

**ADAPTIVE GRAPH-BASED ALGORITHMS FOR
CONDITIONAL ANOMALY DETECTION AND
SEMI-SUPERVISED LEARNING**

by

Michal Valko

MSc. Comenius University, Bratislava, 2005

Submitted to the Graduate Faculty of
the Computer Science Department in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH
COMPUTER SCIENCE DEPARTMENT

This dissertation was presented

by

Michal Valko

It was defended on

August 1st 2011

and approved by

Milos Hauskrecht, PhD, Associate Professor, Computer Science

G. Elisabeta Marai, PhD, Assistant Professor, Computer Science

Diane Litman, PhD, Professor, Computer Science

John Lafferty, PhD, Professor, Machine Learning (Carnegie Mellon University)

Dissertation Director: **Milos Hauskrecht, PhD**, Associate Professor, Computer Science

Copyright © by **Michal Valko**

2011

ADAPTIVE GRAPH-BASED ALGORITHMS FOR CONDITIONAL ANOMALY DETECTION AND SEMI-SUPERVISED LEARNING

Michal Valko, PhD

University of Pittsburgh, 2011

We develop graph-based methods for conditional anomaly detection and semi-supervised learning based on label propagation on a data similarity graph. When data is abundant or arrive in a stream, the problems of computation and data storage arise for any graph-based method. We propose a fast approximate online algorithm that solves for the harmonic solution on an approximate graph. We show, both empirically and theoretically, that good behavior can be achieved by collapsing nearby points into a set of local representative points that minimize distortion. Moreover, we regularize the harmonic solution to achieve better stability properties.

We also present graph-based methods for detecting conditional anomalies and apply it to the identification of unusual clinical actions in hospitals. Our hypothesis is that patient-management actions that are unusual with respect to the past patients may be due to errors and that it is worthwhile to raise an alert if such a condition is encountered. Conditional anomaly detection extends standard unconditional anomaly framework but also faces new problems known as fringe and isolated points. We devise novel nonparametric graph-based methods to tackle these problems. Our methods rely on graph connectivity analysis and soft harmonic solution. Finally, we conduct an extensive human evaluation study of our conditional anomaly methods by 15 experts in critical care.

TABLE OF CONTENTS

PREFACE	xv
1.0 INTRODUCTION	1
1.1 Motivation	1
1.2 Thesis Statement and Main Contributions	4
1.3 Organization of the Thesis	5
2.0 BACKGROUND AND RELATED WORK	6
2.1 Related Work in Anomaly Detection (AD)	6
2.1.1 Unconditional Anomaly Detection	6
2.1.1.1 Graph-based methods	7
2.1.1.2 Approaches with prelabeled anomalies	8
2.1.1.3 Rare Category Detection	9
2.1.2 Conditional Anomaly Detection (CAD)	9
2.2 Background in Graph Theory	11
2.2.1 Graph as a data model	11
2.2.1.1 Stationary distribution of a random walk	12
2.2.2 Regularized Harmonic Function	13
2.2.2.1 Soft harmonic solution	14
2.3 Related work in Online Quantization	16
2.4 Related Work in Semi-Supervised Learning (SSL)	17
2.4.1 Semi-supervised Max-Margin Learning	18
2.4.1.1 Semi-supervised SVMs	18
2.4.1.2 Manifold regularization of SVMs	19

2.4.2 Online Semi-Supervised Learning	19
3.0 METHODS	21
3.1 Conditonal Anomaly Detection	21
3.1.1 Introduction to Conditonal Anomaly Detection	21
3.1.2 Definition of Conditional Anomaly	22
3.1.3 Relationship to Mislabeling Detection	23
3.1.4 Class-outlier approach	24
3.1.5 Discriminative approach	25
3.1.5.1 CAD with Random Walks	26
3.1.6 Regularized discriminative approach	27
3.1.6.1 Regularized Random Walk CAD	28
3.1.6.2 Conditional Anomaly Detection with Soft Harmonic Functions	29
3.2 Semi-Supervised Learning	33
3.2.1 Max-Margin Graph Cuts	34
3.2.2 Joint Quantization and Label Propagation	35
3.2.2.1 Label Propagation:	37
3.2.2.2 Quantization:	37
3.2.2.3 Approximation	38
3.2.2.4 Time Complexity	39
3.3 Online Semi-Supervised Learning With Quantized Graphs	40
3.3.1 Incremental k-centers	41
3.4 Parallel Multi-Manifold Learning	42
4.0 THEORETICAL ANALYSIS	45
4.1 Analysis of Conditional Anomaly Detection	45
4.1.1 RWCAD and its Relationship to the weighted k -NN	45
4.1.2 Soft Harmonic Solution	46
4.2 Analysis of Max-margin Graph Cuts	48
4.2.1 When Manifold Regularization Fails	48
4.2.2 Generalization Error	50
4.2.3 Threshold epsilon	53

4.3	Analysis of Joint Quantization and Label Propagation	55
4.4	Analysis of Online SSL on Quantized graphs	56
4.4.1	Bounding Transduction Error (4.11)	58
4.4.2	Bounding Online Error (4.12)	58
4.4.3	Bounding Quantization Error (4.13)	59
4.4.4	Discussion	63
4.5	Parallel Multi-Manifold Learning	63
5.0	EXPERIMENTS	65
5.1	Datasets	65
5.1.1	Post-surgical cardiac patients (PCP)	65
5.1.1.1	Visit/Demographic Features	67
5.1.1.2	Lab Features	67
5.1.1.3	Medications	68
5.1.1.4	Procedure Features	68
5.1.1.5	Heart Support Device Features	69
5.1.1.6	Orders/Labels	69
5.1.2	UCI ML Datasets	69
5.1.2.1	Digit recognition	69
5.1.2.2	Letter recognition	70
5.1.2.3	Image segmentation	70
5.1.2.4	Coil	70
5.1.2.5	Car	70
5.1.2.6	SecStr	70
5.1.3	Synthetic Datasets	71
5.1.3.1	Core Dataset	71
5.1.3.2	Mixtures of Gaussians	71
5.1.4	Vision Datasets	72
5.2	Algorithms for Comparison	73
5.2.1	Discriminative SVM Anomaly Detection	73
5.2.2	One-Class SVM	74

5.2.3	Quadratic discriminant analysis	74
5.2.4	Weighted NN	74
5.2.5	Parameters for the Graph-Based Algorithms	75
5.3	Evaluations of Semi-Supervised Learning Models	76
5.3.1	Max-margin Graph Cuts Experiments	76
5.3.1.1	Synthetic Problem	76
5.3.1.2	UCI ML Repository Datasets	77
5.3.2	Joint Quantization and Label Propagation Experiments	78
5.3.2.1	Experimental setup	79
5.3.2.2	Results	79
5.3.3	Online Quantized SSL Experiments	80
5.3.3.1	UCI ML Repository Datasets	80
5.3.3.2	Face Recognition	82
5.3.4	Parallel SSL	84
5.3.5	Conclusions	86
5.4	Evaluations of CAD	86
5.4.1	Evaluation of CAD with Known Ground Truth	87
5.4.1.1	CAD on Synthetic Datasets with Known Distribution	87
5.4.1.2	CAD on UCI ML Datasets with ordinal response variable	88
5.4.1.3	CAD on Core Dataset with Fringe Points	89
5.4.1.4	Conclusions	91
5.4.2	Evaluation of Expert Assessed Clinically Useful Anomalies	91
5.4.2.1	Pilot study in 2009	92
5.4.2.2	Soft Harmonic Anomaly Detection	94
5.4.2.3	Conclusions	96
6.0	DISCUSSIONS	98
	BIBLIOGRAPHY	101

LIST OF TABLES

1	Mean anomaly AUROC and variance on three synthetic datasets	88
2	Mean anomaly agreement score and variance on 3 UCI ML datasets	90

