comparison for all users.

Word intrusion user evaluations As noted in the introduction, a popular way to understand the gist of a topic is to look at its n most probable words and try to find their common theme. *Word intrusion* seeks to quantify how easily one can interpret a topic

model in this way [15]. Roughly, for each topic, its list of n most probable words will be *intruded* by a word that is in the n most probable words of another topic. Humans are asked to find the *intruding* words and models are then scored according the % of intruding words found by humans. One would expect that in a semantically coherent list of words, intruding words will be more easily detected.

To measure word intrusion, each user that participated in the study was asked to evaluate a mix of their own and of other users' topics, as well as the topics of the regular method. The number of words that were shown was $n = 10$. Figure 7.5 (right) shows the results of this experiment. Across the board, users performed better on the word intrusion when they were evaluating an interactive topic as opposed to one found by purely unsupervised methods, even when those interactive topics were created by other users.

Quantitative metrics We also compared the two methods across different metrics. We looked at log-likelihood, semantic coherence, which was introduced by Mimno et al. [41], proportion of unique most probable words, and entropy. To calculate log-likelihood, we ran 100 iterations of the Gibbs sampler while keeping the topics of each method fixed. Figure 7.5 (left) shows the different metrics. Averaged across users, the interactive method has slightly higher per token log-likelihood but slightly worse topic coherence at the top $n = 10$ words. Also, the interactive method has more unique most probable words per topic (again for $n = 10$), indicating models that capture topics that are different from each other. Finally, the interactive method has lower entropy, indicating that on average, its topics concentrate on a smaller number of words than the regular method.

7.4 Discussion and future work

In this part of the thesis we considered interactive topic modeling. We studied two different protocols: constraint-based and anchor word-based. Our constraint-based protocol treats the task of topic modeling as one of clustering. It enables a user to perform corrections to a topic model on the spot and if the user took enough time, the desired target clustering could be specified completely. In contrast to previous work where user feedback was incorporated only as soft constraints, our interactive Gibbs sampler allows us to translate this feedback into hard constraints.

The drawback of the constraint-based approach is that the amount of feedback it requires might be prohibitive. Our anchor word-based protocol allows users to trigger large structural changes into a topic model and enables them to quickly create interpretable topics.

One interesting future research direction is to combine the two approaches. For instance, one could restrict the constraint-based interaction to anchor words. Because the presence of an anchor word in a document is sufficient evidence that the subject of document is at least partially about the topic of the anchor word, an interactive user may only need to focus on those. This formulation has the potential to make our constraint-based protocol more efficient.

Chapter 7 contains material as it appears in “Interactive topic modeling with anchor words.” International Conference on Machine Learning 2019. S. Poulis, S. Dasgupta, C. Tosh. The dissertation author was the primary investigator.

Part III

Interactive machine teaching

Chapter 8

Introduction

Machine teaching [25, 51, 3] is the problem of efficiently constructing a dataset that a student’s model will learn from. In principle, machine teaching aims to construct an optimal (usually minimal) such dataset. In contrast to traditional machine learning where training data come from an underlying distribution, as in the statistical learning framework of [56] or are chosen in arbitrary and possibly adversarial manner, as in the online learning framework of [33], data in machine teaching are chosen by a teacher, who knows how to select *helpful* training examples.

Machine teaching is found in several real-world applications. One example is utilizing a teacher (e.g. a domain expert) to train a text classifier. The teacher can teach either by selecting documents from a corpus or even by writing some new ones. These will be used as training data by the text classifier’s learning algorithm. The teacher could conceivably come up with plenty of teaching examples but how can the teacher construct an optimal set?

8.0.1 Cases of machine teaching

In the example above the teacher is a human and the learner is a machine. More generally, machine teaching has the form “Teacher teaches Student” and applications may differ depending on who the teacher and who the student is. In addition to the “Human teaching Machine” example of the text classifier, here are some more cases of machine

teaching.

- “Machine teaching Human”: intelligent tutoring systems, say when the system is teaching vocabulary of a foreign language. The system may ask the student questions about any unmastered vocabulary words.
- “Machine teaching Machine”:
 - Sample compression: given an arbitrary list of labeled examples, retain only a subset of them in a way that allows to recover the labels of all other examples in the list [34, 42].
 - Model compression: given a large, slow, but accurate model, compress it into a much smaller, faster, yet still accurate model [13].
- “Human teaching Human”: psychology, pedagogy. Modeling cognition by choosing which examples to present and in which order to present them.

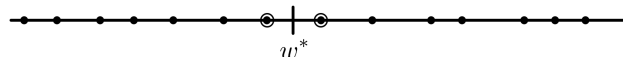
To model such situations several notions of teaching have been developed. One influential model, introduced independently by [25], [51], and [3], is based on the notion of a *teaching set*. Here is a formal definition.

Definition 8.1. *Let \mathcal{X} be any finite instance space and \mathcal{H} any finite set of concepts on \mathcal{X} , so that each $h \in \mathcal{H}$ is of the form $h : \mathcal{X} \rightarrow \{0, 1\}$. Let $h^* \in \mathcal{H}$ denote a target concept. We say $S \subset \mathcal{X}$ is a teaching set for (h^*, \mathcal{H}) if h^* is the only concept in \mathcal{H} that is consistent with the labeled examples $\{(x, h^*(x)) : x \in S\}$.*

An *optimal teacher* is then one who provides the learner with the smallest possible teaching set. The size of this minimal teaching set is often called the *teaching dimension* of the learner’s model.

8.0.2 How much the teacher knows about the student?

Perhaps the most illustrative motivating example for machine teaching is that of thresholds on the line. Here, the target concept is simply a threshold $w^* \in \mathbb{R}$; the input space consists of real numbers, so $\mathcal{X} \subset \mathbb{R}$, and the hypothesis class is $\mathcal{H} = \{h_w : w \in \mathbb{R}\}$, where

$$h_w(x) = \begin{cases} 1 & \text{if } x \geq w \\ 0 & \text{otherwise} \end{cases}$$


In this case, the optimal teaching set consists of the two points in \mathcal{X} nearest w^* , on either side of it.

In this example the teacher is required to know \mathcal{H} , the learner's hypothesis class (here, we will use the terms hypothesis and model interchangeably). This however, may be too strong of an assumption for certain scenarios. When teaching a human for instance, one generally has no idea what the underlying hypotheses might be!

To put this into context, consider a geologist who may want to teach students to categorize rocks into igneous, sedimentary, metamorphic etc. and teaches by picking informative rock samples to show the students. The geologist may know the target hypothesis but there is no way to “transmit” it into the students' minds.

Similarly, when teaching a machine, the general type of concept might be known (a neural net, for instance), but the specifics (number of layers, number of nodes per layers, other parameter settings) may be opaque; and even if they were known, it is unclear how they would be used in choosing a teaching set.

The above scenarios show that requiring the teacher to know the learner's model \mathcal{H} may be unrealistic. Teaching may arguably be more realistic when the teacher does not know the learner's model, i.e. when the learner is a *black-box*. How can a teacher teach when the learner is a black-box?

When \mathcal{H} is known (i.e. when the learner is not a black-box), teaching does not need to be interactive: the teacher just needs to construct a batch of teaching examples beforehand and provide them to the learner in one shot; thereafter, the teacher does not need to see what the learner does with the provided teaching examples.

In this part of the thesis we study the problem of teaching a black-box learner. In particular, we illustrate that teaching such a learner can only be achieved with an interactive teacher. We consider a setting in which the teacher interacts with the learner in rounds: in each round the teacher is allowed to probe the predictions of the learner’s current model, rather like giving the learner a quiz and provides teaching examples accordingly. Intuitively, this strategy allows the teacher to get a better sense of where the learner’s model is and to pick teaching examples more intelligently. Figure 8.1 contrasts non-interactive teaching to the interactive teaching setting that we consider here. We show that without knowing \mathcal{H} , an interactive teacher can pick a teaching set of size at most $O(t \cdot \log |\mathcal{X}| \cdot \log |\mathcal{H}|)$, where t is the optimal teaching set size for \mathcal{H} .



Figure 8.1. Left: A non-interactive teacher that provides examples in one shot. Right: An interactive teacher.

8.0.3 Overview

The rest of Part III is organized as follows. In Section 8.0.4 we start by reviewing previous work.

In Section 9.1 we continue by demonstrating through an example a negative result for non-interactive teaching. We consider a scenario in which there are multiple hypothesis classes $\mathcal{H}_1, \dots, \mathcal{H}_k$ that all contain the target h^* . We require that the teacher only knows

that the learner’s hypothesis is one of $\mathcal{H}_1, \dots, \mathcal{H}_k$ but not which one and continue by showing that under this scenario, a non-interactive teacher must construct a teaching set that consists of all of \mathcal{X} .

In Section 9.2 we present interactive teaching. We consider scenarios in which the teacher has *no knowledge* of the learner’s hypothesis class other than an upper bound on its size or VC dimension. It does not, for instance, have a shortlist of possibilities $\mathcal{H}_1, \dots, \mathcal{H}_k$ as above. We first illustrate that teaching can be viewed as a *set cover* problem: each teaching example eliminates or “covers” some bad hypothesis. Then we use this idea to design an interactive teaching algorithm in which the teacher, by probing the learner’s predictions, incrementally constructs a teaching set that eventually eliminates *all* bad hypotheses.

One interesting use of our teaching algorithm is in shrinking a training set T : finding a subset $S \subset T$ that yields the same final classifier. This can be useful in situations where the computational complexity of training scales poorly (e.g. quadratically) with the number of training instances. In Section 9.3, we illustrate this in experiments with kernel machines and neural nets.

8.0.4 Related work

The literature on teaching can be organized along two main threads: whether the learner is required to be consistent with all teaching examples and whether the teacher has full knowledge of the learner [64].

These two requirements, namely *consistency* and *full knowledge* have been focal in earlier theoretical work on teaching. For example, the classic teaching dimension [25, 51], the recursive teaching dimension [65, 27] and the preference-based teaching dimension [24] all assume both consistency and full knowledge.

Recently, there has been growing interest in settings where these requirements are relaxed. For instance, the work in [35] relaxes consistency by allowing the learner to be an

empirical convex loss minimizer. Work in [63] studies a setting where the teacher targets multiple learners with unknown models. Additional work in [36] relaxes the full knowledge requirement by allowing the teacher to be agnostic to the learner’s hyper-parameters or hypothesis space [36].

Of particular relevance is recent work by [37], which assumes the teacher and the learner use different linear feature spaces. The teacher cannot fully observe the learner’s linear model but knows the learner’s algorithm and can employ active querying to learn the mapping between feature spaces.

Here, we assume the learner is consistent with teaching examples but we do not require knowledge of its concept class or learning algorithm. This setting offers a crisp characterization of teaching black-box learners.

The notion of *sample compression* was introduced by [34] and has been the subject of much further work [e.g., 23, 42]. It is centered on an intriguing question: for a given concept class \mathcal{H} , is it possible to design (1) a learning algorithm \mathcal{A} that operates on labeled samples of some fixed size k , and (2) a procedure that, given any labeled data set, chooses a subset of size k such that when \mathcal{A} is applied to this subset, it produces a classifier consistent with the full data set? A recent result of [42] showed that if \mathcal{H} has VC dimension d , then $k = d2^d$ is always achievable. Our results can be thought of as a form of *adaptive sample compression*, where the concept class \mathcal{H} is unknown and the learning algorithm \mathcal{A} is fixed in advance and also unknown.

Chapter 9

Interactive machine teaching

We start by demonstrating our negative result for the non-interactive teacher. In this case, our teacher is agnostic to the learner’s model.

9.1 Teaching without interaction

A simple teacher, human or machine must somehow come up with informative teaching examples for the learner. Intuitively, if the learner’s concept class \mathcal{H} is known in advance, the teacher only needs to pick influential or “boundary examples”. In our example of thresholds on the line, the teacher knows the threshold w^* and thus, can construct a teaching set that consists of just two data points. We will see shortly that there are cases in which the “boundary examples” are so many that they essentially constitute the entire instance space!

Suppose that we have k concept classes $\mathcal{H}_1, \dots, \mathcal{H}_k$, each of which consists of thresholds along individual coordinates: \mathcal{H}_i consists of all functions $h_{i,w} : \mathcal{X} \rightarrow \{0, 1\}$ of the form

$$h_{i,w}(x) = \begin{cases} 1 & \text{if } x_i > w; \\ 0 & \text{otherwise.} \end{cases}$$

where $w \in \mathbb{R}$. That is, the hypotheses in \mathcal{H}_i only use the i th coordinate of the data. Here, we will assume that our teacher knows only that the learners concept class is one of

$\mathcal{H}_1, \dots, \mathcal{H}_k$, but not which one.

The instance space is a finite set $\mathcal{X} \subset \mathbb{R}^k$ specified as follows. Every point in \mathcal{X} has either *all positive coordinates* or *all negative coordinates*. The target concept h^* is 1 if the coordinates are all positive and 0 if all negative. Thus h^* lies in every \mathcal{H}_i : in particular, $h^* = h_{i,0}$ for all i . Set $\mathcal{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(k)}, -x^{(1)}, \dots, -x^{(k)}\}$, where the $x^{(i)} \in \mathbb{R}_+^k$ are defined as follows:

- The values of the k features of $x^{(i)}$ are $2, 3, 4, \dots, k$, in that order, with a 1 inserted in the i th position.
- Thus $x^{(1)} = (1, 2, 3, \dots, k)$, $x^{(2)} = (2, 1, 3, \dots, k)$, $x^{(3)} = (2, 3, 1, \dots, k)$, and $x^{(k)} = (2, 3, 4, \dots, k, 1)$.

Along any coordinate i , the correct threshold is 0. How can a helpful teacher teach such a concept?

Following the intuition that we established at the beginning of this section, a helpful teacher can construct a teaching set that consists of just two “boundary examples” for each H_i . These are $-x^{(i)}, x^{(i)}$, whose i th coordinates have values $-1, 1$ respectively. In other words: for \mathcal{H}_i , the optimal teaching set consists of $-x^{(i)}$ and $x^{(i)}$.

However, the only teaching set that works for every \mathcal{H}_i simultaneously is all of \mathcal{X} . We summarize this in the following theorem.

Theorem 9.1. *In the construction above, the concept classes $\mathcal{H}_1, \dots, \mathcal{H}_k$ each have teaching set size 2. If a non-interactive teacher does not know which of these concept classes is being used by the learner, the smallest possible teaching set it can provide is all of \mathcal{X} , of size $2k$.*

Proof. Consider any teaching set that leaves out some point in \mathcal{X} , say $x^{(i)}$. Then, if the learner happens to have concept class \mathcal{H}_i , it can consistently set the threshold to be 1.5 along the i th coordinate, since the $k - 1$ positive instances it has seen all have i th coordinate ≥ 2 . Thus it will get $x^{(i)}$ wrong. \square

9.2 Teaching with interaction

In the previous section we saw that it is possible to construct hypothesis classes for which the identities of the “boundary examples” can change dramatically from one hypothesis class to the next. This caused problems for the non-interactive teacher who is agnostic to target. Next, we will see that an interactive teacher can teach a learner whose representation, concept class, and learning algorithm are unknown. Our formulation treats teaching as a set cover problem. We begin by giving the specifics to the set cover problem.

9.2.1 The online set cover problem

The *set cover* problem is defined as follows. Let $X = \{1, 2, \dots, N\}$ be a set of N elements and let S be a family of subsets of X , where $|S| = m$. A *cover* is a collection of sets such that their union is X . Each $s \in S$ has a non-negative cost c_s associated with it. The goal is to find a cover of minimum total cost. The set cover problem is a known NP-hard problem.

Luckily, there is an alternative *online* version of the set cover problem and an elegant algorithm that was given in [1] that finds a set cover within a factor $\log N \cdot \log m$ of optimal. Under this formulation elements from X appear one at a time. The family of subsets S is known in advance to the algorithm and in each round, the algorithm must cover the element that appears by picking some subset $s \in S$. The objective is then to minimize the total cost of the sets that are chosen. For completeness, we describe the details of this algorithm next.

The algorithm maintains a weight $w_s > 0$ for each subset $s \in S$. Initially, $w_s = \frac{1}{2m}$, for each $s \in S$. The weight of each element $j \in X$ is defined as $w_j = \sum_{s \in S_j} w_s$, where S_j is the collection of sets containing j . The algorithm starts with an empty set cover $\mathcal{C} = \emptyset$. Define C to be the set of elements covered by each $s \in \mathcal{C}$. (Initially $C = \emptyset$.) The following potential function is also used throughout the algorithm.

$$\Phi = \sum_{j \in \mathcal{C}} N^{2w_j}.$$

Now when an element j appears the algorithm will choose a set that covers it as follows:

If $w_j < 1$

1. Let k be the minimal integer for which $2^k \cdot w_j > 1$.
2. For each $s \in S_j$, $w_s \leftarrow 2^k \cdot w_s$.
3. Add to \mathcal{C} at most $4 \log n$ elements from S_j at so that Φ does not exceed its value before step 2.

It can be shown that at the end of the algorithm, the cover \mathcal{C} will contain $O(\log m \log N)$ elements. Here is the formal theorem statement.

Theorem 9.2. ([1]) *Let \mathcal{C}_{OPT} denote the optimal set cover of X , where $|X| = N$ and let S be a family of subsets of X , where $|S| = m$. At the end of the online set cover algorithm, $|\mathcal{C}|$ is $O(|\mathcal{C}_{OPT}| \log m \log N)$.*

9.2.2 Teaching as a set cover

How does teaching relate to set cover? We can think of each teaching example as one that eliminates some sub-optimal hypotheses in \mathcal{H} , and a teaching set is a collection of examples that eliminate, or “cover”, *all* sub-optimal hypotheses. By this view, optimal teaching is equivalent to minimum set cover. We will see shortly that the algorithm for online set cover of the previous section can be simulated for interactive teaching. We consider the following model in which the teacher and learner interact.

On each round,

- The teacher supplies one or more teaching examples $(x, y) \in \mathcal{X} \times \{0, 1\}$ to the learner.
- The learner gives the teacher a black-box classifier $h : \mathcal{X} \rightarrow \{0, 1\}$ that is consistent with all the teaching examples it has seen so far.