LIST OF FIGURES

1	Conditional vs. unconditional anomalies	2
2	Disadvantages of nearest neighbor approach for conditional anomaly detection	3
3	a. Similarity graph b. Three regularized harmonic solutions	15
4	Challenges for CAD: 1) fringe and 2) isolated points	25
5	Estimating class-conditional probabilities from two similarity graphs	27
6	Unlabeled orders for ‘positive’ patients	33
7	Running time for different methods on the SecStr dataset	39
8	Estimating likelihood ratio from a single graph	46
9	Linear, cubic, and RBF decision boundaries for different methods.	51
10	The thresholded empirical risk	55
11	Processing of data in the electronic health record	66
12	Examples of temporal features for continuous lab values	68
13	The three synthetic datasets with known underlying distributions	71
14	Snapshots from the environment adaptation and office space datasets	72
15	Face-based authentication dataset (left) and examples of labeled faces (right).	73
16	The weight matrix for 100 negative and 100 positive cases of HPF4 order . . .	75
17	Comparison of SVMs, GC and MR on 3 datasets from the UCI ML repository .	77
18	Coil and Car datasets from UCI ML Repository	81
19	UCI ML: Quality of approximation as a function of time	82
20	UCI ML: Quality of approximation as a function of number of centroids	83
21	Comparison of 3 face recognizers on 2 face recognition datasets	84
22	Speedups in the total, inference, and similarity computation times	85

23	Computation time comparison for the three graph-based methods	89
24	Conditional anomaly detection on a synthetic <i>Core</i> dataset	90
25	Histogram of alert examples in the study according to their alert score	92
26	The relationship between the alert score and the true alert rate	94
27	Histogram of anomaly scores for 2 different tasks	95
28	Medical Dataset: Varying graph size	96
29	Medical Dataset: Varying regularization	97

LIST OF ALGORITHMS

1	RWCAD that calculates the anomaly score	29
2	Quantized semi-supervised learning with principal manifolds	38
3	Online quantized harmonic solution	43

LIST OF EQUATIONS

2.1	Closed form solution for a stationary distribution of the random walk.	12
2.5	Soft harmonic solution	14
2.6	Closed form for soft harmonic solution	15
3.1	Expected value of the n-th order statistic for the standard normal	23
3.2	Posterior probability estimation from a random walk	26
3.3	Regularization of the discriminative approach for CAD	28
3.4	Confidence from the soft label	30
3.5	Anomaly score for soft harmonic anomaly detection	30
3.6	Soft harmonic solution using matrix inversion	31
3.7	Soft harmonic solution using system of linear equations	31
3.8	Compact computation of harmonic solution for the backbone graph	32
3.10	Quantized unconstrained regularization	36
3.11	Joint quantization and soft harmonic solution	36
3.12	Quantization step	37
3.13	Approximate quantization step	37
3.14	Solution for the quantization step	38
4.1	Class-conditional probability of a different label	45
4.2	Approximation of the likelihood ratio	45
4.3	Estimation expressed with similarity weights	46
4.4	Equivalent form of RWCAD algorithm	46
4.5	Linear manifold regularization	48
4.7	Linear support vector machine	49

4.8 Risk of our solutions	50
4.9 Empirical risk on graph induced labels	50
4.10 Elastic net	56
4.14 Upper bound on the online regret	59
4.15 Upper bound on stability of quantized harmonic solution	60
4.16 Upper bound on the quality of quantization	60
4.17 Approximating Laplacian by block-diagonal structure	64
4.18 Approximation bound of block-diagonal structure	64

PREFACE

[Acknowledgments will be placed here in the final version.]

The work was supported by the NIH grants R21 LM009102–01A1 and 1R01LM010019-01A1.

1.0 INTRODUCTION

1.1 MOTIVATION

The focus of this thesis is on new computational anomaly–detection techniques and their application to clinical data and clinical problems.

Anomaly detection techniques are used to identify anomalous (unusual) patterns in data. In clinical settings, these may concern identifications of unusual patient–state outcomes or unusual patient-management decisions.

Our ability to detect unusual events in clinical data may have a tremendous impact on the health care and its quality. First, the identification of an action that differs from an expected or usual pattern of care can aid in detection and prevention of the potential medical errors. According to the HealthGrades study (Wall Street Journal on July 27, 2004), medical errors account for 200 000 preventable deaths a year. Second, the identification of anomalous patient responses can help us to identify new promising treatments.

The existing error detection systems deployed in hospitals are built entirely by human experts. Although these systems are time-consuming and costly to build, they typically do not cover all specialties of the care. Statistical anomaly detection approach for error detection, proposed and studied in this work, relies solely on data that are extracted from existing patient record repositories and no or very limited expert input is required. This reduces the cost of the approach and its deployment.

Typical anomaly detection methods used in data analysis are unconditional and look for outliers with respect to all data attributes. In medical domain these methods would identify unusual patients, that is, patients suffering from a less frequent disease or patients with unusual collection of symptoms. Unfortunately, this does not fit the nature of the prob-

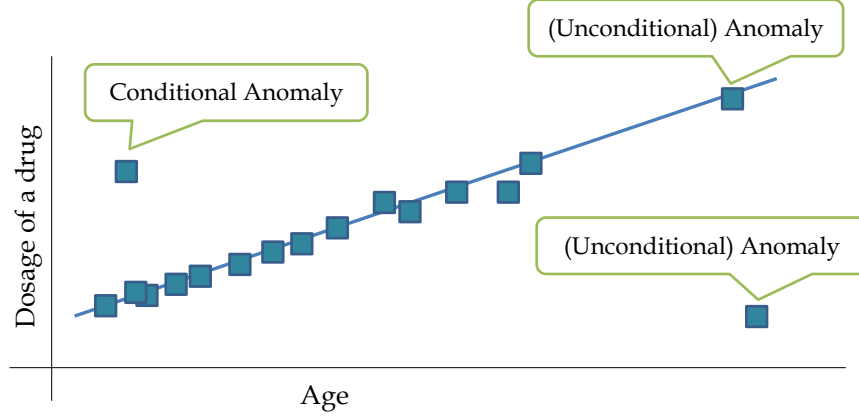


Figure 1: Conditional vs. unconditional anomalies

1

lem we want to solve in error detection; the identification of unusual patient management decisions with respect to past patients who suffer from the same or similar condition. To address this, we are developing a *qualitatively new conditional anomaly detection framework* where the decision event is judged anomalous with respect to patient’s symptoms, state and demographics.

Figure 1 illustrates the concept of a conditional anomaly: The dosage of a drug is shown here as a linear function of the age. Now imagine that we have a young patient that was given a higher dosage of a drug (Figure 1, top left). The amount of dosage is not unusual at all: Indeed, we have other patients with same dosage. What is unusual, it this anomalous dosage with respect to his age: As we can see, the patients that have similar age were given lower dosages. We can say that this dosage was *conditionally anomalous* given patient’s age.

Electronic health records (EHRs) are a necessary requirement for the successful deployment of the methods described here. With the increasing number of medical groups adopting EHR systems [Gans et al., 2005], more people will benefit from reduced medical errors. We imagine that the inclusion of our method into the existing EHR systems will require no extra time from physicians. Our anomaly detection framework serves more as background monitoring system that raises alerts when the confidence of an anomaly is high.