The idea here is that the teacher cannot look inside the black-box classifier, but can test it on examples to get a sense of where its mistakes lie. On each round, the teacher comes up with teaching examples that will help the learner improve on these mistakes. In other words, on each round, a chosen teaching batch will cover bad hypotheses that the learner may have. Next, we present an interactive teaching algorithm that is provably within a factor $\log |\mathcal{X}| \cdot \log |\mathcal{H}|$ of optimal, just like the algorithm for online set cover we described in the previous section!

9.2.3 An interactive protocol

The resulting learning algorithm is shown in Figure 9.1. It is a randomized procedure that begins by drawing values T_x , one for each $x \in \mathcal{X}$, from a suitable exponential distribution. Then the interaction loop begins. A key quantity computed by the algorithm, for any learner-supplied black-box classifier h , is the set of misclassified points,

$$\Delta(h) = \{x \in \mathcal{X} : h(x) \neq h^*(x)\}.$$

Roughly speaking, the points x that are most likely to be chosen as teaching examples are those that have been misclassified multiple times by the learner's models, and for which T_x happens to be small.

Theorem 9.3. *Let t be the size of an optimal teaching set for \mathcal{H} . Pick any $0 < \delta < 1$. With probability at least $1 - \delta$, the algorithm of Figure 9.1 halts after at most $t \log(2|\mathcal{X}|)$ iterations. The number of teaching examples it provides is in expectation at most*

$$(1 + t \lg(2|\mathcal{X}|)) \cdot \left(\ln |\mathcal{H}| + \ln \frac{1}{\delta} \right).$$

The algorithm of Figure 9.1 is efficient and yields a teaching set of size $O(t \cdot \log |\mathcal{X}|)$.

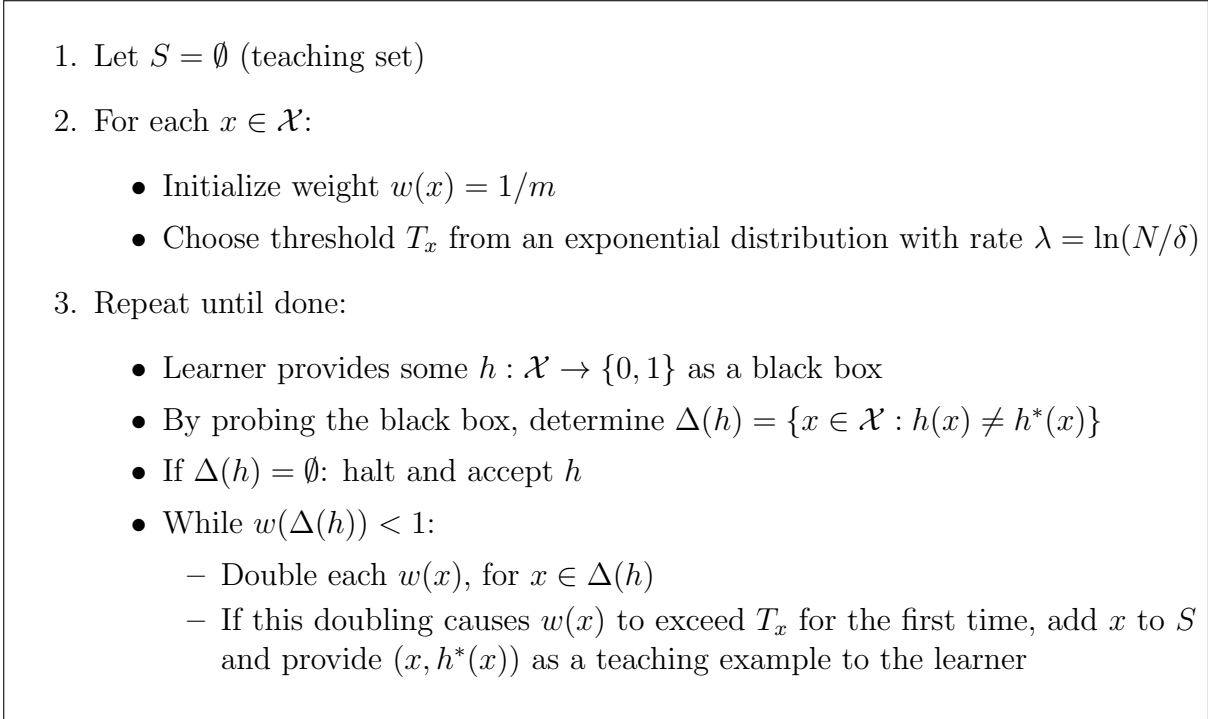


Figure 9.1. The teacher’s algorithm. Here $m = |\mathcal{X}|$ and $N = |\mathcal{H}|$. For $S \subset \mathcal{X}$, we define $w(S) = \sum_{x \in S} w(x)$.

$\log |\mathcal{H}|$), despite having no knowledge of the concept class \mathcal{H} . This can be significantly better than a teaching set of all $|\mathcal{X}|$ points, as we have seen would be needed by a non-interactive teacher.

9.3 Experimental illustration

In this section, we use Algorithm 9.1 to *shrink* several synthetic and real datasets, that is, to find subsets (teaching sets) of the data that yield the same final classifier. This can be useful for reducing storage/transmission costs of training data, or in situations where the computational complexity of training scales poorly with the number of samples.

Suppose the learning algorithm has running time $T(n)$, where n is the size of the training set. Algorithm 9.1 builds a teaching set incrementally, in iterations that involve adding a few points, invoking the learning algorithm, and evaluating the classifier that results. If the teaching set sizes along the way are $t_1 < t_2 < \dots < t_k$, the total training

time is $T(t_1) + \dots + T(t_k)$, which can be much smaller than $T(n)$.

Synthetic data We looked at synthetic data in the form of *moons*, *circles*, and *mixtures*. For each, we generated two-dimensional *separable* and *non-separable* datasets of 4000 points each, by varying the level of noise. We then tested Algorithm 9.1 using SVM learners with linear, quadratic, and RBF kernels. For each simulation we report: (1) the support vectors (SVs) of each learner; (2) the teaching points (TPs), as decided by the algorithm; (3) the points that are both support vectors and teaching points (TPs AND SVs); and (4) teaching curves.

For a support vector machine, it is always possible to create a teaching set of size two by choosing the points so that their perpendicular bisector is the boundary; the maximum-margin objective function will then yield exactly the target classifier. However, any given data set is unlikely to contain such a pair of points. Thus in our examples, the size of the optimal teaching set is not known, although it is certainly upper-bounded by the number of support vectors.

Some of the results are shown in Figure 9.2. For instance, the top left-hand panel shows the result of the teaching algorithm on the moon-shaped data. There are 123 support vectors in the full data set, but a teaching set of just 19 points is found. As can be seen on the right, these points are picked in five batches: the first batch has two points and already brings the accuracy above 75%. Overall, the learning algorithm is called five times, on data sets of size 2, 10, 13, 17, 19; and we get the same effect as calling it on the entire set of 4000 points.

The full range of experiments on synthetic data can be seen in Figures C.1 to C.17 in Appendix C.

Real datasets We also looked at the MNIST and fashion MNIST [60] datasets, both with 60K points.

1. On MNIST, we used an SVM with a quadratic kernel. This data has 32,320 support vectors, and a teaching set of 4,445 points is found (almost all support vectors).
2. On fashion MNIST, we used a convolutional network with 4 different layers of 2d convolutions (32, 64, 128, 128) each followed by a ReLU and a max pooling layer.

The bottom panel of Figure 9.2 shows the teaching curves for these two data sets. In either case, the accuracy achieved on the full training set is below 100%.

For all experiments we used the same termination criterion: the algorithm terminated when it got within .01 of the accuracy of the learner that was trained using the full data. Also, to initialize the weight T_x of each data point we set the confidence parameter δ of Algorithm 9.1 to .1.

Chapter 9 contains material as it appears in “Teaching a black-box learner.” International Conference on Machine Learning 2019. S. Dasgupta, D. Hsu, S. Poulis, X. Zhu. The dissertation author was the primary investigator.

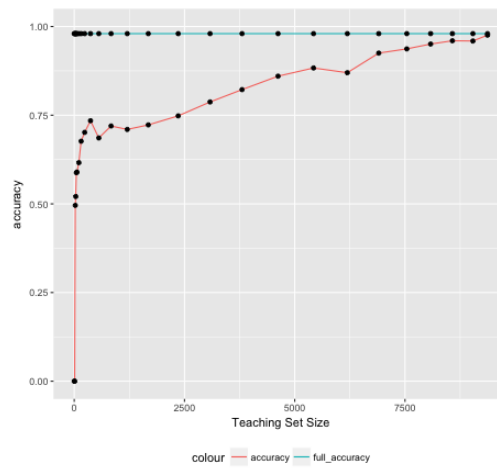
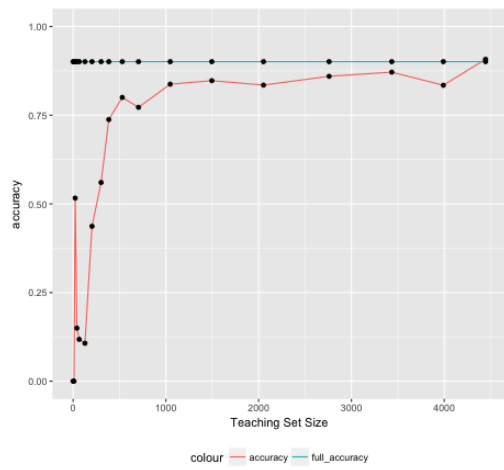
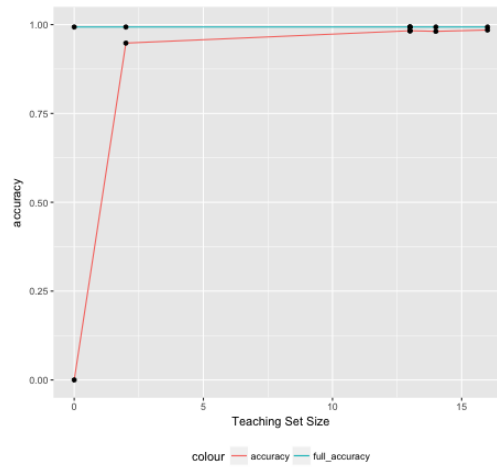
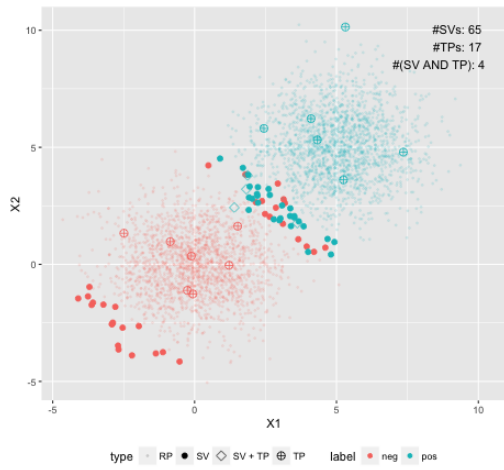
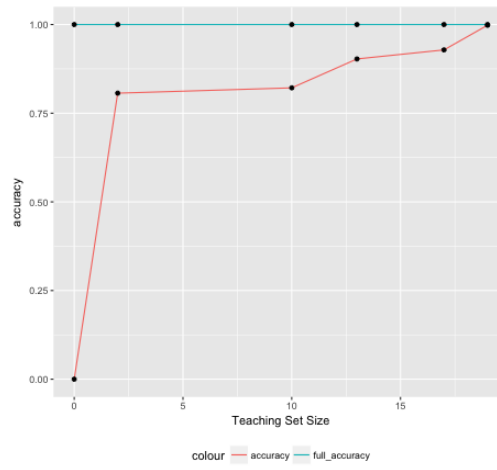
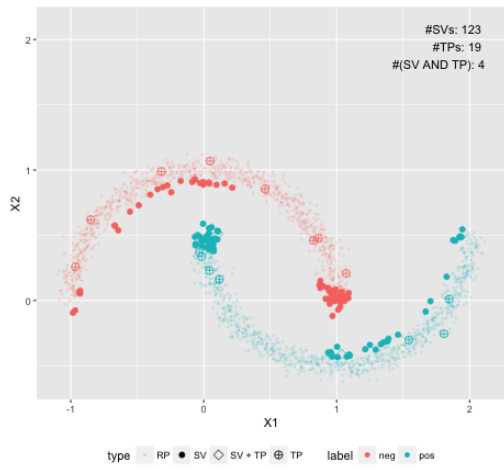


Figure 9.2. Top: ‘Moon’ data with RBF kernel SVM; Middle: ‘Mixtures’ data with quadratic kernel; Bottom: MNIST (quadratic SVM) and Fashion MNIST (CNN).

Appendix A

Supplementary material for Part I

A.1 Theoretical results for the probabilistic disjunction model

A.1.1 Proof of Theorem 3.2

Proof. The problem is clearly in NP. To show hardness, we will use a reduction from 3SAT.

Given a 3SAT instance $\phi(x_1, \dots, x_q) = C_1 \wedge C_2 \wedge \dots \wedge C_p$, where each clause C_j is a disjunction of three literals, create the following topic labeling problem:

- There are $2q$ topics: $t_1, \dots, t_q, t'_1, \dots, t'_q$. Think of t_i as corresponding to the positive literal x_i and t'_i the negative literal \bar{x}_i .
- For each variable x_i , create a document d_i whose topic distribution $\theta^{(d_i)}$ has probability $1/2$ on t_i and on t'_i and zero elsewhere.
- For each clause C_j , create a document d'_j whose topic distribution puts $1/3$ probability on (the t_i or t'_i corresponding to) each of the literals in C_j .
- The data set consists of document-label pairs $(d_i, 0), (d_i, 1), (d'_j, 1)$: a total of $p + 2q$ labeled documents.

Now, suppose there is an assignment $\ell : \{t_1, \dots, t_q, t'_1, \dots, t'_q\} \rightarrow \{0, 1, ?\}$ with

nonzero likelihood. Then for each labeled document (d, y) there is at least one topic t such that $\theta_t^{(d)} > 0$ and $\ell(t) = y$. Now, document d_i appears with label 0 as well as with label 1. Therefore, one of $\ell(t_i), \ell(t'_i)$ must be 0 and one of them must be 1. If $\ell(t_i) = 0, \ell(t'_i) = 1$, we will assign $x_i = 0$. If $\ell(t_i) = 1, \ell(t'_i) = 0$, we will assign $x_i = 1$. To see that this is a satisfying assignment, pick any clause C_j . The corresponding document d'_j has label 1; therefore at least one of the three topics corresponding to its literals must be assigned label 1 under $\ell(\cdot)$. Hence that literal is assigned a value of 1.

Conversely, if ϕ is satisfiable, then the mapping

$$\ell(t_i) = 0, \ell(t'_i) = 1 \text{ if } x_i = 0$$

$$\ell(t_i) = 1, \ell(t'_i) = 0 \text{ if } x_i = 1$$

has nonzero likelihood. □

A.1.2 Proof of Theorem 3.3

Proof. First, fix any t, y with $\ell(t) \neq y$. Under Assumption 3.1, each time topic t is selected, there is less than a $\lambda/2$ probability that the label is y . Conditioned on n_t , the expected value of n_{ty} is therefore at most $\lambda n_t/2$, and by a multiplicative Chernoff bound,

$$\Pr(n_{ty} \geq \lambda n_t) \leq e^{-n_t \lambda/6},$$

which is $\leq \delta/(Tk)$ if $n_t \geq n_o$.

Likewise, for any predictive feature $t \in P$, the expected value of $n_{t, \ell(t)}$ is at least $2\lambda n_t$. Again using a multiplicative Chernoff bound,

$$\Pr(n_{t, \ell(t)} < \lambda n_t) \leq e^{-n_t \lambda/6}.$$

Taking a union bound over all pairs $(t, y) \in [T] \times [k]$, we conclude that with

probability at least $1 - \delta$, the following holds whenever $n_t \geq n_o$:

- If $y \neq \ell(t)$ then $n_{ty} < \lambda n_t$.
- If $t \in P$ then $n_{t,\ell(t)} \geq \lambda n_t$.

Therefore, $\widehat{\ell}(t) = \ell(t)$ for $t \in P$ and ? otherwise. □

A.1.3 Proof of Theorem 3.4

Proof. Pick any predictive topic $t \in P$, and let $y = \ell(t)$. For a document x chosen at random,

$$\begin{aligned} \Pr_x(\text{topic } t \text{ selected}) &\geq \Pr_x(\text{document label} = y) \Pr_x(\text{topic } t \text{ selected} \mid \text{document label} = y) \\ &\geq \mathbb{E}_x \left[\frac{\sum_{t': \ell(t')=y} \theta_{t'}(x)}{\sum_{t' \in P} \theta_{t'}(x)} \cdot c_o \frac{\theta_t(x)}{\sum_{t': \ell(t')=y} \theta_{t'}(x)} \right] = c_o \gamma_t. \end{aligned}$$

Therefore, the expected number of documents that need to be seen before n_t reaches n_o is at most $n_o / (c_o \gamma_t)$. □

A.2 Incorporating feature feedback through regularization

A.2.1 Proof of Theorem 3.6

Recall that we wish to bound $R_n(\mathcal{F})$. The powerful results of [29] achieve this for a wide range of cases: for any $\mathcal{F} = \{w : \|w\| \leq W\}$, where $\|\cdot\|$ satisfies a strong convexity property. Specifically, they show

$$R_n(\mathcal{F}) \leq W \cdot \max_{x \in \mathcal{X}} \|x\|_* \cdot \sqrt{\frac{2}{n}}$$

where \mathcal{X} is the input space, and $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

We now apply this bound to our setting, where our regularizer norm is $\|\cdot\|_A$ for positive definite A .

Lemma A.1. *Pick any positive definite $p \times p$ matrix A and consider the Mahalanobis norm $\|\cdot\|_A$ on \mathbb{R}^p .*

1. *The function $\|\cdot\|_A^2$ is 2-strongly convex. In particular, for any $u, v \in \mathbb{R}^p$ and $0 \leq \alpha \leq 1$,*

$$\alpha\|u\|_A^2 + (1 - \alpha)\|v\|_A^2 - \|\alpha u + (1 - \alpha)v\|_A^2 = \alpha(1 - \alpha)\|u - v\|_A^2.$$

2. *The dual norm of $\|\cdot\|_A$ is $\|\cdot\|_{A^{-1}}$.*

Proof. The first assertion follows directly by expanding the expression. For the second, we note that the dual norm of $\|\cdot\|_A$ is defined by

$$\|x\|_* = \sup_{\|y\|_A \leq 1} x \cdot y.$$

We will show that this is $\|x\|_{A^{-1}}$.