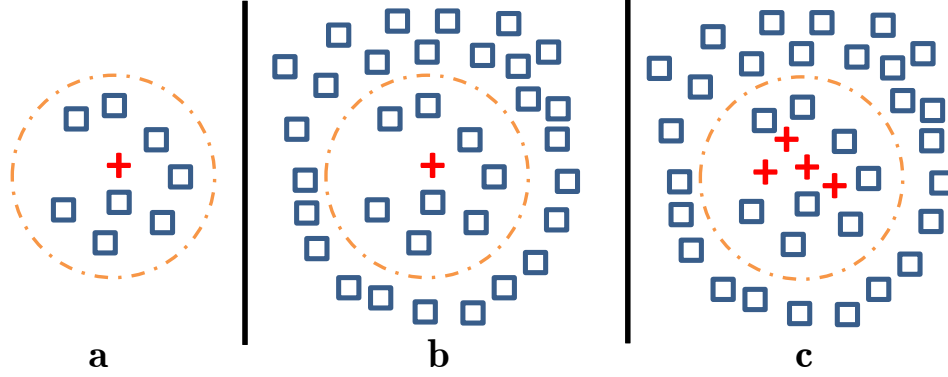


Figure 2: Disadvantages of nearest neighbor approach for conditional anomaly detection

Moreover, conditional anomaly detection can be very useful for detecting unusual behaviors, outcomes, or unusual attribute pairings in many other domains [Das et al., 2008]. Examples of such problems are the detection of unusual actions or outcomes in medicine [Hauskrecht et al., 2007], investments [Rubin et al., 2005], law [Aktolga et al., 2010], social networks [Heard et al., 2010], politics [Kolar et al., 2010] and other fields [Das et al., 2008]. In all these domains, the outcome strongly depends on the context (patient conditions, economy and market, case circumstances, etc.), hence the outcome is unusual only if it is compared to the examples with the same context.

Throughout this thesis, we build on label propagation on a data similarity graph. Unlike local neighborhood methods based on the nearest neighbors, it respects the structure of the manifold and lets us account for more complex interactions in the data. Figure 2 illustrates a potential benefit of label propagation, where the goal is to detect that the positive (+) example has an anomalous label, conditioned on its placement. The positive (+) label in 2b is more anomalous than the one in 2a, but nearest neighbor (NN) would consider them equal (in only considers the points within the displayed circle). Moreover, NN approach would find clustered (+) anomalies in 2c normal because it ignores the data beyond the nearest neighbors.

1.2 THESIS STATEMENT AND MAIN CONTRIBUTIONS

The problem of conditional anomaly detection could be approached by

1. extending one-class (unconditional) anomaly methods (Section 3.1.4)
2. classification and claiming misclassified examples as conditionally anomalous (Section 3.1.5)

Both of these approaches suffer from problems described in Section 3.1. In this thesis we develop the methodology to address these problems. We take a graph-based approach, because it is non-parametric, incorporates the manifold assumption and can also easily take advantage of unlabeled data. We present the following main contributions:

- We introduce a non-parametric **graph-based** methods and show how they can *handle unconditional outliers* (Section 3.1.6).
- We show how a **soft harmonic solution** on data similarity graphs can be used for conditional anomaly detection (Section 3.1.6.2).
- We propose the **online harmonic function** solution and show how to compute its approximation efficiently (Section 3.3).
- We show how to combine max-margin and semi-supervised learning to **max-margin graph cuts** semi-supervised learning (Section 3.2).
- We prove **performance bounds** for our online algorithm in a semi-supervised setting on quantized graphs (Section 4.4).
- We show how to compute label propagation on a graph and the centroids of a backbone graph **jointly**. (Section 3.2.2)
- We conduct a **human experts' evaluation** with a panel of physicians of detected conditional anomalies and show the benefits of our methods (Section 5.4.2.1).

Based on the aforementioned contributions, we claim the following:

Our graph-based methods can detect conditional and filter unconditional anomalies.

1.3 ORGANIZATION OF THE THESIS

- In Chapter 2, we outline the necessary background and related work in anomaly detection (Section 2.1), semi-supervised learning (Section 2.4), and online quantization (Section 2.3).
- Chapter 3 presents the novel methods for conditional anomaly detection (Section 3.1.5.1 and Section 3.1.6.2), semi-supervised learning (Section 3.2) and the online semi-supervised learning (Section 3.3).
- Chapter 4 presents the theoretical analysis of the methods from Chapter 3. In particular, the analysis of max-margin graph cuts (Section 4.2) and the analysis of the online semi-supervised learning on quantized graphs (Section 4.4).
- Chapter 5 presents the experimental results on various synthetic and real-world datasets, notably the medical datasets from University of Pittsburgh Medical Center.

The parts of this thesis previously appeared in [Hauskrecht et al., 2007, Hauskrecht et al., 2010, Valko et al., 2008, Valko and Hauskrecht, 2008, Valko and Hauskrecht, 2010, Valko et al., 2010, Valko et al., 2011, Kveton et al., 2010b, Kveton et al., 2010a].

2.0 BACKGROUND AND RELATED WORK

2.1 RELATED WORK IN ANOMALY DETECTION (AD)

2.1.1 Uncoditional Anomaly Detection

In this section we review previous approaches for traditional anomaly detection. A comprehensive review of many anomaly detection approaches can be found in [Markou and Singh, 2003a] and [Markou and Singh, 2003b].

[Scholkopf et al., 1999] proposed one-class SVM, that only needs positive (or non-anomalous) examples to learn the margin. The idea is that the space origin (zero) is treated as the only example of the 'negative' class. In that way, the learning essentially estimates the support of the distribution. The data, that do not fall into this support, have negative projections and can be considered anomalous.

[Eskin, 2000] assumes that the number of anomalies is significantly lower than the number of normal cases. Author defines a distribution for the data as a mixture of majority (M) and anomalous distribution(A): $D = (1 - \lambda)M + \lambda A$. He then iteratively partitions the dataset into the majority set M_t and the anomalous set A_t . At the beginning $A_0 = \emptyset, M_0 = D$. At each step t , it is determined whether the case x_t is an anomaly. x_t is considered anomalous, if its displacement to the anomaly set ($M_t = M_{t-1} \setminus \{x\}$ and $A_t = A_{t-1} \cup \{x\}$) increases the log-likelihood LL_{t-1} of the dataset by a predefined threshold c . If $LL_t - LL_{t-1} \leq c$, x_t remains marked as a normal case ($M_t = M_{t-1}$ and $A_t = A_{t-1}$). At the end, we get the final partition of D into a normal and an anomalous set.

The curse of high dimensionality is of concern in [Aggarwal and Yu, 2001]. Authors search for the abnormal lower dimensional projections, by dividing each attribute into the

equi-depth (the same range of f cases) ranges. Assuming statistical independence, each k -dimensional sub-cube in this grid should contain the fraction of f^k of total cases. Authors then search for k -dimensional sub-cubes, where presence of points is significantly lower than expected. As the brute force search for projections is computationally infeasible, authors use genetic algorithms to perform the search.

In [Breunig et al., 2000], authors expand k -distance (distance to the k nearest neighbor) to get the so-called reachability distance for the object O wrt. to p as $\text{reach_dist}(O, p) = \max(k_distance(p), \text{dist}(O, p))$. Using this smoothed distance, they define *local outlier factor* (LOF), which expresses the degree of considered object being outlier with respect to its neighborhood. LOF depends of $MinPts$, the number of nearest points to define local neighborhood. Although this is data-dependent, authors propose to calculate the maximum LOF for $MinPts$ within reasonable range (which was 30–50 in their experiments) and threshold this value. The bigger the LOF the more anomalous the object is. Authors give bounds for LOF and prove there are tight for important cases. For example, LOF is close to one for objects within the clusters. A useful property of LOFs is that it works well with cluster of different densities.

[Lazarevic and Kumar, 2005] applies bagging approach to improve the performance of local anomaly detectors. In every round of the algorithm a subset of features is selected and a local anomaly detector (such as LOF (LOF [Breunig et al., 2000]) is applied. Every round produces a scoring of all data which is at the end combined to the final score using either breadth-first or cumulative-sum approach.