First, take

$$y = \frac{A^{-1}x}{\sqrt{x^T A^{-1}x}}.$$

Then

$$\|y\|_A^2 = y^T A y = \frac{x^T A^{-1} A A^{-1} x}{x^T A^{-1} x} = 1$$

so $\|y\|_A = 1$. Moreover, $x \cdot y = \sqrt{x^T A^{-1} x} = \|x\|_{A^{-1}}$.

Conversely, pick any y with $\|y\|_A \leq 1$. Then

$$x \cdot y = x^T A^{-1/2} A^{1/2} y = (A^{-1/2} x)^T (A^{1/2} y) \leq \|A^{-1/2} x\|_2 \|A^{1/2} y\|_2 = \|x\|_{A^{-1}} \|y\|_A \leq \|x\|_{A^{-1}}.$$

□

If w^* is the sparse target classifier, the function class of interest is $\mathcal{F} = \{w : \|w\|_A \leq \|w^*\|_A\}$ and by [29] we have

$$R_n(\mathcal{F}) \leq \|w^*\|_A \cdot \max_{x \in \mathcal{X}} \|x\|_{A^{-1}} \sqrt{\frac{2}{n}}$$

Let $R = \{i \in [p] : w_i^* \neq 0\}$ denote the relevant features. We can split any x into its relevant and other components, $x = (x_R, x_o)$, and when we downweight the diagonal R -entries of A by a factor of c , we get

$$\|x\|_{A^{-1}}^2 = \|x_o\|_2^2 + c\|x_R\|_2^2$$

whereas

$$\|w^*\|_A^2 = \frac{1}{c}\|w\|_2^2$$

(assuming we have captured all the features on which w^* is non-zero). Thus

$$R_n(\mathcal{F}) \leq \|w^*\|_2 \cdot \max_{x \in \mathcal{X}} \sqrt{\left(\frac{1}{c}\|x_o\|_2^2 + \|x_R\|_2^2\right)} \sqrt{\frac{2}{n}}.$$

A.3 Proof of Lemma 4.1

Proof. Consider the optimization problem for computing the support vector classifier using the Mahalanobis regularizer.

$$\begin{aligned} & \underset{w}{\text{minimize}} && \frac{1}{2}\|w\|_A^2 + C \sum_{i=1}^N \xi_i \\ & \text{subject to} && \xi_i \geq 0, \quad y_i(x_i^T w + b) \geq 1 - \xi_i, \quad \forall i. \end{aligned} \tag{A.1}$$

The Lagrangian of (A.1) is

$$L(w, b, \xi, \mu, \alpha) = \frac{1}{2} \|w\|_A^2 + C \sum_{i=1}^N \xi_i - \sum_i^N \mu_i \xi_i - \sum_i^N \alpha_i [y_i (x_i^T w + b) - (1 - \xi_i)],$$

where the α_i, μ_i are the Lagrange multipliers. It easy to see that the Lagrange dual function L_D is

$$L_D(\mu, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T A^{-1} x_j.$$

which corresponds to the ℓ_2 -regularized SVM with data $(A^{-1/2}x_i, y_i)$. □

A.4 Experiments

A.4.1 Data sets

20 NewsGroups: The 20-Newsgroups collection is a set of approximately 20,000 newsgroup documents, partitioned evenly across the 20 different newsgroups. The documents are postings about politics, sports, technology, religion, science etc., and contain subject lines, signature files, and quoted portions of other articles. Some of the newsgroups are very closely related to each other (e.g., IBM computer system hardware *vs* Macintosh computer system hardware), while others are unrelated (e.g., misc for sale *vs* social religion and christian). A processed version of the data set was obtained. The original data set can be found on Jason Rennie’s website. ¹.

Reuters-21578: This is another widely used collection for text categorization research. The documents appeared on the Reuters newswire in 1987 and were manually classified into several topics by personnel from Reuters Ltd. See [31] for further details on the data set. Sub-collections **R10** (10 classes with the highest number of topics) and **R90** (at least one positive and one training example) are usually considered for text categorization tasks. As our goal here was to consider single-labeled data, all the documents with less than or with more than one label were eliminated, resulting in **R8** (8 classes) and **R52** (52 classes).

webkb: This data set contains web pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base project of the CMU text learning group ².

cade: The documents in this collection correspond to web pages extracted from the CADE Web Directory, which points to Brazilian web pages classified by human experts in 12 classes, including services, education, sciences, sports, culture etc.

¹<http://qwone.com/~jason/20Newsgroups/>

²<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-4/text-learning/www/index.html>

ohsumed: This data set includes medical abstracts from the MeSH (Medical Subject Headings) categories of the year 1991 ³ on 23 cardiovascular disease categories. We only considered documents with a single label.

For each data set we only considered tokens that occurred at least 3 times. Figure A.1 below provides a summary of the data as they were used in the experiment.

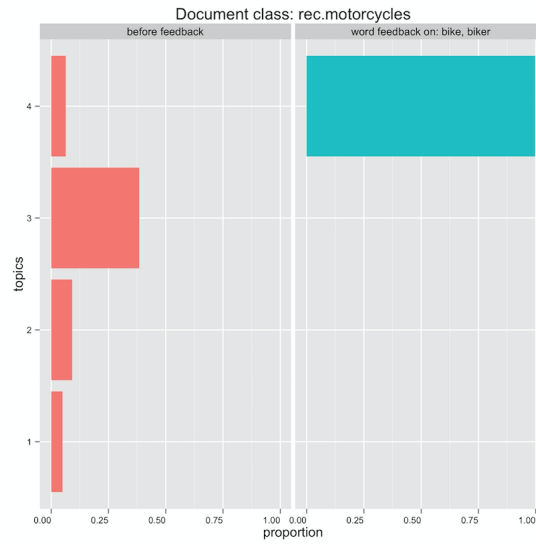
	# tokens	# training docs	# test docs	# topics	# classes
20 NewsGroups (20ng)	33,223	11,293	7,528	200	20
Reuters 8 (R8)	7,744	5,485	2,189	80	8
Reuters 52 (R52)	8,868	6,532	2,568	520	52
cade	68,983	27,322	13,661	120	12
webkb	7,644	2,803	1,396	40	4
ohsumed	13,627	3,357	4,043	230	23

Figure A.1. Summary of the datasets and the number of topics used in the experiment

³<ftp://medir.ohsu.edu/pub/ohsumed>

A.4.2 Results

An example of a PDM from the **20ng** dataset is shown in figure A.2. Figures A.3 - A.8 show our experimental results for each one of the data sets in more detail. Figures A.9- A.10, show the *amount* of feedback over time.



	Topic 1	Topic 2	Topic 3	Topic 4
1	gener	air	unit	bike
2	process	heat	engin	dod
3	thi	temperatur	cross	ride
4	sinc	water	bnr	motorcycl
5	effect	cold	adjust	bmw
6	anoth	pressur	link	rider
7	requir	hot	pre	helmet
8	real	fan	replac	sun
9	result	effect	nick	drink
10	case	ga	put	biker

Figure A.2. Top : Topic representation of a document with the class **rec.motorcycles** before and after feature feedback on **bike** and **biker**. Bottom: Descriptive words of the topics that are present in the document.

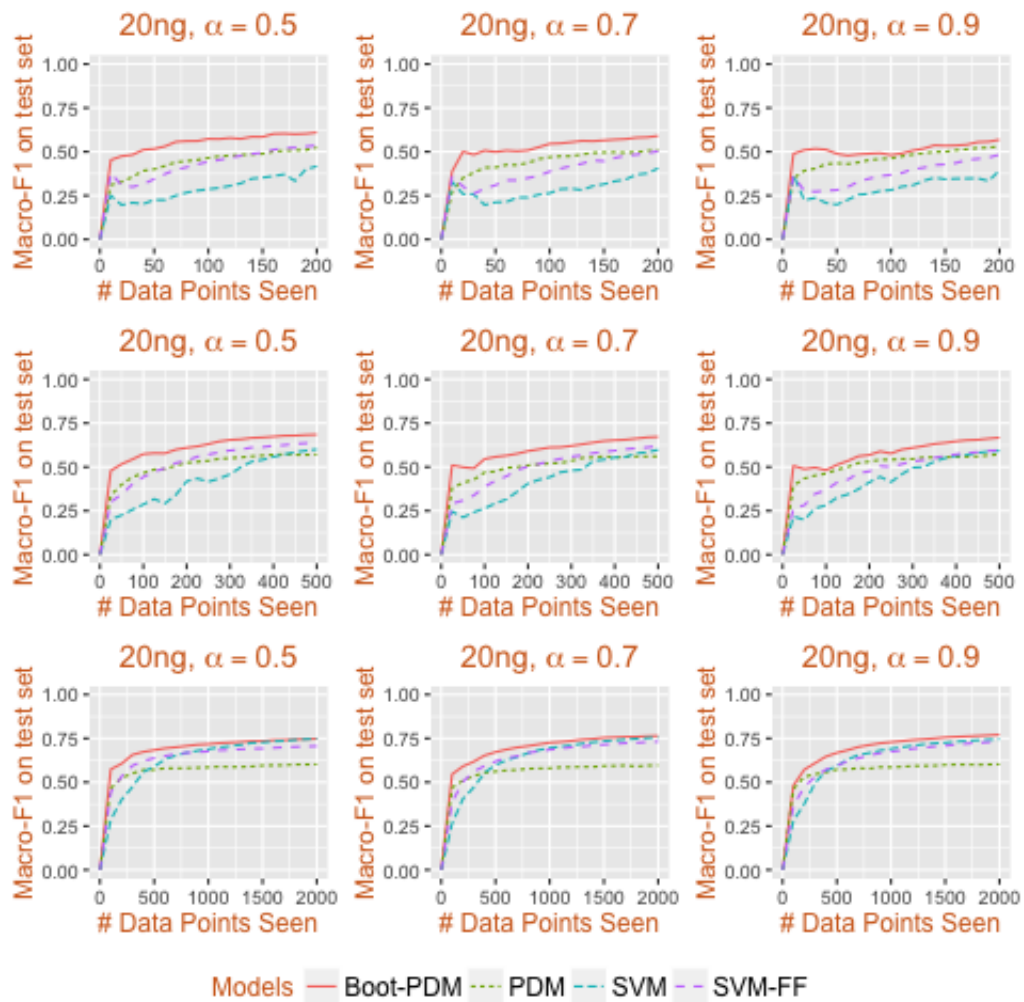


Figure A.3. 20ng

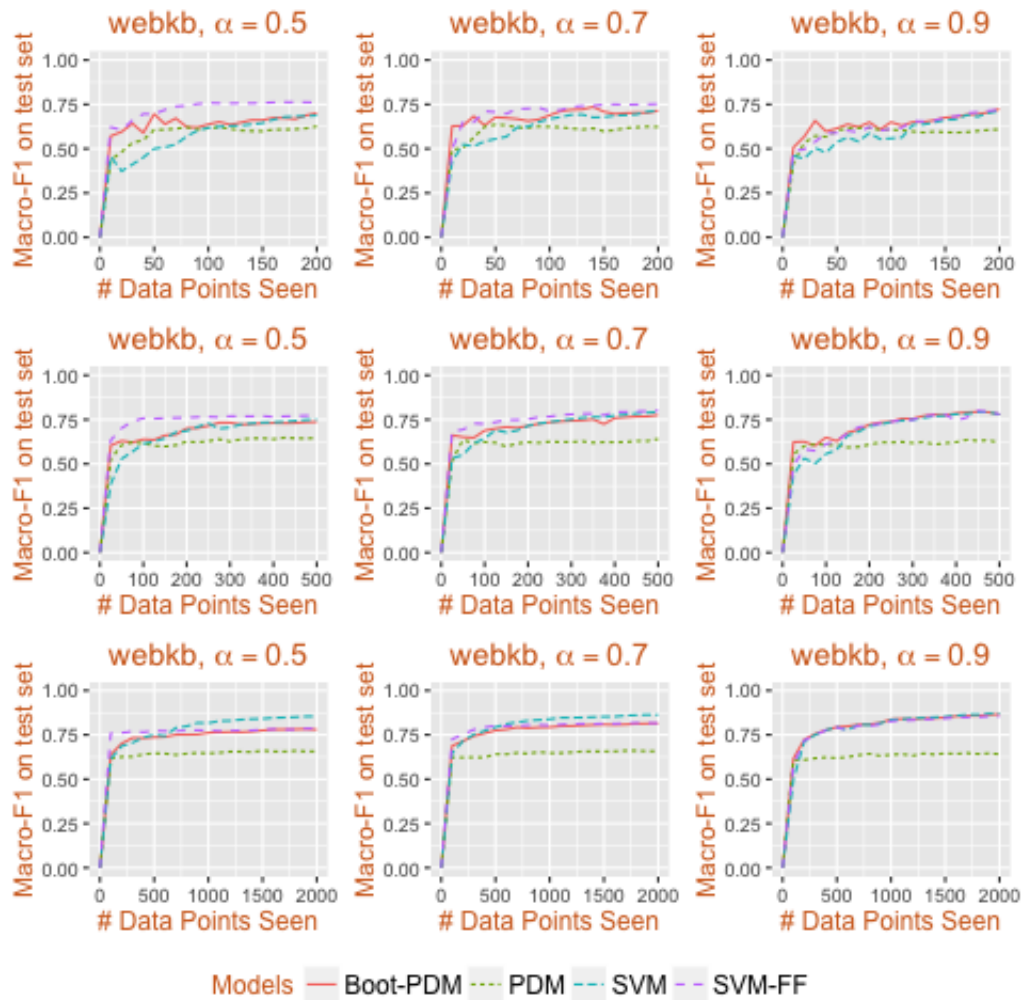


Figure A.4. webkb

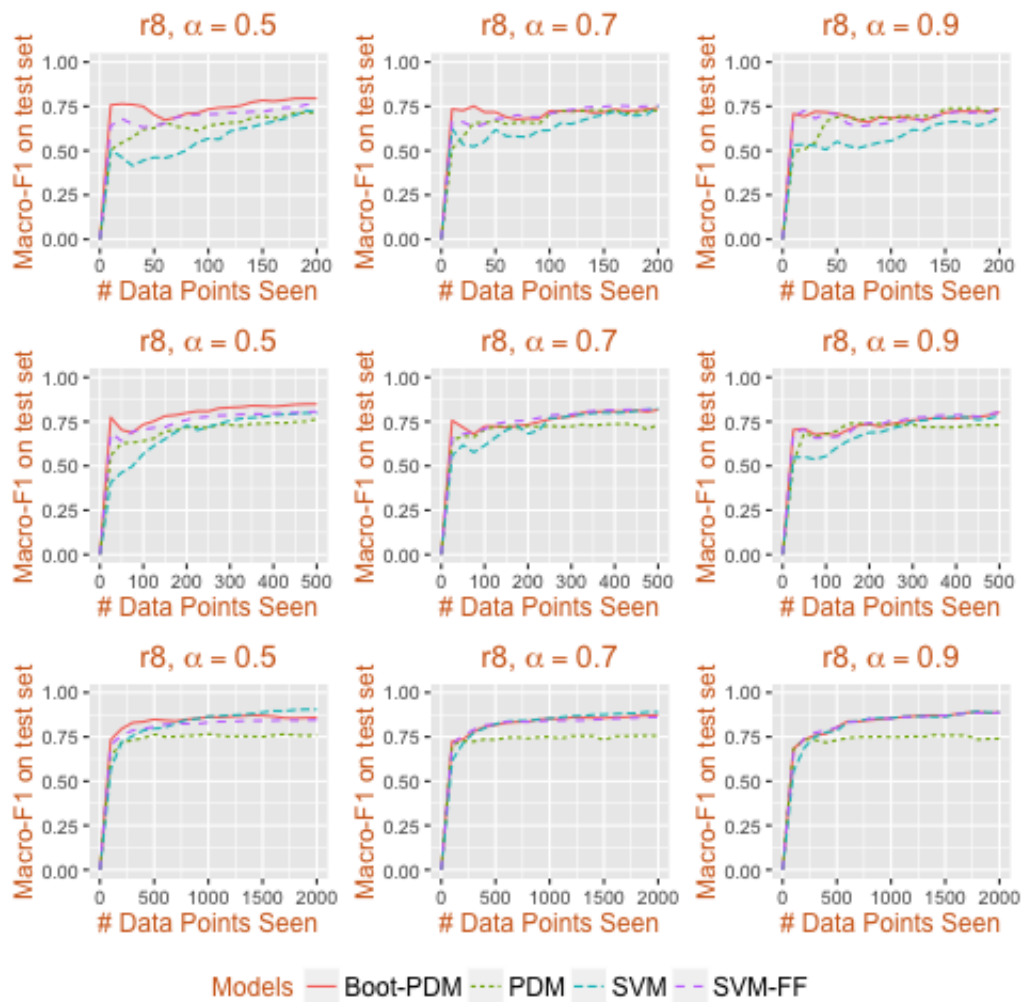


Figure A.5. R8

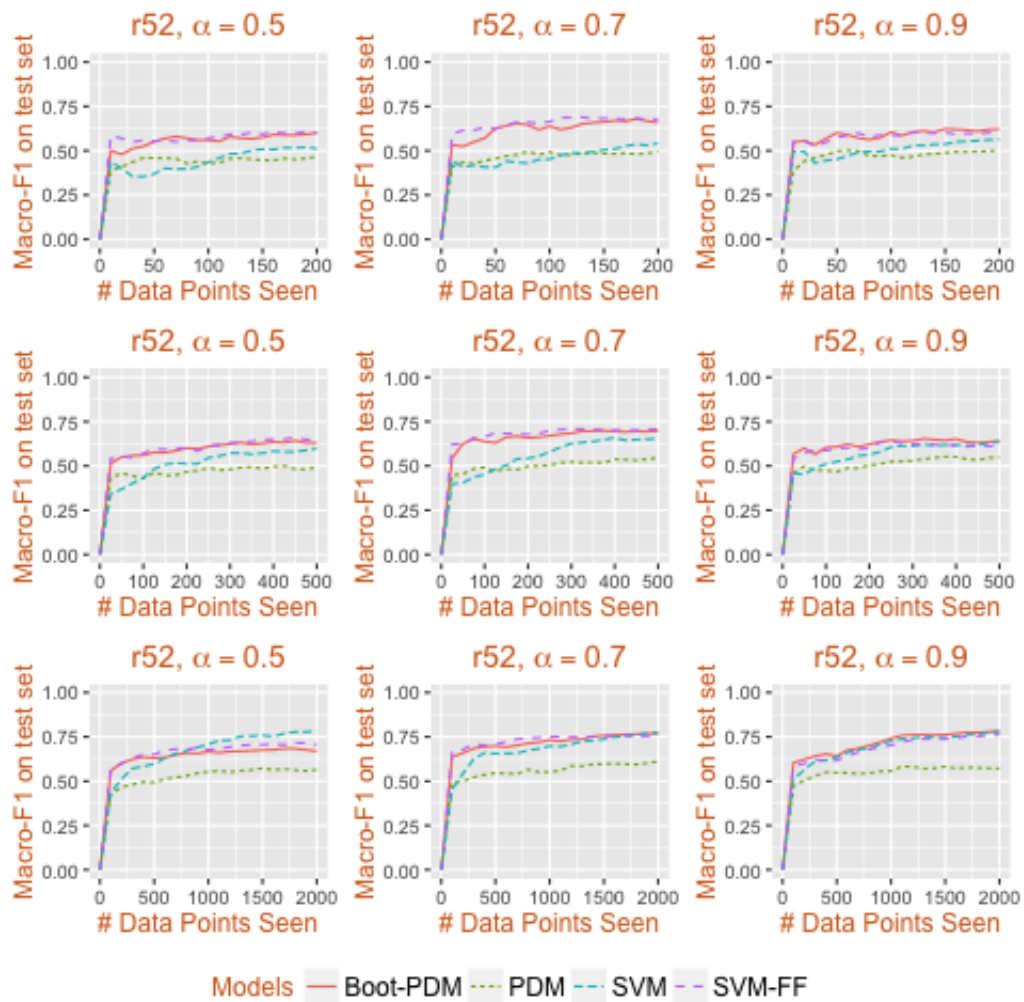


Figure A.6. R52

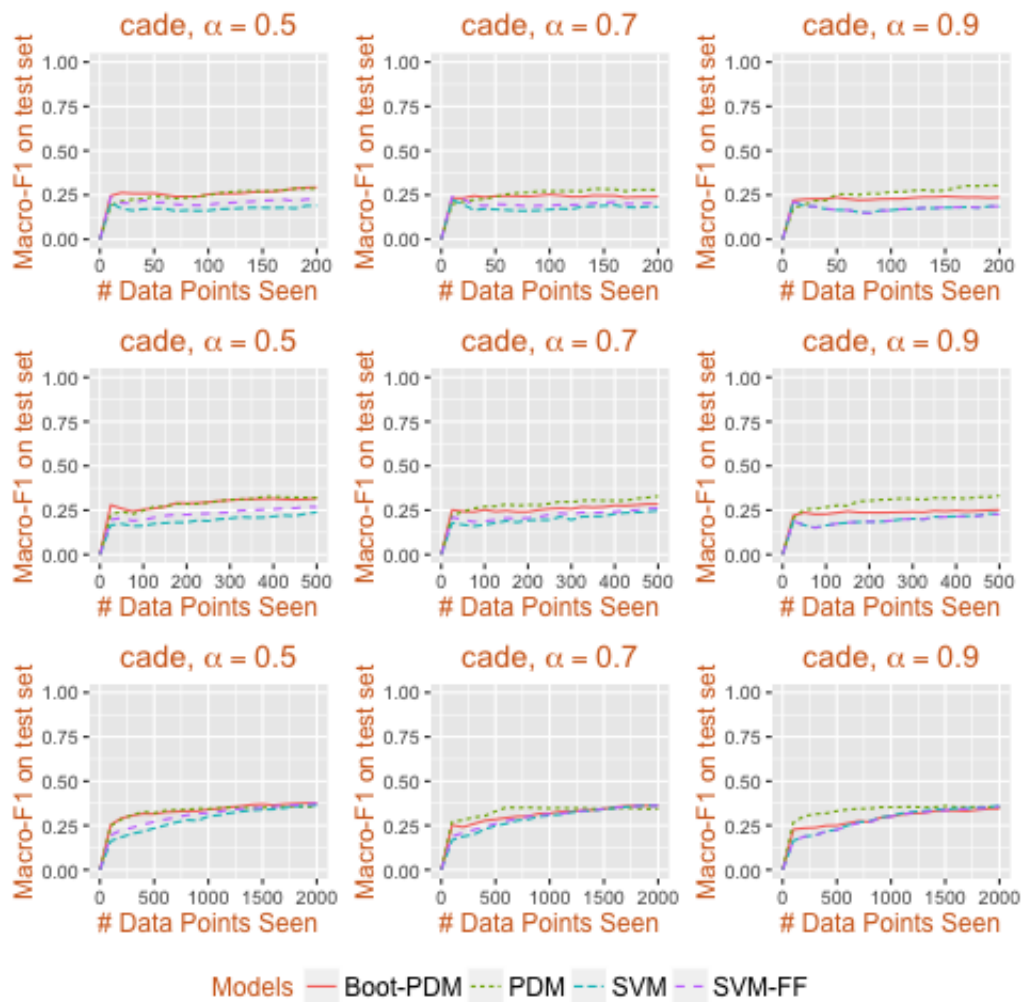


Figure A.7. Cade

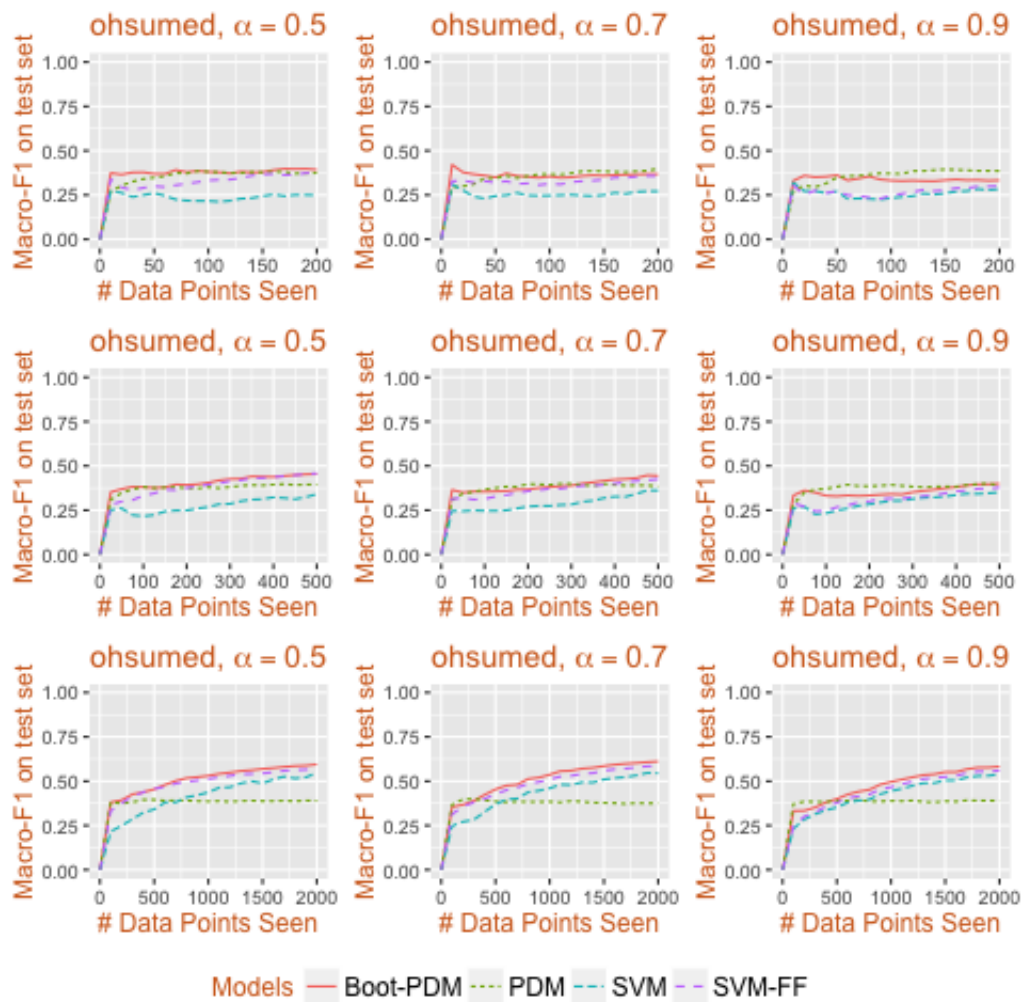


Figure A.8. Ohsumed

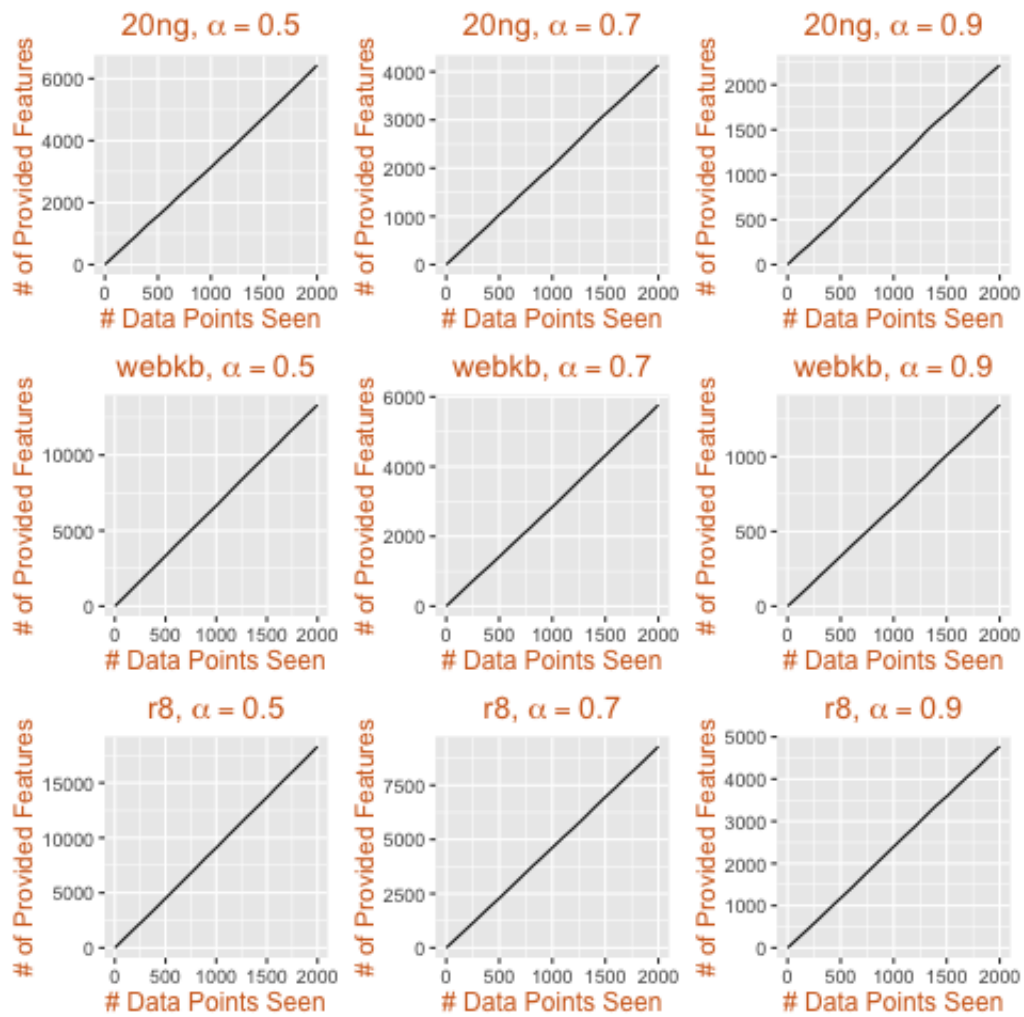


Figure A.9. Amount of Feature Feedback

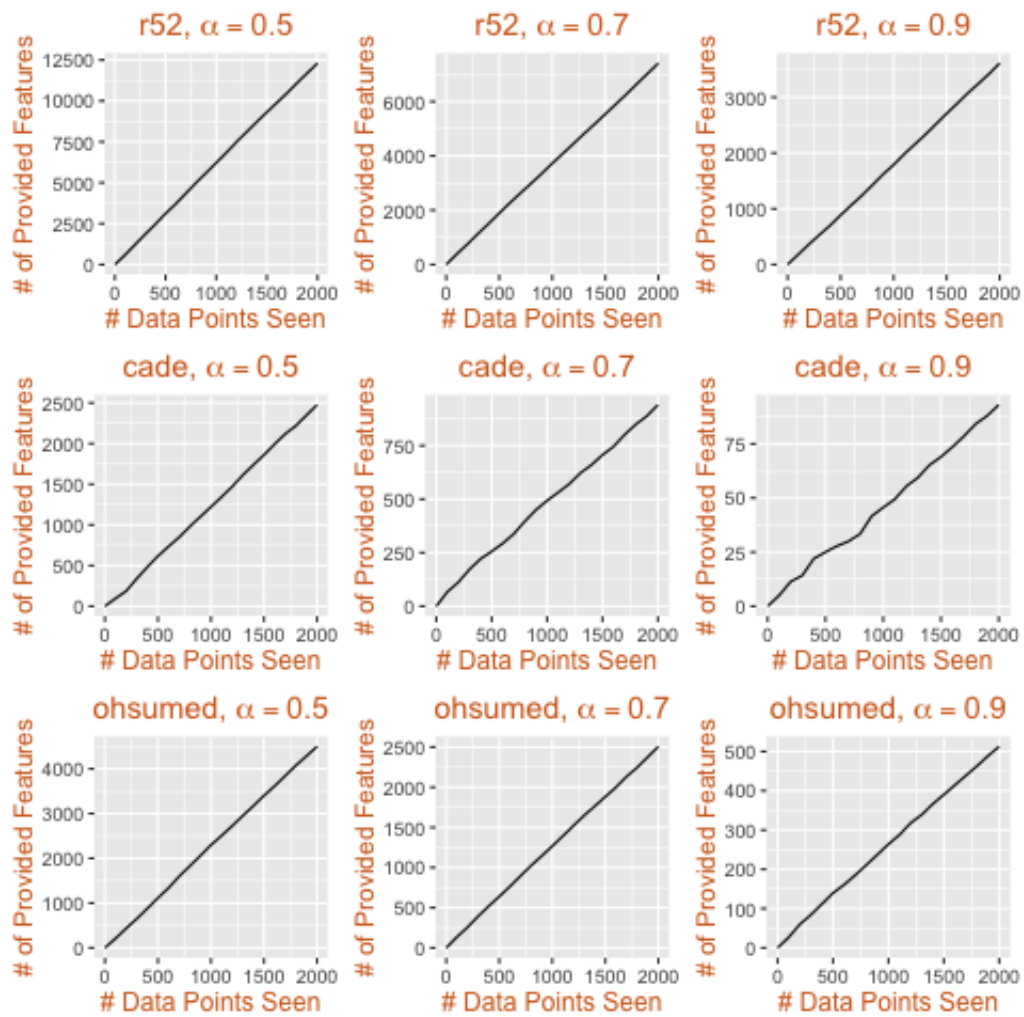


Figure A.10. Amount of Feature Feedback

A.4.3 Human Experiment.

Figure A.11 depicts the interface that was used to solicit labels and feature feedback from human annotators. Annotators were given the option to select a number of features from a list. They were also given the ability to insert a feature from the document that was not in the list.