[Syed and Rubinfeld, 2010] use minimum enclosing ball approach to detect anomalies in the clinical data similar to data that we use in this work. The authors learn minimum volume hypersphere that encloses the data for all patients. The anomaly score is defined as the distance from the center. They showed that this unsupervised approach performed at par with the supervised approaches with prelabeled examples.

2.1.1.1 Graph-based methods [Akoglu et al., 2010] performs anomaly detection on weighted graphs when nodes do not follow discovered power laws between number of neighbors and the properties of the local neighborhood subgraph (total number of edges, total

weight and the principal eigenvalue of the weighted adjacency graph). The outlier score is defined as a distance to the fitting line. To account for the points that fit the line but are far away from all other examples, the authors combine their methods with a density based method, such as LOF [Breunig et al., 2000].

[He et al., 2007] is a semi-supervised method that propagates the labels until a heuristic stopping criterion. Moreover, it used unlabeled data to better estimate the prior in the case that the empirical distribution is skewed from the true distribution.

[Moonesignhe and Tan, 2006] use random walks to detect outliers. They build their weight matrix either by cosine similarity or by a number of shared neighbors after thresholded cosine similarity. Anomalous nodes are identified as those with low connectivity. Connectivity is calculated using Markov chain using weights/similarities as a transition matrix. Starting from the uniform connectivity assigned at the step 0, connectivity is spread according to similarity matrix until convergence.

2.1.1.2 Approaches with prelabeled anomalies [Chawla et al., 2003] combine boosting scheme with SMOTE (Synthetic Minority Over-sampling TEchnique). They do that in every iteration of smoothing. For continuous data, SMOTE generates new sample by sampling a data point and one of its k nearest neighbors and taking a random point on segment between them in the space. For discrete data, a new point is created as a majority vote of the k nearest neighbors for each feature. The authors show improvement with this method over just smoothing, just SMOTE and applying SMOTE once before the boosting for a minority class. The SMOTEboost approach generally improves recall and does not cause significant degradation in precision, thus improving the F-measure.

[Ma and Perkins, 2003] use support vector regression to learn the underlying temporal model (time event is modeled as a linear regression function of the previous events). A surprise is defined as the value outside the tolerance range. Given the fixed length of the event, a probability of number of surprises actually happening is calculated. When that is too small, an anomaly is declared.

2.1.1.3 Rare Category Detection [Pelleg and Moore, 2005] aims to detect rare category which presumably correspond to the interesting anomalies in a pool-based active learning framework. After a human expert labels some examples, the Gaussian mixture is fit to data. Different hinting heuristics are then used to propose the new examples to be label by the expert. Authors propose *interleave* heuristics which takes one example per mixture a time with low fit probability, not taking to account any mixture weight. This hinting heuristic appears to be superior to the low-likelihood one (suggesting examples with the overall low fit probability) and ambiguous one (suggesting examples with uncertain class membership).

[He and Carbonell, 2008] attempts to detect rare categories in the data, assuming that examples from the rare category are self-similar, tightly grouped and we have some knowledge about the class priors. Nearest neighbor based statistic is used to actively sample points corresponding to points with the maximum change in the local density.

2.1.2 Conditional Anomaly Detection (CAD)

We start with a short summary of our work. In [Hauskrecht et al., 2007], we introduced the concept of the conditional anomaly detection (CAD) and showed its potential for the medical records. For each case, we took its nearest neighbors and learned a Bayesian belief network (BBN) or a naïve Bayes model (NB) from them. The cases with low class-conditional probabilities were deemed anomalous. We discovered that while for BBN it was better to use all the cases for learning, for a more restricted NB a small neighborhood was beneficial. The main problem with learning structure of BBN is that it does not scale beyond couple dozens of features. In [Valko and Hauskrecht, 2008], we showed the benefit of distance metric learning for the selection of closest cases. We also used the softmax model [McCullagh and Nelder, 1989] to calculate the class-conditional probability of a probabilistic one nearest neighbor (similarly as in [Goldberger et al., 2004]) for this purpose. In [Valko et al., 2008], we introduced a new anomaly measure based on the distance from the hyperplane learned by SVM [Vapnik, 1995] and performed the initial experiments on the PCP (Section 5.1.1) dataset. We later conducted an extensive human evaluation study with the panel of 15 physicians in [Hauskrecht et al., 2010]. Aside from our work which will be reviewed in more

detail in later chapters, we also describe other early work along these lines.

[Valizadegan and Tan, 2007] used kernel based weighted nearest neighbor approach to jointly estimate probabilities of the examples being mislabeled. The joint estimation is posed as an optimization problem and solved with Newton methods. A regularization is needed to avoid one of the classes being completely deemed to be mislabeled.

In [Song et al., 2007], user defines a partitioning of the features into two groups: the **indicator** features – those that can be directly indicative of an anomaly and the **environmental** features, which cannot, but can influence the indicator ones. The indicator (y) and the environmental (x) variables are modeled separately both as the mixtures of multivariate Gaussians ($y \sim U$ and $x \sim V$). A mapping function is defined between those mixtures as a probability of choosing a Gaussian for an indicator variable given an environmental one $p(V_j|U_i)$. The authors assume following generative process for a datapoint $\langle x, y \rangle$: If x is a sample from U_i then a die is tossed, according to $p(V_j|U_i)$, to determine which Gaussian from V will produce y and subsequently y is produced. Since it is not known, which U_i was x sampled from, likelihood of $f_{CAD}(y|\Theta, x)$ is computed as a weighted sum over Gaussians U_i . Model is learned via EM, either directly – optimizing all parameters at once (named as DIRECT), optimizing first parameters for Gaussians and then for the mapping function (FULL) or optimizing the indicator Gaussians, the environmental Gaussians and the mapping function separately (SPLIT).

The work on cross-outlier detection [Papadimitriou and Faloutsos, 2003] is also related to CAD. Papadimitriou and Faloutsos [Papadimitriou and Faloutsos, 2003] defined the notion of the cross-outliers as examples that seem normal when considered the distribution of examples from the assigned class, but are abnormal when considering the samples from the other class. For each sample (\mathbf{x}, y) , they compute two statistics based on the similarity of \mathbf{x} to its neighborhood from the samples belonging to class y and samples not belonging to class y . An example is considered anomalous if the first statistic is significantly smaller than the second statistic. Unfortunately, the method is not very robust to fringe points (Figure 4) [Papadimitriou and Faloutsos, 2003].

[Das, 2009] in his thesis aims to detect several kinds of individual and group anomalies. The methods relevant to this work are *conditional* and *marginal* methods for individual

record anomalies, ignoring rare values. For the data t and the subsets of attributes (A, B, C) he computes the ratios of the form $\frac{P(A, B)}{P(A)P(B)}$ for the marginal and $\frac{P(A, B|C)}{P(A|C)P(B|C)}$ for the conditional method. The goal is to find unusual occurrences of the attribute values. The records that have those ratios low are considered anomalous. The normalization of the joint probabilities by the marginal probabilities takes care of rare records, because those have also small marginals. The thesis describes several speedups to compute the ratios for exponentially many subgroup to allow the methods to scale up.