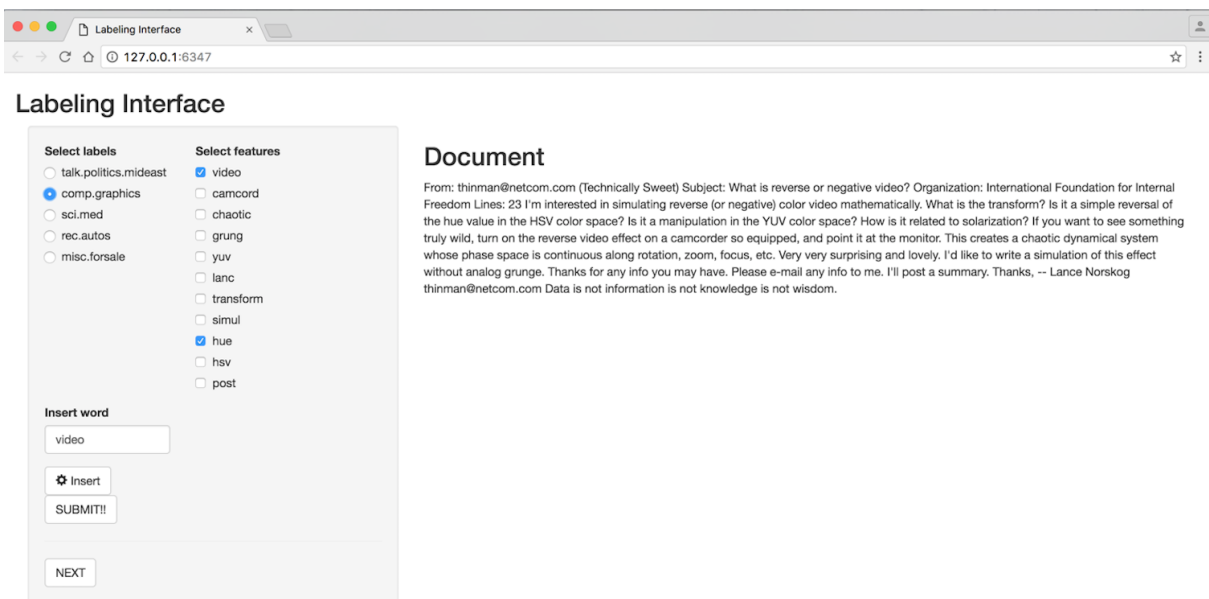


Figure A.11. Interface used in Human Experiment

Appendix B

Supplementary material for Part II

Proof of Lemma 6.1

Proof. Recall from LDA that the posterior probability of a topic vector z given a corpus x can be written as

$$\begin{aligned}
 Pr(z | x) &\propto \prod_{t=1}^K \frac{\left(\prod_w \Gamma \left(n_t^{(w)}(z) + \beta \right) \right) \left(\prod_d \Gamma \left(n_t^{(d)}(z) + \alpha \right) \right)}{\Gamma \left(n_k(z) + W\beta \right)} \\
 &= \frac{\overbrace{\left(\prod_{t=1}^K \prod_w \Gamma \left(n_t^{(w)}(z) + \beta \right) \right)}^{r_w(z)} \overbrace{\left(\prod_{t=1}^K \prod_d \Gamma \left(n_t^{(d)}(z) + \alpha \right) \right)}^{r_d(z)}}{\underbrace{\prod_{t=1}^K \Gamma \left(n_t(z) + W\beta \right)}_{r(z)}}
 \end{aligned}$$

Suppose $s \in \text{CC}(\mathcal{C})$. Say $k \in \{1, \dots, K\}$ and z satisfy that for all $s' \in N(s)$, $z_{s'} \neq k$. And let \hat{z} be the topic vector that satisfies $\hat{z}_s = k$ and $\hat{z}_{-s} = z_{-s}$. Then by the posterior probability of LDA, we have

$$\begin{aligned}
 Pr(z_s = k | z_{-s}, x, \mathcal{C}) &\propto Pr(z_s = k, z_{-s} | x) \\
 &= Pr(\hat{z} | x) \\
 &= \frac{r_w(\hat{z})r_d(\hat{z})}{r(\hat{z})}.
 \end{aligned}$$

We can work out each of the above terms separately.

$$\begin{aligned}
r_w(\widehat{z}) &= \prod_{t=1}^K \prod_w \Gamma \left(n_t^{(w)}(\widehat{z}) + \beta \right) \\
&= \left(\prod_{w \in s} \prod_{t=1}^K \Gamma \left(n_t^{(w)}(\widehat{z}) + \beta \right) \right) \left(\prod_{w \notin s} \prod_{t=1}^K \Gamma \left(n_t^{(w)}(\widehat{z}) + \beta \right) \right) \\
&= \left(\prod_{w \in s} \frac{\Gamma \left(n_k^{(w)}(z_{-s}) + n^{(w)}(s) + \beta \right)}{\Gamma \left(n_k^{(w)}(z_{-s}) + \beta \right)} \prod_{t=1}^K \Gamma \left(n_t^{(w)}(z_{-s}) + \beta \right) \right) \\
&\quad \left(\prod_{w \notin s} \prod_{t=1}^K \Gamma \left(n_t^{(w)}(z_{-s}) + \beta \right) \right) \\
&= \left(\prod_{w \in s} \frac{\Gamma \left(n_k^{(w)}(z_{-s}) + n^{(w)}(s) + \beta \right)}{\Gamma \left(n_k^{(w)}(z_{-s}) + \beta \right)} \right) \left(\prod_w \prod_{t=1}^K \Gamma \left(n_t^{(w)}(z_{-s}) + \beta \right) \right) \\
&\propto \prod_{w \in s} \frac{\Gamma \left(n_k^{(w)}(z_{-s}) + n^{(w)}(s) + \beta \right)}{\Gamma \left(n_k^{(w)}(z_{-s}) + \beta \right)}
\end{aligned}$$

We can do similar derivations for $r_d(\widehat{z})$.

$$\begin{aligned}
r_d(\widehat{z}) &= \prod_{t=1}^K \prod_d \Gamma \left(n_t^{(d)}(\widehat{z}) + \alpha \right) \\
&= \left(\prod_{d \in s} \prod_{t=1}^K \Gamma \left(n_t^{(d)}(\widehat{z}) + \alpha \right) \right) \left(\prod_{d \notin s} \prod_{t=1}^K \Gamma \left(n_t^{(d)}(\widehat{z}) + \alpha \right) \right) \\
&= \left(\prod_{d \in s} \frac{\Gamma \left(n_k^{(d)}(z_{-s}) + n^{(d)}(s) + \alpha \right)}{\Gamma \left(n_k^{(d)}(z_{-s}) + \alpha \right)} \prod_{t=1}^K \Gamma \left(n_t^{(d)}(z_{-s}) + \alpha \right) \right) \\
&\quad \left(\prod_{d \notin s} \prod_{t=1}^K \Gamma \left(n_t^{(d)}(z_{-s}) + \alpha \right) \right) \\
&= \left(\prod_{d \in s} \frac{\Gamma \left(n_k^{(d)}(z_{-s}) + n^{(d)}(s) + \alpha \right)}{\Gamma \left(n_k^{(d)}(z_{-s}) + \alpha \right)} \right) \left(\prod_d \prod_{t=1}^K \Gamma \left(n_t^{(d)}(z_{-s}) + \alpha \right) \right) \\
&\propto \prod_{d \in s} \frac{\Gamma \left(n_k^{(d)}(z_{-s}) + n^{(d)}(s) + \alpha \right)}{\Gamma \left(n_k^{(d)}(z_{-s}) + \alpha \right)}
\end{aligned}$$

Finally, we can do the same exact thing to $r(z)$.

$$\begin{aligned}
 r(\hat{z}) &= \prod_{t=1}^K \Gamma(n_t(\hat{z}) + W\beta) \\
 &= \Gamma(n_k(\hat{z}) + W\beta) \prod_{t \neq k} \Gamma(n_t(\hat{z}) + W\beta) \\
 &= \frac{\Gamma(n_k(z_{-s}) + n(s) + W\beta)}{\Gamma(n_k(z_{-s}) + W\beta)} \prod_{t=1}^K \Gamma(n_t(z_{-s}) + W\beta) \\
 &\propto \frac{\Gamma(n_k(z_{-s}) + n(s) + W\beta)}{\Gamma(n_k(z_{-s}) + W\beta)}
 \end{aligned}$$

Putting the above together, we get the lemma. □

Anchor words

debate isis black banks mexico chinese food google cnn cruz politics water college star debt convention officers flag army rock radio wages netflix joe students income russian season users
 judge parents space gun park kelly gold elect amazon airport products child vice driving older africa survey weapons hate steps names drug chinas letter michael app victims rules kids sports
 county model fast travel film book music lawsuit internet carolina healthy seat golf degree buildings putin notes immigration delegates scandal syria obamas career african intelligence rio
 heart dollar zika test tesla king fans ryan disney hearing female dream museum girl suspect grand box entertainment funds gas request poll bush student sexual johnson chicago sex terror super
 supreme olympics cities foundation hospital iran baby modern beijing historic asia beat hollywood india trial crash drivers terrorist warner page search iraq kasich flight retirement brazil
 uber played voting japan sources digital stores puerto awards doctors opposition martin feature eye nuclear syrian laws lines trust girls ford budget border journalists island fake documents
 mobile residents training microsoft loans movie opec screen gay award crude society earth french assault station benefits masa platform birth estate korea prison schools warren governor
 famous emergency paris yahoo virginia samsung smith character lawyers aid prosecutors victim womens cleveland obamacare ban hotel returns plane nbc cable climate hot olympic art gawker scott
 goldman images deals stars launch exchange muslim elections retail troops apps brain editor pence airlines brown mexican homes france yellen patients devices prince region ratings phones dow
 infrastructure venezuela blue orlando sea bond videos cuba deaths generation payments fuel hall sale indiana cancer brussels voice missing falluja buffett gender article design natural
 funding emails scene verizon blood viewers drone floor tour aleppo winner arizona practice manufacturing protests allies revealed researchers israel currency drugs feet card disease
 campaigns germany profit moments driver hike sites turkey commission coast strike savings song michelle smart construction teams williams original facts pennsylvania presidents cuts rep
 mayor square murder jones beach rubio software immigrants protesters cast systems club web harder debates shootings sat dog northern language canada housing camera false operation records
 tweet minimum vehicles passengers ads britain aircraft peace jersey damage speed trillion cup trip studies percent van block ballot dallas transition fees council round investigators cbs
 traffic paper boy steve gains land tariffs assets poor athletes scheduled conditions tweets michigan items foot agencies truth uncertainty fashion title ticket restaurant brands
 unemployment index controversy smaller century electric losses injured mission lewandowski suit quality property lee changing cards assistant startup transgender sound corruption owners
 discovered muslims communities green speaker adviser rule magazine ruling iraq treasury gap insurance investing active science fine camp bonds auto korean train referendum suicide france
 producers approved threats bell proposal citys zuckerberg scientists refugees fraud stake england remove harassment storm strategist wealth japanese overseas cooper virtual interviews
 brother ban weather values

Merge words

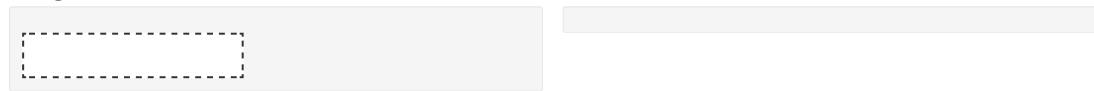


Figure B.1. The initial list of candidate anchor words that was presented to users. Users initialized topics that they wanted to create by dragging and dropping a candidate anchor in the dotted box labeled as ‘Merge words’.

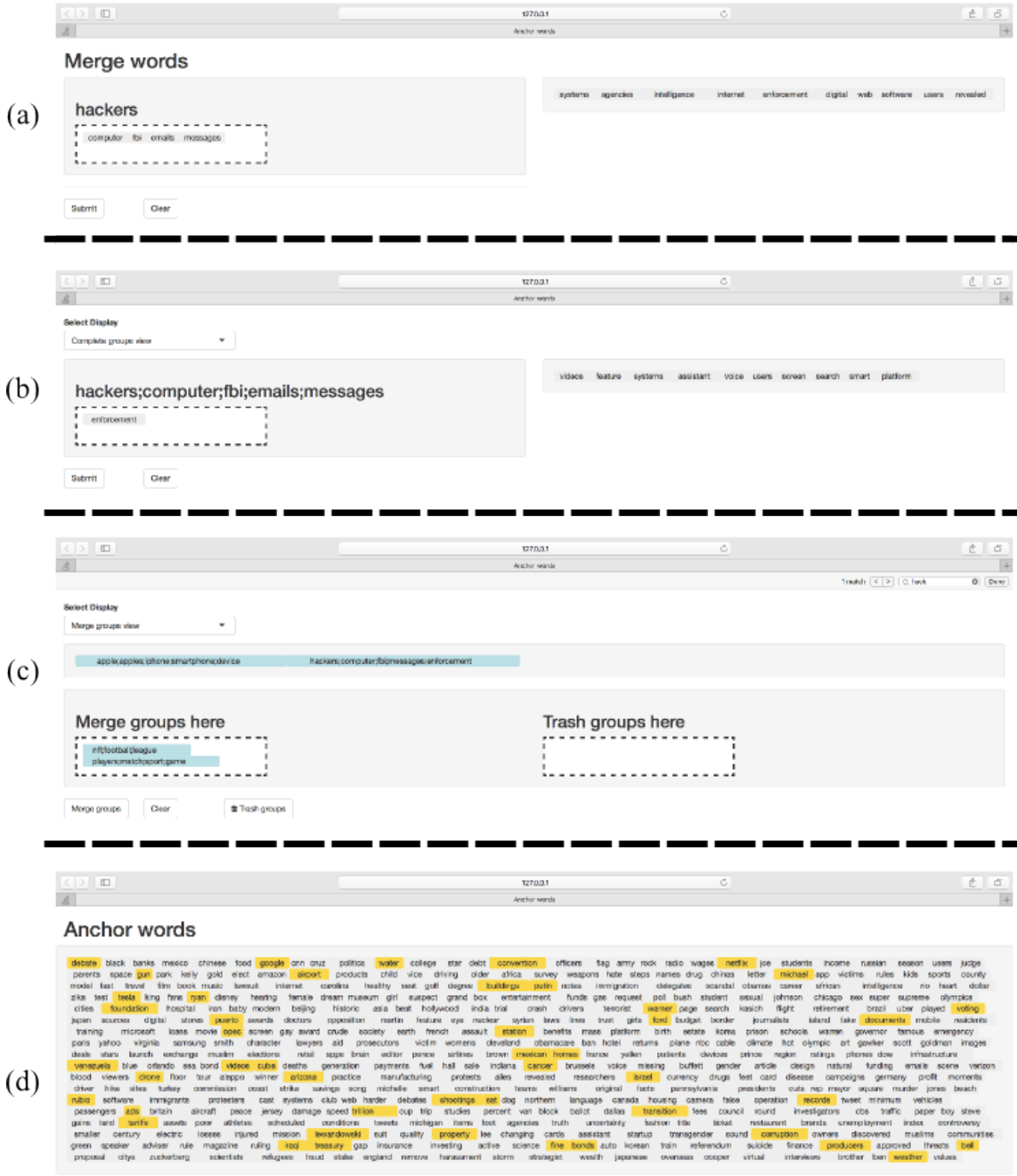


Figure B.2. (a) Merge anchor words view. (b) Complete groups view. (c) Merge and trash groups view. (d) Suggest anchor words view.

Table B.1. Topics created by user 1

Interactive	verizon yahoo warner internet company deal tech billion media mayer
Regular	tech company netflix million stock billion investors market published companies
Interactive	police black protests african government march protesters mass anti law
Regular	police officers dallas shot protest officer killed shooting man law
Interactive	rio gold olympic brazil olympics games athletes zika training team
Regular	zika health rio virus olympics games states house president government
Interactive	students college school schools kids student education university pay program
Regular	scenes show marshall left work back national published clinton media
Interactive	britain brexit england british pound european europe london france goal
Regular	winners golden million series back published globes company home film
Interactive	rate banks rates bank fed interest stocks jobs debt economy
Regular	fed trump rates economy rate market yellen jobs president growth
Interactive	fargo buffett wells million jobs stock bank company ceo clinton
Regular	wells fargo bank banks million accounts employees company sales stumpf
Interactive	uber drivers driving cars car vehicles driver ride traffic safety
Regular	uber company china million drivers companies tech published billion chinese
Interactive	trump nbc million donald clinton viewers campaign fox cbs president
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	plane aircraft obama flight passengers seat boeing fuel security airlines
Regular	korea korean china military government march nuclear report president company
Interactive	disney show company film movie twitter media star china box
Regular	comedy show ladies fey night live series film host awards
Interactive	tesla car cars electric auto vehicles musk driving teslas loans
Regular	tesla car musk company cars teslas published electric driving million
Interactive	workers rate unemployment jobs manufacturing wage job prices recession cuts
Regular	fed trump rates economy rate market yellen jobs president growth
Interactive	computer hackers system systems feature information screen technology security assistant
Regular	company windows tech microsoft published million work twitter police security
Interactive	debate million trumps debates vice viewers election elect night fox
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	trump lewandowski market trumps donald clinton campaign investors president tech
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	japan china japanese trade global countries international chinese worlds asia
Regular	company tech amazon sales market billion companies stock published walmart
Interactive	netflix oil tech china entertainment energy screen show shows original
Regular	tech company netflix million stock billion investors market published companies
Interactive	nfl players football games team game league season sports fans
Regular	winners golden million series back published globes company home film
Interactive	obamacare health care insurance plan benefits coverage federal pay exchanges
Regular	zika health rio virus olympics games states house president government
Interactive	nuclear korea korean weapons iran kim defense military ballistic foreign
Regular	korea korean china military government march nuclear report president company
Interactive1	israel form peace class israeli freewheel national rail sync foreign
Regular	sync rail freewheel trump class form clinton input banner div
Interactive	cnn viewers media cbs kelly journalists sources fox network coverage
Regular	trump dylan clinton media bob campaign president twitter show published
Interactive	ford cars car auto vehicles jobs mexico manufacturing driving trade
Regular	ford car cars mexico company sales million published president police
Interactive	space station international moments attacks launch president company home tech
Regular	space station international moments trump notable crew clinton russian campaign
Interactive	immigrants law immigration children trumps plan dream place living wall
Regular	clinton trump president campaign harry hillary donald sanders potter back
Interactive6	twitter tweet tweets web anti gawker fake tweeted hogan harassment
Regular	trump dylan clinton media bob campaign president twitter show published
Interactive	boy girl family mother parents child children baby girls unfolds
Regular	prince remembers purple princes music city police family home death
Interactive	police officers shot shooting shootings victims suspect dallas gun killed
Regular	police officers dallas shot protest officer killed shooting man law
Interactive	russian putin russia election court opposition political obama foreign states
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	venezuela economy government prices brexit production european country president crude
Regular	oil prices production million saudi market energy barrels billion company
Interactive1	falluja isis battle forces syria attacks attack syrian military government
Regular	syrian isis city forces aleppo march syria government refugees group
Interactive	trump kasich cruz rubio clinton republican sanders delegates freewheel campaign
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	google apple note devices iphone phones phone samsung software app
Regular	apple iphone company tech apples million sales published phone stock

Table B.2. Topics created by user 2

Interactive	france obama french attacks paris european germany attack brexit england
Regular	brussels attacks terror airport police attack march paris security isis
Interactive	trump clinton donald campaign journalists media president national magazine republican
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	puerto island sea obama house back debt coast states class
Regular	president obama memorial american national police clinton happening watched updated
Interactive	music song voice group freewheel records rail sync form class
Regular	trump dylan clinton media bob campaign president twitter show published
Interactive	amazon netflix internet tech company google disney stock movie investors
Regular	tech company netflix stock million investors billion market media published
Interactive	obamacare fargo benefits health insurance care plan wells federal coverage
Regular	wells fargo bank banks million accounts employees company stumpf sales
Interactive	uber drivers app driving car cars ride company million cities
Regular	tesla car musk company cars teslas electric published driving autopilot
Interactive	airport plane flight travel aircraft airlines security passengers international brussels
Regular	brussels attacks terror airport police attack march paris security isis
Interactive	study researchers science national studies natural million research school found
Regular	earth live star planet space system water show light back
Interactive	students college school schools kids student education university high pay
Regular	scenes show marshall left work back national published media film
Interactive	police trial judge prison court hearing officers attorney department charges
Regular	police officers dallas shot protest officer shooting killed man law
Interactive	food restaurant company chipotle sales million market restaurants fast customers
Regular	company amazon tech sales market billion stock companies walmart investors
Interactive	protests protesters anti brazil government march law protest called violence
Regular	president obama memorial american national police clinton happening watched updated
Interactive	birth child children family baby mother health form care work
Regular	scenes show marshall left work back national published media film
Interactive	economy government published companies clinton banks president country billion economic
Regular	fed trump rates economy rate market yellen jobs growth president
Interactive	female womens house party election men things updated found unfolds
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	climate change natural conditions exxon gas power prices museum water
Regular	earth live star planet space system water show light back
Interactive	gender transgender sex law carolina court rights gay parents men
Regular	million actors highest show star company published clinton president work
Interactive	immigrants trumps mexico trade border jobs immigration plan society wall
Regular	trump students mexican missing clinton trumps donald campaign mexico president
Interactive	cancer health medical doctors hospital patients care study drug disease
Regular	zika health rio virus olympics games states house president government
Interactive	tax income clinton workers jobs rate economy job americans fed
Regular	fed trump rates economy rate market yellen jobs growth president
Interactive	iran oil opec production saudi prices deal crude barrels energy
Regular	oil prices production million saudi market energy barrels billion company
Interactive	retirement budget savings request billion government financial campaign plan security
Regular	fed trump rates economy rate market yellen jobs growth president
Interactive	michelle move obamas obama program lady house president visits speech
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	device images camera photo videos video caught body hands features
Regular	scenes show marshall left work back national published media film
Interactive	book kelly page magazine fox show host photo led allegations
Regular	trump dylan clinton media bob campaign president twitter show published
Interactive	china korean korea chinese chinas japan region military japanese beijing
Regular	korea korean military china nuclear march government report security company
Interactive	trump cnn debate nbc million clinton donald viewers campaign network
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	victims assault sexual victim company fox orlando watched rape attack
Regular	police officers dallas shot protest officer shooting killed man law
Interactive	gun president shooting mass assault weapons shot nuclear killed officers
Regular	police officers dallas shot protest officer shooting killed man law
Interactive	apple iphone phone devices phones note samsung smartphone company tech
Regular	apple iphone company tech apples sales million phone published stock
Interactive	music prince star song fans show fargo wells awards live
Regular	prince remembers purple princes music city twitter death minnesota family
Interactive	devices users smartphone microsoft app google zuckerberg phone internet apps
Regular	google company tech million companies app published billion googles business
Interactive	russian computer fbi system hackers putin information russia intelligence systems
Regular	korea korean military china nuclear march government report security company
Interactive	game nfl games team football players rio olympic league sports
Regular	winners golden series million back globes published film won home
Interactive	cruz kasich debate trumps politics convention rubio pence party democratic
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	syrian refugees isis syria forces turkey military city aleppo government
Regular	syrian isis forces city aleppo march syria government refugees military

Table B.3. Topics created by user 3

Interactive	russian putin russia intelligence obama election information president foreign report
Regular	obama president trump clinton visits house march india january barack
Interactive	china chinas chinese products beijing global kong hong trade foreign
Regular	trump fed economy rates market rate president clinton jobs published
Interactive	dallas police shot officers protest officer shooting department killed man
Regular	police officers dallas shot protest officer law killed man back
Interactive	rio olympic olympics games athletes brazil zika team gold park
Regular	minister prime company million published justin Trudeau police government president
Interactive	transgender law gender president carolina court sex school public bill
Regular	trump clinton company president million dylan published media campaign back
Interactive	water storm weather florida coast emergency hurricane damage city area
Regular	fire police india deadly temple officials company man million published
Interactive	puerto debt island states back house bill home oil job
Regular	trump fed economy rates market rate president clinton jobs published
Interactive	film star show awards disney entertainment director series movie actor
Regular	trump comedy show company million published president back series work
Interactive	film internet netflix media cable tech million fox disney sanders
Regular	trump clinton company president million dylan published media campaign back
Interactive	trade mexico jobs border trumps tariffs mexican immigrants canada american
Regular	trump students mexican clinton missing trumps donald president campaign mexico
Interactive	gas natural prices oil infrastructure fed fuel lines construction security
Regular	oil prices million market company production saudi published billion companies
Interactive	muslim trump clinton campaign muslims immigrants trumps attack attacks american
Regular	trump clinton donald campaign trumps president republican presidential hillary election
Interactive	famous fashion million design worlds auction brands stores home buildings
Regular	company tesla car million published cars musk tech market billion
Interactive	prince music song rock purple death records minnesota group young
Regular	prince company million city published police home back twitter family
Interactive	bank buffett fargo wells oil stock banks market investors financial
Regular	wells fargo bank million company banks published accounts employees sales
Interactive	google apple yahoo verizon phone devices apples note iphone phones
Regular	apple iphone company tech million published billion sales stock companies
Interactive	cruz kasich president clinton convention politics rubio party delegates voters
Regular	trump clinton donald campaign trumps president republican presidential hillary election
Interactive	falluja isis battle syrian forces syria government refugees turkey attacks
Regular	attacks brussels terror airport police attack march isis security paris
Interactive	space station international moments study live found attacks natural school
Regular	company live earth million trump published president back star american
Interactive	trump debate cnn fox nbc cable media lewandowski network trumps
Regular	trump clinton donald campaign trumps president republican presidential hillary election

Table B.4. Topics created by user 4

Interactive	apple internet iphone company tech phone note phones apples billion
Regular	apple iphone company tech apples sales phone million stock published
Interactive	google yahoo verizon microsoft tech workers mayer data app quarter
Regular	google company tech million companies app published billion googles search
Interactive	china obama india president canada trade countries global asia international
Regular	china philippines city police duterte president things chinese government million
Interactive	britain brexit france obama european british london europe attacks french
Regular	winners golden series globes back film published million won home
Interactive	banks rates fed rate treasury assets interest campaign government losses
Regular	fed rates trump economy rate market yellen jobs growth interest
Interactive	politics million democratic election sanders debate party political vote trumps
Regular	clinton trump sanders campaign hillary democratic president donald clintons bernie
Interactive	korea japan china military japanese defense region trade international global
Regular	korea korean military china nuclear march government report security company
Interactive	michelle obama move obamas program lady house kids visits girls
Regular	trump president clinton campaign reagan republican house presidential ronald court
Interactive	mexico trade border tariffs canada jobs tariff countries goods wall
Regular	ford car cars mexico sales company published market trade police
Interactive	carolina law virginia sanders house democratic freewheel michigan senate water
Regular	clinton trump sanders campaign hillary democratic president donald clintons bernie
Interactive	college students school schools student university education high job program
Regular	scenes show marshall left work back film national published media
Interactive	cruz Kasich politics bush Rubio republican florida trumps delegates party
Regular	trump donald campaign trumps clinton republican president cruz presidential election
Interactive	china chinese chinas overseas beijing trade global international kong worlds
Regular	china philippines city police duterte president things chinese government million
Interactive	black african brown family man country poor police children americans
Regular	memorial obama president police american national happening service family updated
Interactive	rio brazil president government protests olympics games march anti mass
Regular	zika health rio virus olympics games states house government olympic
Interactive	city hotel police battle mayor homes forces president iraqi residents
Regular	memorial obama president police american national happening service family updated
Interactive	internet computer hackers million online system russian technology access software
Regular	google company tech million companies app published billion googles search
Interactive	gun black victims shooting shot assault weapons nuclear officers family
Regular	police officers dallas shot protest officer shooting killed man law
Interactive	mexico border trade trumps canada jobs national united region american
Regular	syrian isis forces city aleppo march syria government refugees military
Interactive	fbi information journalists emails intelligence letter law documents court statement
Regular	dylan media bob twitter show fox published company voice president
Interactive	politics vice pence indiana freewheel form rail sync nominee class
Regular	clinton trump sanders campaign hillary democratic president donald clintons bernie
Interactive	disney park film company show national twitter back movie media
Regular	scenes show marshall left work back film national published media
Interactive	internet users zuckerberg mark online facebooks technology access free app
Regular	google company tech million companies app published billion googles search
Interactive	transgender gender schools school sex president law public court gay
Regular	harry potter president back published show house part work twitter
Interactive	hate sources freewheel twitter rail attack attacks sync tweet form
Regular	dylan media bob twitter show fox published company voice president
Interactive	stores amazon sales company products store tech retail macys profit
Regular	company amazon tech sales market billion stock companies walmart investors
Interactive	jobs workers infrastructure products rate manufacturing job economy overseas growth
Regular	fed rates trump economy rate market yellen jobs growth interest
Interactive	health drug patients drugs hospital care disease study heart death
Regular	zika health rio virus olympics games states house government olympic
Interactive	property home estate housing battle prices taxes homes prince company
Regular	uber china company drivers million companies published tech billion chinese
Interactive	trump cnn media nbc fox clinton donald network journalists campaign
Regular	trump million trumps donald campaign clinton president worlds republican media
Interactive	drone space drones launch tech damage happening city government including
Regular	korea korean military china nuclear march government report security company
Interactive	loans debt tax clinton income published government assets billion campaign
Regular	fed rates trump economy rate market yellen jobs growth interest
Interactive	fed dow market rates stocks investors rate economy markets interest
Regular	fed rates trump economy rate market yellen jobs growth interest
Interactive	oil gas iran prices opec production saudi energy deal arabia
Regular	oil prices production million saudi market energy barrels billion company
Interactive	retirement funds budget savings fees fund investing request benefits social
Regular	windows company microsoft tech published work police twitter security back
Interactive	banks oil bank goldman campaign government financial pay billion published
Regular	fed rates trump economy rate market yellen jobs growth interest
Interactive	journalists twitter media tweets hate group web attack attacks anti
Regular	dylan media bob twitter show fox published company voice president
Interactive	police judge victims court trial prison officers enforcement department man
Regular	police officers dallas shot protest officer shooting killed man law
Interactive	president laws national hillary public states presidential happening messenger bill
Regular	trump president clinton campaign reagan republican house presidential ronald court

Table B.5. Topics created by user 5

Interactive	isis syrian refugees attacks terror syria military attack forces city
Regular	syrian isis city forces aleppo march syria government refugees group
Interactive	google apple iphone phone note phones apples device devices users
Regular	apple iphone company tech apples million sales published phone stock
Interactive	trumps mexico border trade mexican immigrants tariffs jobs immigration wall
Regular	trump students mexican missing clinton trumps donald campaign mexico president
Interactive	muslim trump clinton campaign muslims attack donald hate attacks president
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	trump debate donald clinton cooper class campaign president kelly cnn
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	players game nfl football league season games team night national
Regular	scenes show marshall left work back national published clinton media
Interactive	rio players olympic olympics sports games athletes golf team gold
Regular	winners golden million series back published globes company home film
Interactive	water storm weather damage florida coast conditions hurricane city rain
Regular	zika health rio virus olympics games states house president government
Interactive	fbi russian intelligence putin russia information cnn emails clintons security
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	retirement savings benefits older bonds social obamacare age stocks insurance
Regular	zika health rio virus olympics games states house president government
Interactive	students college school student schools university education high program job
Regular	scenes show marshall left work back national published clinton media
Interactive	gun police victims shooting government weapons mass president judge officers
Regular	police officers dallas shot protest officer killed shooting man law
Interactive	black protests mass african march anti government protesters back protest
Regular	police officers dallas shot protest officer killed shooting man law
Interactive	yahoo verizon company tech deals billion deal business million published
Regular	company tech amazon sales market billion companies stock published walmart
Interactive	tax income workers rate clinton unemployment manufacturing benefits jobs gap
Regular	fed trump rates economy rate market yellen jobs president growth
Interactive	tax banks bank funds income rates returns clinton jobs stock
Regular	fed trump rates economy rate market yellen jobs president growth
Interactive	debt loans losses loan published payments pay interest auto company
Regular	company tech amazon sales market billion companies stock published walmart
Interactive	rio olympic zika olympics brazil games virus brain disease health
Regular	zika health rio virus olympics games states house president government
Interactive	banks economy market fed jobs rate investors bank hike companies
Regular	fed trump rates economy rate market yellen jobs president growth
Interactive	uber drivers car driving cars tesla driver electric ride musk
Regular	tesla car musk company cars teslas published electric driving million
Interactive	cancer study health heart patients researchers brain care drug medical
Regular	earth live star planet space system water company million back
Interactive	korean nuclear iran korea weapons deal kim military defense foreign
Regular	korea korean china military government march nuclear report president company
Interactive	airport flight plane aircraft airlines passengers international security american worlds
Regular	brussels attacks terror airport police attack march paris security isis
Interactive	oil gas prices opec production energy saudi fuel crude barrels
Regular	oil prices production million saudi market energy barrels billion company
Interactive	museum art million design visitors work london history national part
Regular	earth live star planet space system water company million back
Interactive	fbi computer system hackers intelligence systems information security cnn email
Regular	korea korean china military government march nuclear report president company
Interactive	players williams team cup career won title teams womens player
Regular	scenes show marshall left work back national published clinton media
Interactive	film star music prince show disney rock song awards company
Regular	prince remembers purple princes music city police family home death
Interactive	india obama china president global trade africa poor delhi march
Regular	korea korean china military government march nuclear report president company
Interactive	chinas chinese china beijing products quality kong hong growth region
Regular	uber company china million drivers companies tech published billion chinese
Interactive	transgender gender law sex assault sexual fox gay men media
Regular	trump dylan clinton media bob campaign president twitter show published
Interactive	trump cruz politics bush kasich trumps rubio party freewheel convention
Regular	trump clinton donald campaign trumps republican president presidential hillary cruz
Interactive	court judge president supreme hearing ruling laws justice public federal
Regular	trump clinton president campaign donald trumps republican presidential hillary house
Interactive	brexit european obama britain british london germany england vote europe
Regular	winners golden million series back published globes company home film

Appendix C

Supplementary material for Part III

C.1 Vapnik-Chervonenkis dimension

Definition C.1. For any hypothesis class \mathcal{H} and any $S \subseteq \mathcal{X}$,

$$\Pi_{\mathcal{H}}(S) = \{h \cup S : h \in \mathcal{X}\}.$$

Equivalently, if $S = \{x_1, \dots, x_m\}$ then we can think of $\Pi_{\mathcal{H}}$ as the set of vectors $\Pi_{\mathcal{H}} \subseteq \{0, 1\}^m$ defined by

$$\Pi_{\mathcal{H}}(S) = \{h(x_1), \dots, h(x_m) : h \in \mathcal{H}\}.$$

Thus, $\Pi_{\mathcal{H}}(S)$ is the set of all the behaviors on S that are realized by \mathcal{H} .

Definition C.2. If $\Pi_{\mathcal{H}}(S) = \{0, 1\}^m$ (where $m = |S|$), then we say that S is shattered by \mathcal{H} . Thus, S is shattered by \mathcal{H} if \mathcal{H} realizes all possible behaviors S .

Definition C.3. The Vapnik-Chervonenkis (VC) dimension of \mathcal{H} , is the cardinality d of the largest set S shattered by \mathcal{H} .

C.2 Proof of Theorem 9.3

The proof of Theorem 9.3 follows that of the original online set cover algorithm [1]. We provide it here for reference and because it differs on several small details.

Lemma C.4. *Let t be the size of an optimal teaching set for \mathcal{H} . Then the total number of doubling steps performed by the algorithm is at most $t \cdot \lg(2m)$, and at any point in time,*

$$\sum_{x \in \mathcal{X}} w(x) \leq 1 + t \cdot \lg(2m).$$

Proof. First, $w(x) \leq 2$ for all x , always. This is because $w(x)$ increases only during a doubling step, which happens only if x belongs to a subset of \mathcal{X} of total weight < 1 .

Let $T^* \subset \mathcal{X}$ denote an optimal teaching set, of size t . By definition, T^* must intersect $\Delta(h)$ for all $h \neq h^*$. Now, a doubling step doubles the weight of each $x \in \Delta(h)$, and thus some element of T^* . And since the weight of an individual point begins at $1/m$ and never exceeds 2, the total number of doubling steps cannot exceed $t \cdot \lg(2m)$.