2.2 BACKGROUND IN GRAPH THEORY

2.2.1 Graph as a data model

Many of the methods presented here are based on a graph representation of the data. Having some data, we create a undirected weighted graph $G = (V, E)$ with set of vertices V and set of edges E , associating every data point with a graph vertex. Next, we define a non-negative weight function $V \times V \rightarrow \mathbb{R}$ such that $w_{ij} = w_{ji}$. In the case that $\{i, j\} \notin E(G)$, $w_{ij} = 0$. Let the similarity matrix $W = \{w_{ij}\}$ denote a matrix of all edge weights which encode how similar are the vertices to each other. We define degree d_i of the vertex i as the sum of all edges coinciding with i :

$$d_i = \sum_j w_{ij}$$

and the diagonal matrix D with $D_{ii} = d_i$. Let volume $\text{vol}(G)$ of graph G be the sum of all its weights:

$$\text{vol}(G) = \text{vol}(W) = \sum_i d_i = \sum_{i,j} w_{ij}$$

Now, let us define an unnormalized graph Laplacian L as

$$L(G) = L(W) = D - W$$

and the symmetric normalized graph Laplacian as

$$L_{\text{sym}}(G) = L_{\text{sym}}(W) = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

It can be easily shown that for any $\mathbf{h} \in \mathbb{R}^n$:

$$\mathbf{h}^\top L \mathbf{h} = \frac{1}{2} \sum_{ij} w_{ij} (h_i - h_j)^2.$$

2.2.1.1 Stationary distribution of a random walk We now describe a way to compute a stationary distribution of a (non-absorbing) random walk on the data similarity graph in a closed form. Let us define the random walk as follows. In every step of a random walk, we jump from a node to its neighbors proportionally to their mutual weight:

$$P(\mathbf{x}_i \rightarrow \mathbf{x}_j) = \frac{W_{ij}}{\sum_{j'} W_{ij'}}$$

Let D be the diagonal matrix with the sum of weights W on the diagonal: $D_{ii} = \sum_{j'} W_{ij'}$ for all i . Then the transition matrix of just described random walk is $P = D^{-1}W$. The approximation we use here is that we estimate the class conditional probability with the proportion of the time that this random walk spends in the evaluated example [Lee and Wasserman, 2010]. We can calculate this proportion from the stationary distribution of this random walk [Chung, 1997]. Let s be the row vector of the stationary distribution of a random walk with transition matrix P . For a stationary distribution s it has to hold that $sP = s$. Note that $\mathbf{1}D = \mathbf{1}W$, where $\mathbf{1}$ is all one row vector. It is easy to verify that

$$s = \frac{\mathbf{1}W}{\text{vol}(W)} \tag{2.1}$$

satisfies the definition:

$$sP = \frac{\mathbf{1}WP}{\text{vol}(W)} = \frac{\mathbf{1}DP}{\text{vol}(W)} = \frac{\mathbf{1}DD^{-1}W}{\text{vol}(W)} = \frac{\mathbf{1}W}{\text{vol}(W)} = s$$

The equation (2.1) enables us to compute the stationary distribution in a closed form.

2.2.2 Regularized Harmonic Function

In this section, we build on the harmonic solution [Zhu et al., 2003]. Moreover, we show how to regularize it such that it can interpolate between semi-supervised learning (SSL) on labeled examples and SSL on all data. A standard approach to SSL on graphs is to minimize the quadratic objective function

$$\begin{aligned} \min_{\ell \in \mathbb{R}^n} \quad & \ell^\top L \ell \\ \text{s.t.} \quad & \ell_i = y_i \text{ for all } i \in l; \end{aligned} \tag{2.2}$$

where ℓ denotes the vector of predictions. This problem has a closed-form solution:

$$\ell_u = (D_{uu} - W_{uu})^{-1} W_{ul} \ell_l,$$

which satisfies the *harmonic property* $\ell_i = \frac{1}{d_i} \sum_{j \sim i} w_{ij} \ell_j$ ($i \sim j$ denotes that i neighbours j), and therefore is commonly known as the *harmonic solution*.

Since the solution can be also computed as:

$$\ell_u = (I - P_{uu})^{-1} P_{ul} \ell_l,$$

it can be viewed as a product of a random walk on the graph W with the transition matrix $P = D^{-1}W$. The probability of moving between two arbitrary vertices i and j is w_{ij}/d_i , and the walk terminates when the reached vertex is labeled. Therefore, harmonic solution is a form of *label propagation* on the data similarity graph. Each element of the solution is given by:

$$\begin{aligned} \ell_i &= (I - P_{uu})_{iu}^{-1} P_{ul} \ell_l \\ &= \underbrace{\sum_{j:y_j=1} (I - P_{uu})_{iu}^{-1} P_{uj}}_{p_i^{(+1)}} - \underbrace{\sum_{j:y_j=-1} (I - P_{uu})_{iu}^{-1} P_{uj}}_{p_i^{(-1)}} \\ &= p_i^{(+1)} - p_i^{(-1)}, \end{aligned}$$

where p_i^{+1} and p_i^{-1} are probabilities by which the walk starting from the vertex i ends at vertices with labels $+1$ and -1 , respectively. Therefore, when ℓ_i is rewritten as $|\ell_i| \text{sgn}(\ell_i)$,

$|\ell_i|$ can be interpreted as a *confidence* of assigning the label $\text{sgn}(\ell_i)$ to the vertex i . The maximum value of $|\ell_i|$ is 1, and it is achieved when either $p_i^{+1} = 1$ or $p_i^{-1} = 1$. The closer the confidence $|\ell_i|$ to 0, the closer are the probabilities p_i^{+1} and p_i^{-1} to 0.5, and the more *uncertain* is the label $\text{sgn}(\ell_i)$.

We propose to control the confidence of labeling by regularizing the Laplacian L as $L + \gamma_g I$, where γ_g is a scalar and I is the identity matrix. Similarly to (2.2), the corresponding problem

$$\begin{aligned} \min_{\ell \in \mathbb{R}^n} \quad & \ell^\top (L + \gamma_g I) \ell \\ \text{s.t.} \quad & \ell_i = y_i \text{ for all } i \in l; \end{aligned} \tag{2.3}$$

can be computed in a closed form

$$\ell_u = (L_{uu} + \gamma_g I)^{-1} W_{ul} \ell_l. \tag{2.4}$$

and we will refer to it as *regularized HS*. It can be also interpreted as a random walk on the graph W with an extra sink. At every step, a walk at node x_i may terminate at the sink with probability $\gamma_g / (d_i + \gamma_g)$ where d_i is the degree of the current node in the walk. Therefore, the scalar γ_g essentially controls how the “confidence” $|\ell_i|$ of labeling unlabeled vertices decreases with the number of hops from labeled vertices. The proposed regularization will essentially drive the confidence of distant vertices to zero.

2.2.2.1 Soft harmonic solution A related problem to (2.2) is the one when the constraints representing the fit to the data are enforced in a soft manner [Cortes et al., 2008]. One way of achieving this is by solving a following problem:

$$\ell^\star = \min_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top C (\ell - \mathbf{y}) + \ell^\top K \ell, \tag{2.5}$$

where $K = L + \gamma_g I$ is the regularized Laplacian of the similarity graph, C is a diagonal matrix such that $C_{ii} = c_l$ for all labeled examples, and $C_{ii} = c_u$ otherwise, and \mathbf{y} is a vector of pseudo-targets such that y_i is the label of the i -th example when the example is labeled,

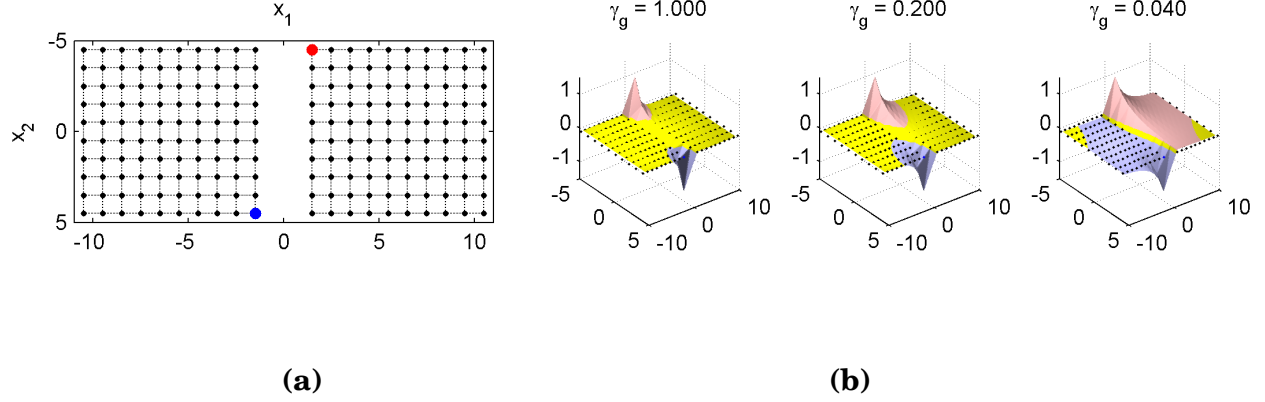


Figure 3: **a.** Similarity graph **b.** Three regularized harmonic solutions

and $y_i = 0$ otherwise. The appealing property of (2.5) is that its solution can be computed in closed form as follows [Cortes et al., 2008]:

$$\ell^* = (C^{-1}K + I)^{-1}\mathbf{y} \quad (2.6)$$

We will use soft harmonic solution (2.5) particularly in the theoretical analysis in Chapter 4.