During each doubling step, $w(\Delta(h))$, and thus $\sum_x w(x)$, increases by at most 1. The lemma follows by noting that the initial value of this summation is 1, and there are at most $t \cdot \lg(2m)$ doubling steps. \square

Lemma C.5. *With probability at least $1 - \delta$, at the end of any iteration of the main loop, any hypothesis $h \neq h^*$ with $w(\Delta(h)) \geq 1$ is invalidated by the teaching examples.*

Proof. Fix any $h \neq h^*$ and consider the first point in time at which $w(\Delta(h)) \geq 1$. Recall that the thresholds T_x are drawn from an exponential distribution with rate $\lambda = \ln(N/\delta)$. Thus the probability, over the random choice of thresholds, that no point in $\Delta(h)$ is chosen

as a teaching example is

$$\begin{aligned}
\prod_{x \in \Delta(h)} \Pr(w(x) \leq T_x) &= \prod_{x \in \Delta(h)} \exp(-\lambda w(x)) \\
&= \exp(-\lambda w(\Delta(h))) \\
&\leq \exp(-\lambda) = \frac{\delta}{N}.
\end{aligned}$$

Now take a union bound over all N hypotheses in \mathcal{H} . □

Lemma C.6. *The expected total number of teaching examples provided is at most $(1 + t \lg(2m)) \ln(N/\delta)$.*

Proof. The probability that any particular $x \in \mathcal{X}$ is eventually provided as a teaching example is

$$\begin{aligned}
&\Pr(\text{final value of } w(x) \text{ exceeds } T_x) \\
&= 1 - \Pr(T_x > w(x)) \\
&= 1 - \exp(-\lambda w(x)) \leq \lambda w(x)
\end{aligned}$$

where $\lambda = \ln(N/\delta)$ is the rate parameter of the exponential distribution from which T_x is chosen. Thus

$$\mathbb{E}[|S|] \leq \sum_{x \in \mathcal{X}} \lambda w(x) \leq \lambda(1 + t \lg(2m)),$$

where the last inequality invokes Lemma C.4. □

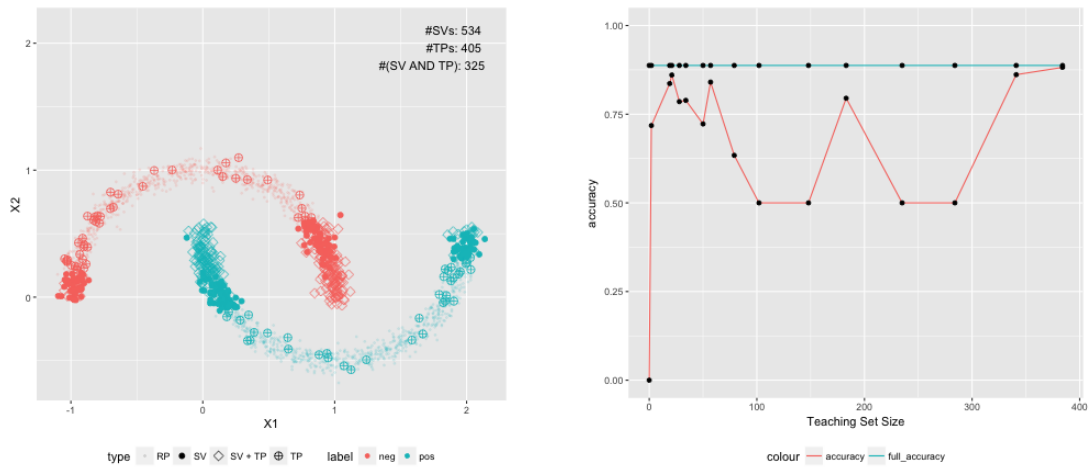


Figure C.1. Moon-shaped dataset (separable), Linear kernel

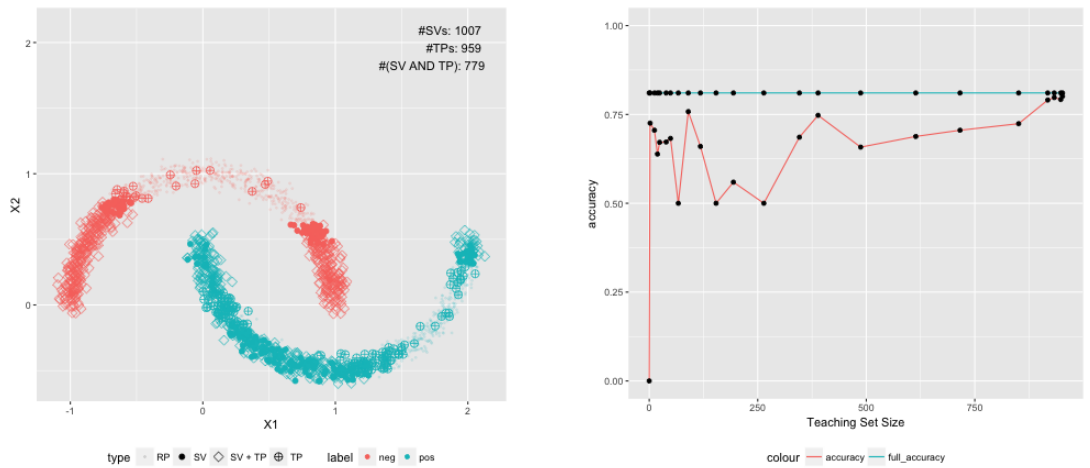


Figure C.2. Moon-shaped dataset (separable), Quadratic kernel

C.3 Experimental results

Below, we give the full set of experimental results on synthetic and real datasets.

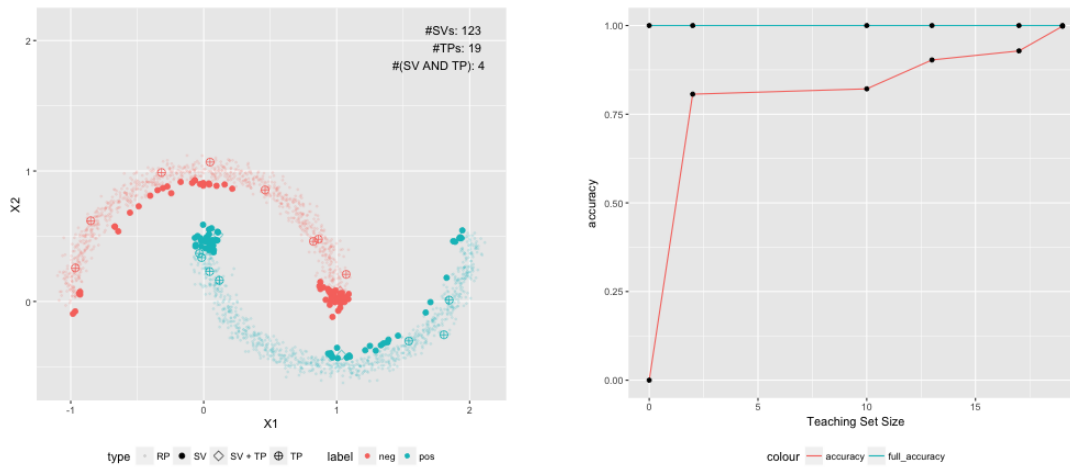


Figure C.3. Moon-shaped dataset (separable), RBF kernel

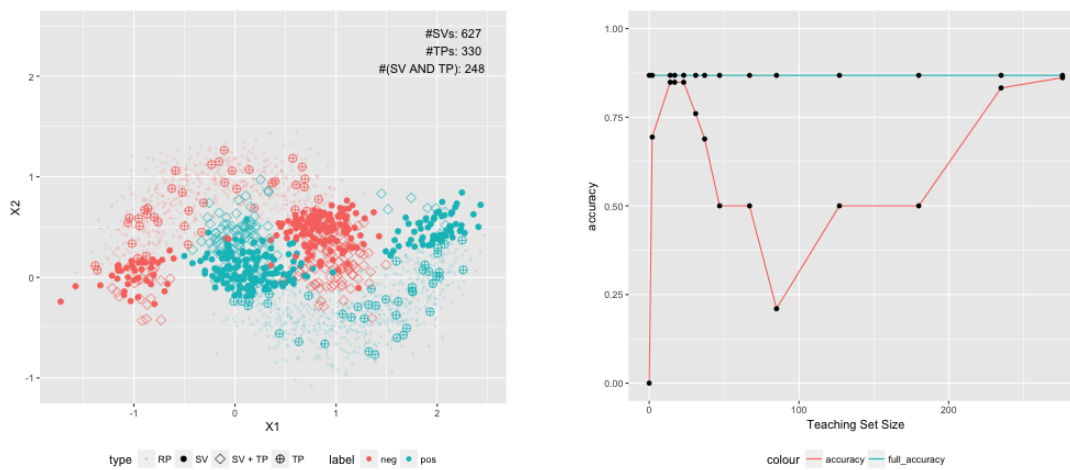


Figure C.4. Moon-shaped dataset (non-separable), Linear kernel

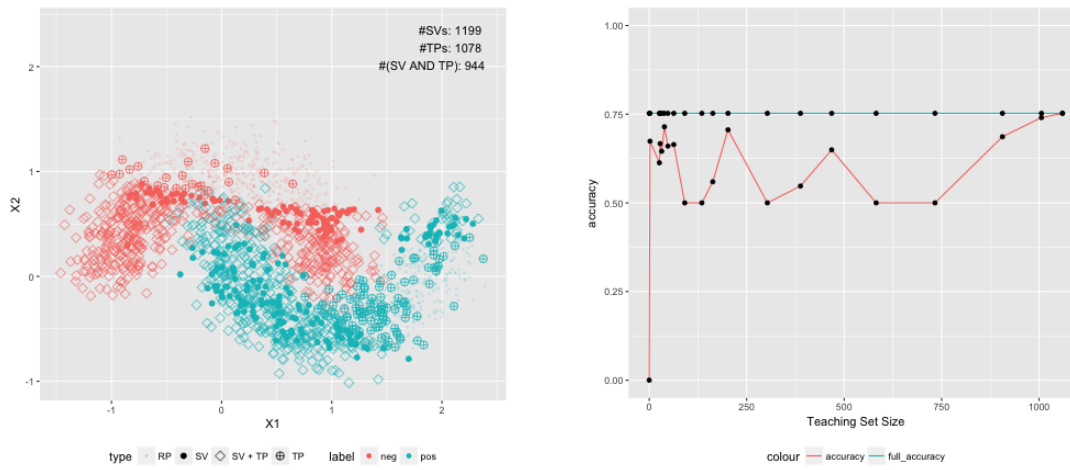


Figure C.5. Moon-shaped dataset (non-separable), Quadratic kernel

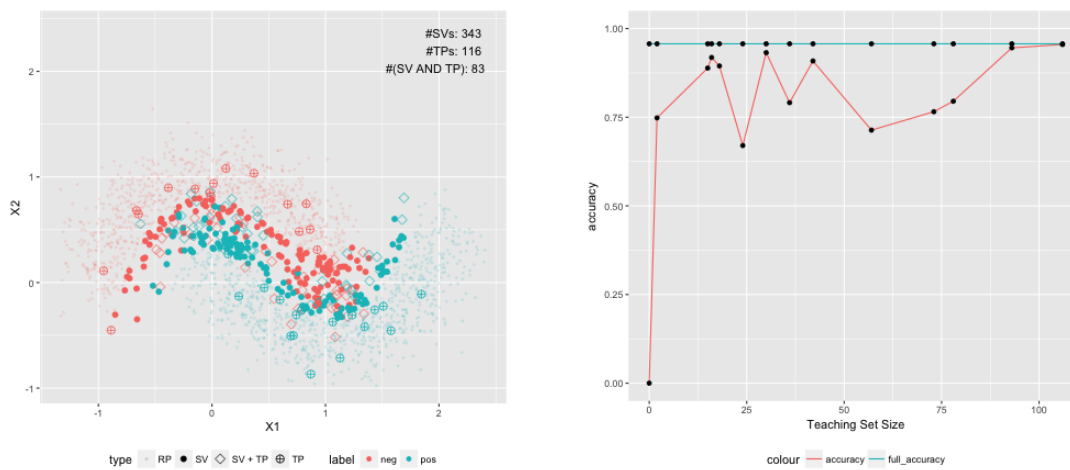


Figure C.6. Moon-shaped dataset (non-separable), RBF kernel

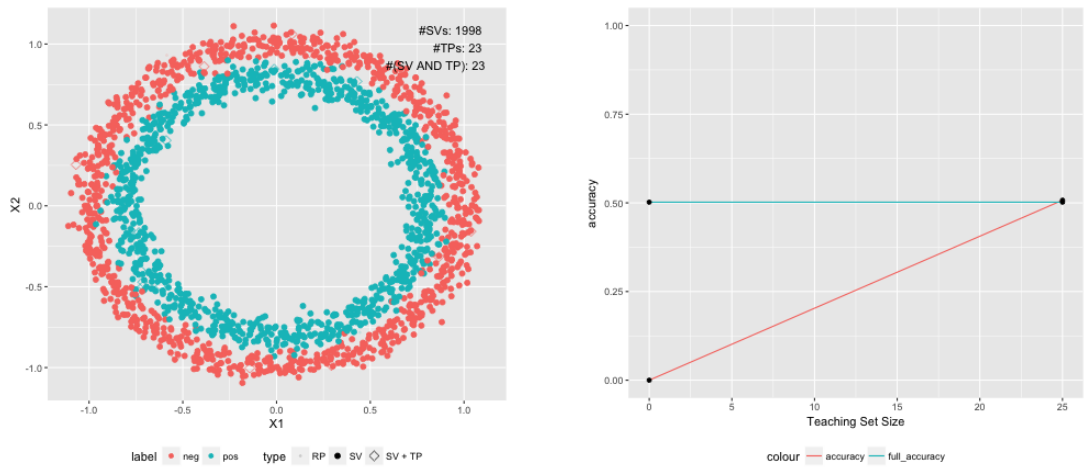


Figure C.7. Circular dataset (separable), Linear kernel

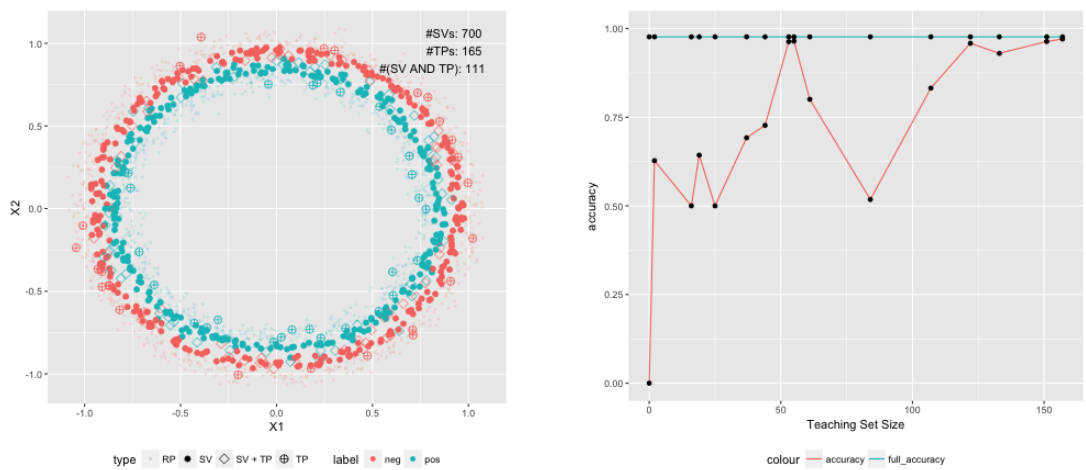


Figure C.8. Circular dataset (separable), Quadratic kernel

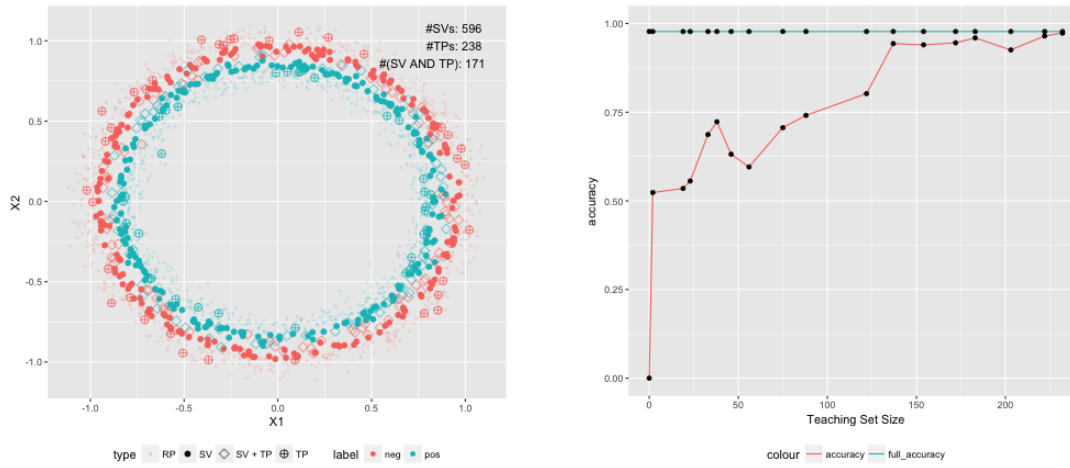


Figure C.9. Circular dataset (separable), RBF kernel

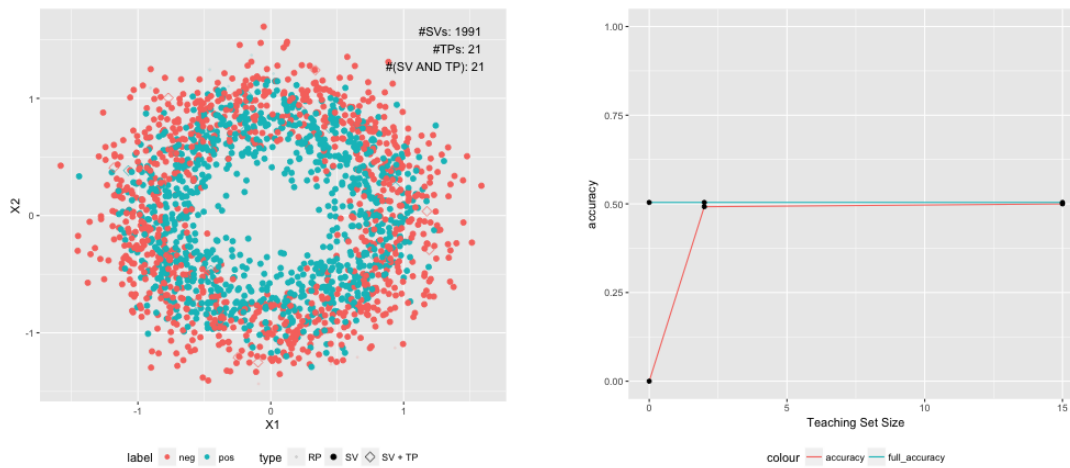


Figure C.10. Circular dataset (non-separable), Linear kernel

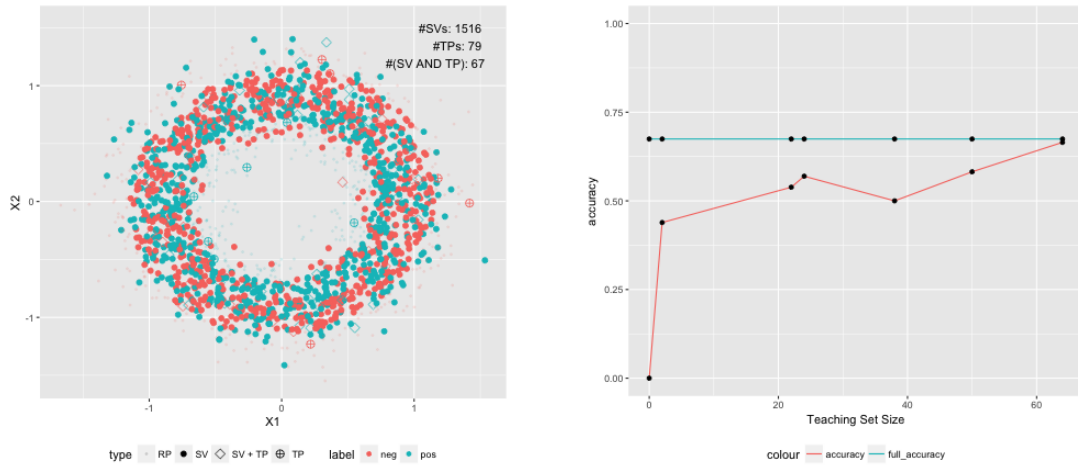


Figure C.11. Circular dataset (non-separable), Quadratic kernel

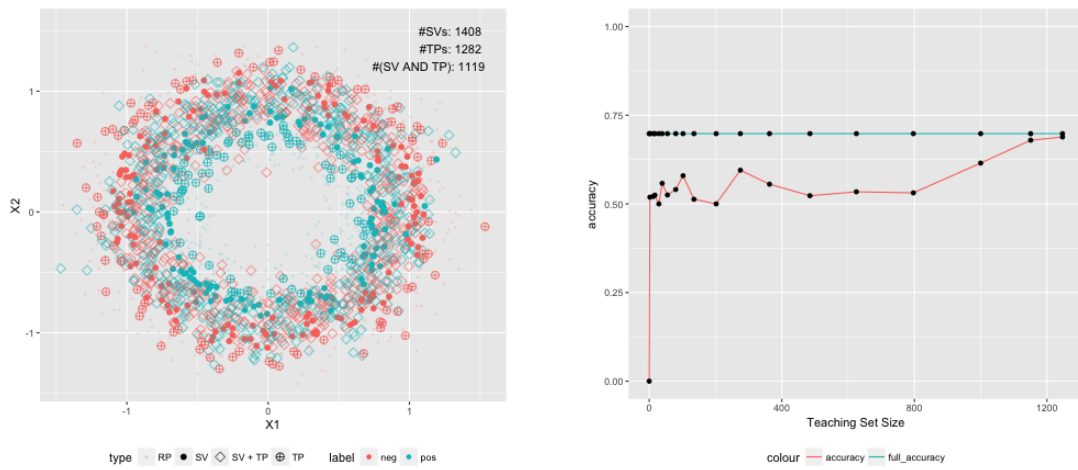


Figure C.12. Circular dataset (non-separable), RBF kernel

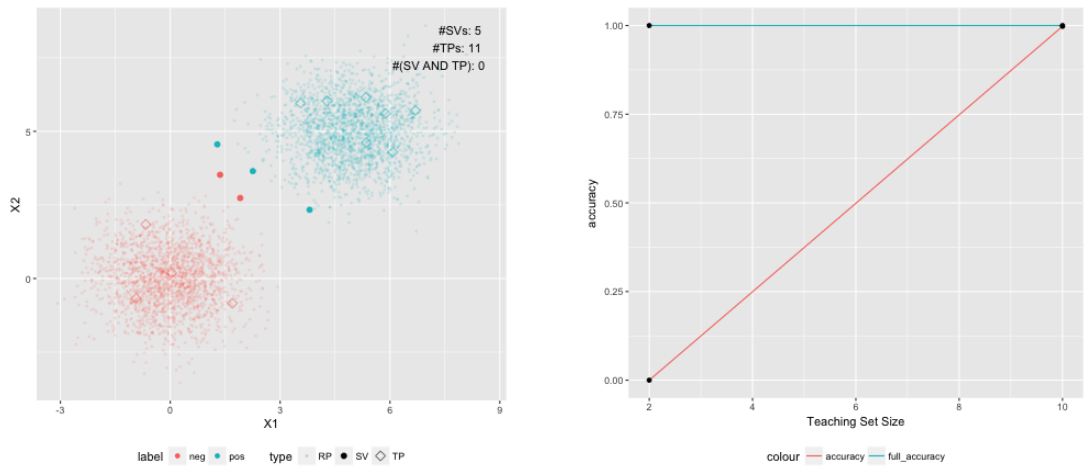


Figure C.13. Mixtures of Gaussians dataset (separable), Linear kernel

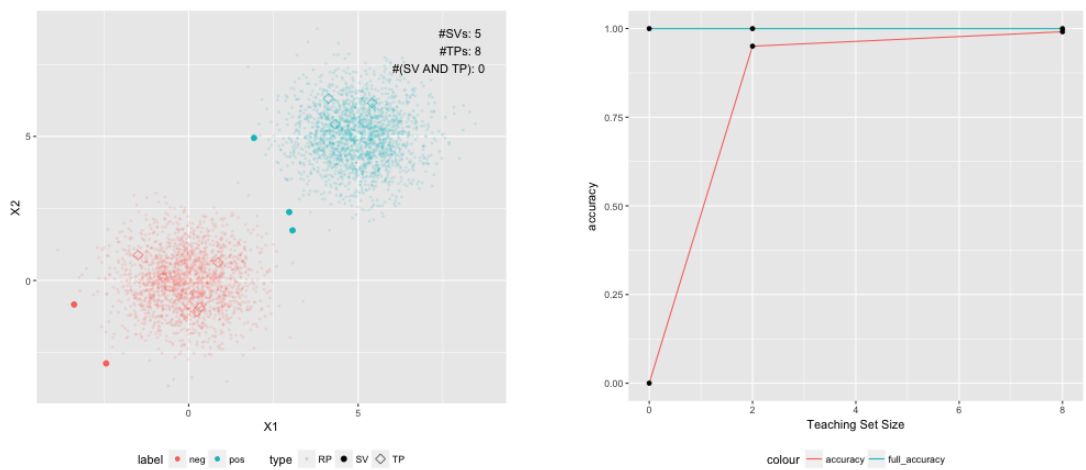


Figure C.14. Mixtures of Gaussians dataset (separable), Quadratic kernel

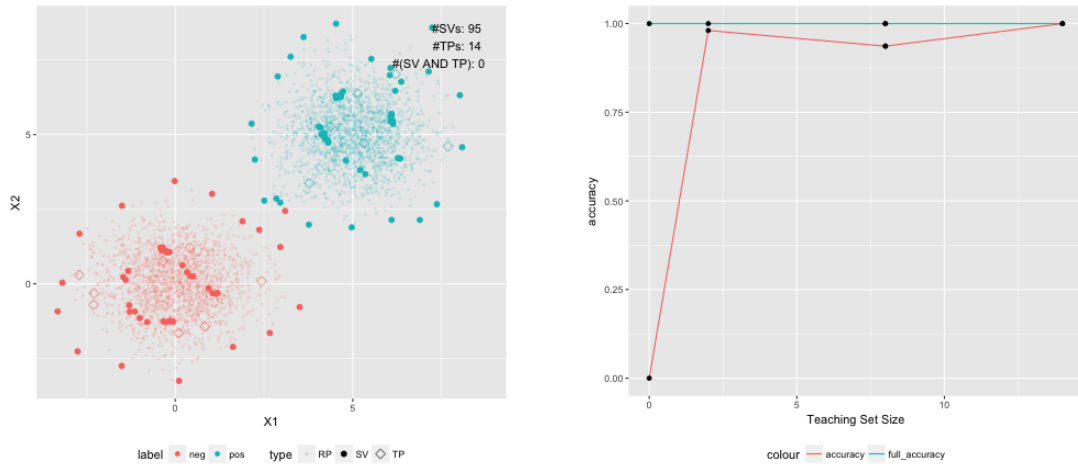


Figure C.15. Mixtures of Gaussians dataset (separable), RBF kernel

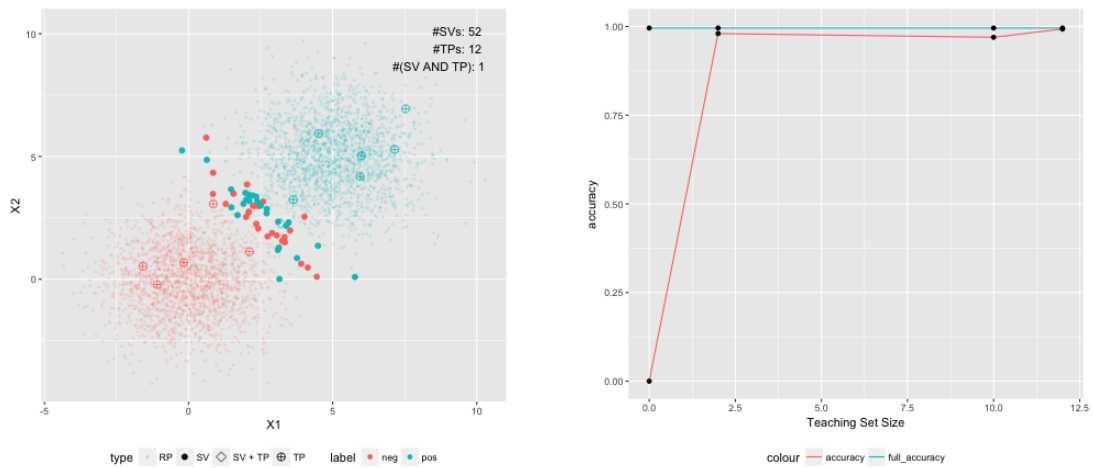


Figure C.16. Mixtures of Gaussians dataset (non-separable), Linear kernel

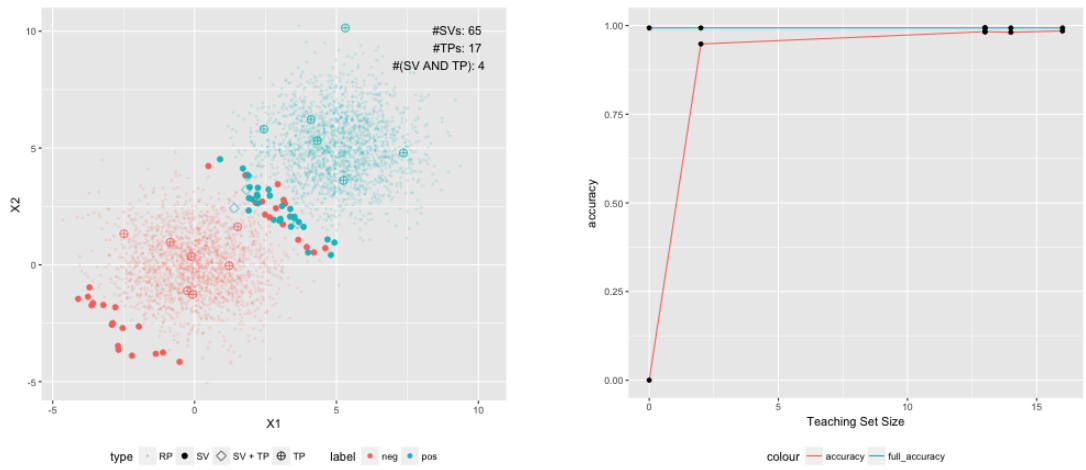


Figure C.17. Mixtures of Gaussians dataset (non-separable), Quadratic kernel

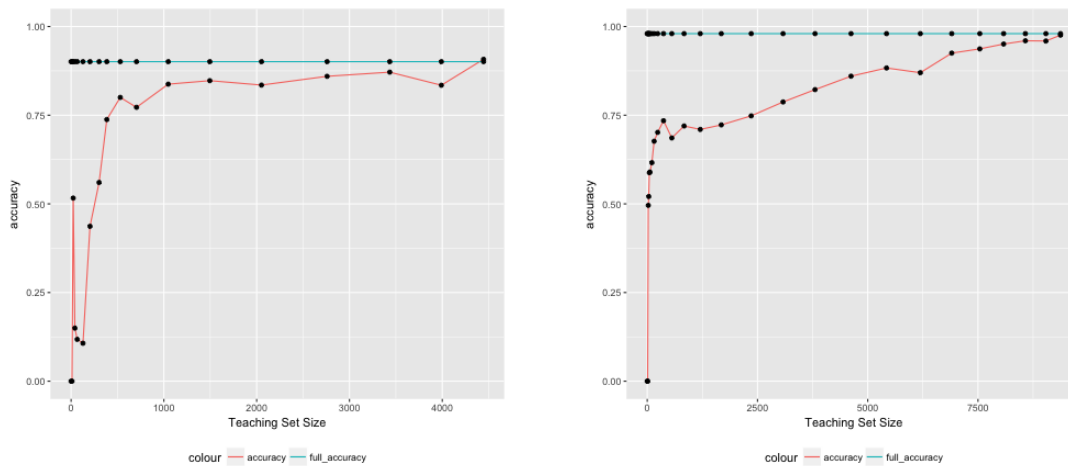


Figure C.18. (a) MNIST data set, quadratic kernel SVM (b) Fashion MNIST data set, convolutional neural network

Table C.1. Number of SVs, TPs, and points that are both SVs and TPs on MNIST.

# SVs	32,320
# TPs	4,445
#TPs AND SVs	4,357

Bibliography

- [1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and S. Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.
- [2] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 25–32, 2009.
- [3] M. Anthony, G. Brightwell, D. Cohen, and J. Shawe-Taylor. On exact specification by examples. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 311–318, 1992.
- [4] S. Arora, R. Ge, and A. Moitra. Learning topic models—going beyond svd. In *Foundations of Computer Science (FOCS)*, pages 1–10, 2012.
- [5] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning*, pages 280–288, 2013.
- [6] H. Ashtiani, S. Kushagra, and S. Ben-David. Clustering with same-cluster queries. In *Advances in Neural Information Processing Systems*, pages 3216–3224, 2016.
- [7] P. Awasthi, M-F. Balcan, and K. Voevodski. Local algorithms for interactive clustering. In *Proceedings of the 31st International Conference on Machine Learning*, pages 550–558, 2014.
- [8] M-F. Balcan and A. Blum. Clustering with interactive feedback. In *International Conference on Algorithmic Learning Theory*, pages 316–328, 2008.
- [9] P. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.
- [10] D. Blei and J. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

- [11] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 1:601–608, 2002.
- [12] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, 2010.
- [13] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.
- [14] A. Cardoso-Cachopo. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.
- [15] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.
- [16] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
- [17] W.B. Croft and R. Das. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 349–368, 1990.
- [18] S. Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011.
- [19] A. Dayanik, D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 493–500. ACM, 2006.
- [20] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *2011 International Conference on Computer Vision*, pages 1395–1402. IEEE, 2011.
- [21] G Druck, G Mann, and A McCallum. Reducing annotation effort using generalized expectation criteria (technical report 2007-62). *University of Massachusetts, Amherst*, 2007.
- [22] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of ACM Special Interest Group on Information Retrieval*, 2008.

- [23] S. Floyd and M.K. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.
- [24] Z. Gao, C. Ries, H. Simon, and S. Zilles. Preference-based teaching. *Journal of Machine Learning Research*, 18(31):1–32, 2017.
- [25] S.A. Goldman and M.J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.
- [26] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.
- [27] L. Hu, R. Wu, T. Li, and L. Wang. Quadratic upper bound for recursive teaching dimension of finite VC classes. In *Proceedings of the 30th Conference on Learning Theory, COLT*, pages 1147–1156, 2017.
- [28] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine learning*, 95(3):423–469, 2014.
- [29] S. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.
- [30] M. Lee and D. Mimno. Low-dimensional embeddings for interpretable anchor-based topic inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1328, 2014.
- [31] D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
- [32] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine learning*, pages 577–584, 2006.
- [33] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
- [34] N. Littlestone and M.K. Warmuth. Relating data compression and learnability. *Unpublished*, 1986.
- [35] J. Liu and X. Zhu. The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25, 2016. URL <http://jmlr.org/papers/v17/15-630.html>.
- [36] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L.B. Smith, J. Rehg, and L. Song.

- Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 2149–2158, 2017.
- [37] W. Liu, B. Dai, X. Li, Z. Liu, J. Rehg, and L. Song. Towards black-box iterative machine teaching. In *International Conference on Machine Learning*, pages 3147–3155, 2018.
- [38] J. Lund, C. Cook, K. Seppi, and J. Boyd-Graber. Tandem anchoring: A multiword anchor approach for interactive topic modeling. In *Association for Computational Linguistics*, 2017.
- [39] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.
- [40] R. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 483–486. IEEE, 2004.
- [41] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272, 2011.
- [42] S. Moran and A. Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM*, 63(3), 2016.
- [43] J. Petterson, W. Buntine, S. Narayanamurthy, T. Caetano, and A. Smola. Word features for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1921–1929, 2010.
- [44] H. Raghavan and J. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86. ACM, 2007.
- [45] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 841–846, 2005.
- [46] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *The Journal of Machine Learning Research*, 7:1655–1686, 2006.
- [47] B. Recht, C. Re, J. Tropp, and V. Bittorf. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222,

2012.

- [48] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, volume 2, pages 538–545, 2002.
- [49] S. Dasgupta, A. Dey, N. Roberts, and S. Sabato. Learning from discriminative feature feedback. In *Advances in Neural Information Processing Systems*, pages 3955–3963, 2018.
- [50] B. Settles. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In *Empirical Methods in Natural Language Processing*, 2011.
- [51] A. Shinohara and S. Miyano. Teachability in computational learning. *New Generation Computing*, 8(4):337–347, 1991.
- [52] V. Sindhwani, P. Melville, and R. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 953–960. ACM, 2009.
- [53] K. Small, B. Wallace, T. Trikalinos, and C.E. Brodley. The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 865–872, 2011.
- [54] Q. Sun and G. DeJong. Explanation-augmented svm: an approach to incorporating domain knowledge into svm learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 864–871. ACM, 2005.
- [55] C. Tosh and S. Dasgupta. Diameter-based active learning. In *ICML*, 2017.
- [56] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [57] S. Vikram and S. Dasgupta. Interactive bayesian hierarchical clustering. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2081–2090, 2016.
- [58] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th Annual International Conference on Machine Learning*, pages 577–584, 2001.
- [59] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333. ACM, 2004.

- [60] H. Xiao, K. Rasul, and R. Vollgraf. Fashion MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv ePrints*, 2017.
- [61] O.F. Zaidan and J. Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of EMNLP 2008*, pages 31–40, October 2008.
- [62] O.F. Zaidan, J. Eisner, and C. Piatko. Using “annotator rationales” to improve machine learning for text categorization. In *NAACL HLT 2007; Proceedings of the Main Conference*, pages 260–267, April 2007.
- [63] X. Zhu, J. Liu, and M. Lopes. No learner left behind: On the complexity of teaching multiple learners simultaneously. In *The 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.
- [64] X. Zhu, A. Singla, S. Zilles, and A.N. Rafferty. An overview of machine teaching. *ArXiv e-prints*, January 2018. <https://arxiv.org/abs/1801.05927>.
- [65] S. Zilles, S. Lange, R. Holte, and M. Zinkevich. Models of cooperative teaching and learning. *J. Mach. Learn. Res.*, 12:349–384, 2011.