Several examples of how γ_g affects the regularized solution are shown in Figure 3. Figure 3a shows an example of a simple data adjacency graph. The vertices of the graph are depicted as dots. The red and blue dots are labeled vertices. The edges of the graph are shown as dotted lines and weighted as $w_{ij} = \exp[-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2]$. Figure 3b. shows three regularized harmonic function solutions on the data adjacency graph from Figure 3a. The plots are cubic interpolations of the solutions. The pink and blue colors denote parts of the feature space \mathbf{x} where $\ell_i > 0$ and $\ell_i < 0$, respectively. The yellow color marks regions where the confidence $|\ell_i|$ is less than 0.05. When $\gamma_g = 0$, the solution turns into the ordinary harmonic function solution. When $\gamma_g = \infty$, the confidence of labeling unlabeled vertices decreases to zero. Finally, note that our regularization corresponds to increasing all eigenvalues of the Laplacian L by γ_g [Smola and Kondor, 2003]. In Section 4.2, we use this property to bound the generalization error of our solutions.

2.3 RELATED WORK IN ONLINE QUANTIZATION

Given n data points and a typical graph construction method, exact computation of HS has a space and time complexity of $\Omega(n^2)$ in general due to the construction of an $n \times n$ similarity matrix. Furthermore, exact computation requires an inverse operation on an $n \times n$ similarity matrix which takes $O(n^3)$ in most practical implementations¹. For applications with large data size (e.g., exceeding thousands), the exact computation or even storage of HS becomes infeasible, and problems with n in the millions are entirely out of reach.

An influential line of work in the related area of graph partitioning approaches the computation problem by reducing the size of the graph, collapsing vertices and edges, partitioning the smaller graph, and then uncoarsening to construct a partition for the original graph [Hendrickson and Leland, 1995, Karypis and Kumar, 1999]. Our work is similar in spirit but provides a theoretical analysis for a particular kind of coarsening and uncoarsening methodology.

Our aim is to find an effective *data preprocessing* technique that reduces the size of the data and coarsens the graph [Madigan et al., 2002, Mitra et al., 2002]. There are two types of approaches widely used in practice for data preprocessing:

1. data quantization based approaches, which aim to replace the original data set with a small number of high quality “representative” points that capture relevant structure [Goldberg et al., 2008, Yan et al., 2009];
2. Nyström method based approaches, which aim to explore low-rank matrix approximations to speedup the matrix operations [Fowlkes et al., 2004]).

While it is useful to define such preprocessors, it is not satisfactory to simply reduce the size of similarity matrix to speed up the matrix calculations. so that the related matrix operation can be performed in a desired time frame.

What is needed is an explicit connection between the amount of data reduction that is achieved by a preprocessor and the subsequent effect on the classification error. Some widely used data preprocessing approaches are based on data quantization, which replaces

¹The complexity can be further improved to $\mathcal{O}(n_u^{2.376})$ by using the Coppersmith-Winograd algorithm.

the original data set with a small number of high quality centroids that capture relevant structure [Goldberg et al., 2008, Yan et al., 2009].

Such approaches are often heuristic and do not quantify the relationship between the noise induced by the quantization and the final prediction risk. An alternative approach to the computation problem is the *Nyström method*, a low rank matrix approximation method that allows faster computation of the inverse. This method has been widely adopted, particularly in the context of approximations for SVMs [Drineas and Mahoney, 2005, Williams and Seeger, 2001, Fine and Scheinberg, 2001] and spectral clustering [Fowlkes et al., 2004].

However, since the Nyström method uses interactions between subsampled points and *all* other data points, storage of all points is required and thus, it becomes unsuitable for infinitely streamed data. To our best knowledge, we are not aware of any online version of Nyström method that could process an unbounded amount of streamed data. In addition, in an offline setting, Nyström-based methods have inferior performance than the quantization-based methods, if both of them are given the same time budget for computation, which was shown in an early work on the spectral clustering [Yan et al., 2009], as well as a computational cost of $O(nk^2)$ where k is the rank of the approximation.

Using incremental k -centers [Charikar et al., 1997] which has provable worst case bound on the distortion, we quantify the error introduced by quantization. Moreover, using regularization we show that the solution is stable, which gives the desired generalization bounds.

An interesting method is introduced in [Aggarwal et al., 2003], which address context drift, or *evolution* in the data streams. Cluster can emerge and die based on approximated recency. But again this method is a heuristic and comes with no guarantees on the quality of the quantization.

2.4 RELATED WORK IN SEMI-SUPERVISED LEARNING (SSL)

[Zhu et al., 2003] extend their previous work [Zhu et al., 2003] to Gaussian processes by no longer assuming that soft labels are fixed to the observed data. Instead they assume the data generation process $\mathbf{x} \rightarrow \mathbf{y} \rightarrow \mathbf{t}$, where $\mathbf{y} \rightarrow \mathbf{t}$ is a noisy label generation with process

modeled by a sigmoid. The posterior is not Gaussian and the authors use Laplace approximation to compute $p(\mathbf{y}_L, \mathbf{y}_U | \mathbf{t}_L)$. They discuss using different kernels for the learning of graph weights, for example the tanh-weighted graph. And optimize it either by maximizing the likelihood of labeled data or maximizing the alignment to labeled data.

[Fergus et al., 2009] used the convergence of the eigenvectors of the normalized Laplacian to eigenfunctions of weighted Laplace-Beltrami operators to scale graph-based SSL to millions of examples. Assuming that the underlying distribution has a product form (which is a reasonable assumption after a PCA projection), they estimated the density using histograms for each dimension independently. Therefore, they only needed to solve d generalized eigenvector problems on the backbone graph, where d is the dimension of the data. Moreover, they only used the k smallest eigenvectors and subsequently needed to solve only one $k \times k$ least squares problem.

2.4.1 Semi-supervised Max-Margin Learning

Most of the existing work on semi-supervised max-margin learning can be viewed as manifold regularization of SVMs [Belkin et al., 2006] or semi-supervised SVMs with the hat loss on unlabeled data [Bennett and Demiriz, 1999]. The two approaches are reviewed in the rest of the section.

2.4.1.1 Semi-supervised SVMs Semi-supervised support vector machines with the *hat loss* $\hat{V}(f, \mathbf{x}) = \max\{1 - |f(\mathbf{x})|, 0\}$ on unlabeled data [Bennett and Demiriz, 1999]:

$$\min_f \sum_{i \in l} V(f, \mathbf{x}_i, y_i) + \gamma \|f\|_K^2 + \gamma_u \sum_{i \in u} \hat{V}(f, \mathbf{x}_i) \quad (2.7)$$

compute max-margin decision boundaries that avoid dense regions of data. The hat loss makes the optimization problem non-convex. As a result, it is hard to solve the problem optimally and most of the work in this field has focused on approximations. A comprehensive review of these methods was done by [Zhu, 2008].

In comparison to semi-supervised SVMs, learning of max-margin graph cuts (3.9) is a convex problem. The convexity is achieved by having a two-stage learning algorithm. First,

we infer labels of unlabeled examples using the regularized harmonic function solution, and then, we minimize the corresponding convex losses.

2.4.1.2 Manifold regularization of SVMs Manifold regularization of SVMs [Belkin et al., 2006]:

$$\min_{f \in \mathcal{H}_K} \sum_{i \in I} V(f, \mathbf{x}_i, y_i) + \gamma \|f\|_K^2 + \gamma_u \mathbf{f}^\top L \mathbf{f}, \quad (2.8)$$

where $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))$, computes max-margin decision boundaries that are smooth in the feature space. The smoothness is achieved by the minimization of the regularization term $\mathbf{f}^\top L \mathbf{f}$. Intuitively, when two examples are close on a manifold, the minimization of $\mathbf{f}^\top L \mathbf{f}$ leads to assigning the same label to both examples.

In some aspects, manifold regularization is similar to max-margin graph cuts. In particular, note that its objective (2.8) is similar to the regularized harmonic function solution (2.3). Both objectives involve regularization by a manifold, $\mathbf{f}^\top L \mathbf{f}$ and $\ell^\top L \ell$, regularization in the space of learned parameters, $\|f\|_K^2$ and $\ell^\top I \ell$, and some labeling constraints $V(f, \mathbf{x}_i, y_i)$ and $\ell_i = y_i$. Since max-margin graph cuts are learned conditionally on the harmonic function solution, the problems (3.9) and (2.8) may sometimes have similar solutions. A necessary condition is that the regularization terms in both objectives are weighted in the same proportions, for instance, by setting $\gamma_g = \gamma/\gamma_u$. We adopt this setting when manifold regularization of SVMs is compared to max-margin graph cuts in Section 5.3.1.

2.4.2 Online Semi-Supervised Learning

The online learning formulation of SSL is which is suitable for *adaptive* machine learning systems. In this setting, a few labeled examples are provided in advance and set the initial bias of the system while unlabeled examples are gathered online and update the bias continuously. In the online setting, learning is viewed as a repeated game against a potentially adversarial nature. At each step t of this game, we observe an example \mathbf{x}_t , and then predict its label \hat{y}_t . The challenge of the game is that after the game started we do not observe the true label y_t . Thus, if we want to adapt to changes in the environment, we have to rely on indirect forms of feedback, such as the structure of data.

Despite the usefulness of this paradigm for practical adaptive algorithms [Grabner et al., 2008, Goldberg et al., 2008], there is not much success in applying this paradigm to realistic problems, especially when data arrive at a high rate such as in video applications. [Grabner et al., 2008] applies online semi-supervised boosting to object tracking, but uses a heuristic method to greedily label the unlabeled examples. This method learns a binary classifier, where one of the classes explicitly models outliers. In comparison, our approach is multi-class and allows for implicit modeling of outliers. The two algorithms are compared empirically in Section 5.3.3. [Goldberg et al., 2008] develop an online version of manifold regularization of SVMs. Their method learns max-margin decision boundaries, which are additionally regularized by the manifold. Unfortunately, the approach was never applied to a naturally online learning problem, such as adaptive face recognition. Moreover, while the method is sound in principle, no theoretical guarantees are provided.

3.0 METHODS

3.1 CONDITONAL ANOMALY DETECTION

3.1.1 Introduction to Conditonal Anomaly Detection

Anomaly detection is the task of finding unusual elements in a set of observations. Most existing anomaly detection methods in data analysis are unconditional and look for outliers with respect to all data attributes [Breunig et al., 2000, Akoglu et al., 2010, Markou and Singh, 2003a, Markou and Singh, 2003b, Chandola et al., 2009]. *Conditional anomaly detection* (CAD) [Chandola et al., 2009] is the problem of detecting unusual values for a subset of variables given the values of the remaining variables. In other words, one set of variables defines the context in which the other set is examined for anomalous values.

CAD can be extremely useful for detecting unusual behaviors, outcomes, or unusual attribute pairings in many domains [Das et al., 2008]. Examples of such problems are the detection of unusual actions or outcomes in medicine [Hauskrecht et al., 2007], investments [Rubin et al., 2005], law [Aktolga et al., 2010], social networks [Heard et al., 2010], politics [Kolar et al., 2010] and other fields [Das et al., 2008]. In all these domains, the outcome strongly depends on the context (patient conditions, economy and market, case circumstances, etc.), hence the outcome is unusual only if it is compared to the examples with the same context.

In this work, we study a special case of CAD that tries to identify the unusual values for just one target variable given the values of the remaining variables (attributes). The target variable is assumed to take on a finite set of values which we also refer to as labels, because of its similarity to the classification problems. Therefore, we refer to conditional

anomalies as mislabelings [Valizadegan and Tan, 2007] or cross-outliers [Papadimitriou and Faloutsos, 2003]. Our objective is to develop robust conditional anomaly methods that work well for high-dimensional datasets and let us capture various non-linearities in the underlying space. This work is motivated primarily by clinical and biomedical datasets and applications. These datasets are highly heterogeneous, and may include hundreds of lab results of different nature, medications and procedures performed during hospital stay. In general, the distributions are multi-modal, reflecting many different patients' conditions [Hauskrecht et al., 2010].

3.1.2 Definition of Conditional Anomaly

In general, the concept of (conditional) anomaly in data in the existing literature is somewhat ambiguous and several definitions has been proposed in the past [Markou and Singh, 2003a, Markou and Singh, 2003b]. Typically, an example is considered anomalous when it is not expected from some underlying model. A number of anomaly detection methods have been developed for this purpose (Section 2.1.1). Conditional anomaly detection (CAD) problem (Section 2.1.2) is different but equally useful in practice. It seeks to detect unusual values for a subset of variables \mathcal{Y} given the values for the remaining variables \mathcal{X} . Since in this thesis we focus on CAD on one variable, we provide the definition for this case only.

Intuitively, we can define a conditional anomaly as follows: Given a set of n past observed examples $(\mathbf{x}_i, y_i)_{i=1}^n$ (with possible label noise), *conditional anomaly* is any instance i among recent m examples $(\mathbf{x}_i, y_i)_{i=n+1}^{n+m}$ for which y_i is unusual. In this statement, we assume that the past observed examples $(\mathbf{x}_i, y_i)_{i=1}^n$ are given. We do not assume that their labels are perfect, they may also be subject to the label noise.

Let us motive a formal definitional of conditional anomaly by assuming that the y_i is a continuous variable and has a standard normal distribution:

$$y_i | \mathbf{x}_i \sim \mathcal{N}(0, 1).$$

As the standard normal distribution is an unimodal distribution with zero mean, the most anomalous values are the ones with the largest absolute value. Assuming a random sample

of the size n , $Y^{(n)} = y_1, y_2, \dots, y_n$, the extremal values for this distributions correspond to the first and the n -th order statistic. The expected n -th order statistic for the standard normal can be approximated as $n \rightarrow \infty$ as [Cramér, 1999]:

$$Y_{(n)}^{(n)} \approx \sqrt{2 \ln n} \quad (3.1)$$

Therefore, the more samples we have, the larger extremal value we are likely to see and the less we should be surprised by them. This motivates our definition which assumes some probabilistic model of data (not necessarily normal) and depends on the sample size n :

Definition 1. *Given any probabilistic model P and a random sample $(\mathbf{x}_i, y_i)_{i=1}^n$, a **conditional anomaly** of the c -level in the value y_i given \mathbf{x}_i is any instance i , such that $P(y_i | \mathbf{x}_i) = O(e^{-cn})$.*

It is not common that we would have an access to such model or that we would be able to estimate the class conditional probabilities reliably (especially in high dimensions). Therefore, in practice may need to assess the anomalies otherwise (eg. using human experts).

Not knowing the underlying model, that generates the (attribute, label) pairs, may lead to two major complications illustrated in Figure 4. First, a given instance may be far from the past observed data points (e.g. patient cases). Because of the lack of the support for alternative responses, it is difficult to assess the anomalousness of these instances. We refer to these instances as *isolated points*. Second, the examples on the boundary of the class distribution support may look anomalous due to their low likelihood. These boundary examples are also known as *fringe points* [Papadimitriou and Faloutsos, 2003]. Both of those we aim to avoid when looking for conditional anomalies.

3.1.3 Relationship to Mislabeling Detection

The work on CAD, when the target variable is restricted to discrete values only, is closely related to the problem of mislabeling detection [Brodley and Friedl, 1999]. The objective in this line of work to 1) to make a yes/no decision on whether the examples are mislabeled, and 2) to improve the classification accuracy by removing the mislabeled examples. Jiang et al. [Jiang and Zhou., 2004] used an ensemble of neural nets to remove suspicious sam-

ples to create a k -NN classifier. Sanchez et al. [Sanchez et al., 2003] introduced several k -NN based approaches including *Depuration*, *Nearest Centroid Neighborhood* (NCN) and *Iterative k -NCN*. Brodley et al. [Brodley and Friedl, 1999] tried different approaches to remove mislabeled samples including single and ensemble classifiers. Bagging and boosting are applied in [Verbaeten and Assche., 2003] to detect and remove mislabeled examples. Valizadegan and Tan [Valizadegan and Tan, 2007] introduced an objective function based on the weighted nearest neighbor approach to identify the mislabeled examples and solved it with Newton method.

While the objective of mislabeling detection research is to improve the classification accuracy by removing or correcting mislabeled examples, the objective of CAD is different: CAD is interested in ranking examples according to the severity of conditional anomalies in data. This is the main reason our evaluations of CAD in Chapter 5 measure the rankings of the cases being anomalous and not the improved classification accuracy when we remove them. Nevertheless, we do compare (Section 5.4) to the methods typically used in mislabeling detection.

There are various solutions to implement the conditional anomaly detection. We continue by outlining two baseline approaches.

3.1.4 Class-outlier approach

The simplest approach is to use one of the unconditional anomaly detection methods: For every possible class value y we learn a separate anomaly detection model M_y using only the values of \mathbf{x} attributes in the data. An example (\mathbf{x}_i, y_i) is anomalous if \mathbf{x}_i is anomalous in $M_{y=y_i}$. We refer to this approach as the *class-outlier* approach. The anomaly detection model M_y can be implemented with any unconditional anomaly detection model, such as 1-class SVM [Scholkopf et al., 1999], local outlier factor [Breunig et al., 2000] and many others [Chandola et al., 2009, Markou and Singh, 2003a, Markou and Singh, 2003b].

The class-outlier approach comes with some limitations. Such an approach detects anomalies with respect to its class label and ignores the examples from the other classes. This may not work well for those examples for which \mathbf{x} is away from all of the classes and

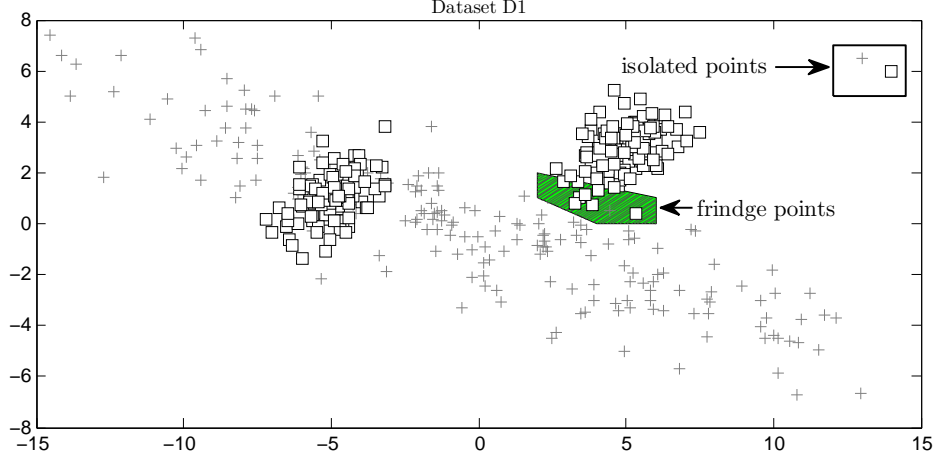


Figure 4: Challenges for CAD: 1) **fringe** and 2) **isolated** points

hence \mathbf{x} is an anomaly itself. To illustrate this, let us assume we have two classes (-1 and $+1$) and an example $(\mathbf{x}, -1)$ such that \mathbf{x} is an anomaly in $M_{y=-1}$. The class-outlier approach compares this example to all examples with the same label (-1) and declares it to be an anomaly. However, the problem is when \mathbf{x} is also an anomaly with respect to $M_{y=+1}$. In such a case it is unclear whether y should be -1 or $+1$ and hence the conclusion stating that $(\mathbf{x}, -1)$ is a conditional anomaly may be incorrect.

The other problem with class-outlier approach is that those methods often declare *fringe* points (Figure 4) as anomalies. Fringe points [Papadimitriou and Faloutsos, 2003] are points on the outer boundary of a distribution support for a specific class.

3.1.5 Discriminative approach

Another approach to detect conditional anomalies is to estimate the posterior $P(y_i|\mathbf{x}_i)$ for the observed example (\mathbf{x}_i, y_i) and use the posterior to measure how anomalous the data example is [Song et al., 2007, Hauskrecht et al., 2007, Hauskrecht et al., 2010, Valko et al., 2008]. According to Definition 1, an example is conditionally anomalous if the probability of the opposite label for this example is high. Various classification machine learning models

can be used to estimate the posterior from the past data. For example, one can use the logistic regression model or generative probabilistic models such as probabilistic graphical models that come with an immediate probabilistic interpretation. However, the output of other classification models, such as SVM, can be modified and transformed to produce a probabilistic output. For example, for the non-parametric Parzen window, the posterior probability can be estimated by summing the kernel weights for all examples with the same class label and by normalizing with the sum of weights for all examples.

$$P(y = y_i | \mathbf{x}_i) = \frac{\sum_{y_j = y_i} K(\mathbf{x}_j, \mathbf{x}_i)}{\sum_j K(\mathbf{x}_j, \mathbf{x}_i)}$$

We will assume $y \in \{-1, +1\}$ from now on, but the generalization to the multi-class case is straightforward. Without loss of generality, we assume that the testing example \mathbf{x}_i has $y_i = +1$. We want to compute $P(y_i \neq +1 | \mathbf{x}_i)$ to see whether this quantity is not too high, which would mean that y_i is conditionally anomalous given \mathbf{x}_i . Using Bayes theorem we get:

$$P(y \neq +1 | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y = -1)P(y = -1)}{\sum_{c \in \{-1, +1\}} P(\mathbf{x}_i | y = c)P(y = c)} \quad (3.2)$$

Since we model both prior and class-conditional density, this is a generative model.

3.1.5.1 CAD with Random Walks The following method is an example of the discriminative approach. Let (\mathbf{x}_i, y_i) be the new example that we want to evaluate and $P(\mathbf{x}_i | y = +1)$ and $P(\mathbf{x}_i | y = -1)$ the probabilities we want to compute for (3.2). In this part we show how we can estimate $P(\mathbf{x}_i | y)$ from the similarity graphs constructed separately for each class. A similarity graph for a set of examples is built by assigning each example to a node in the graph. The edges between the nodes and their weights represent the similarities between the examples

To explain our method, let us consider (again, without the loss of generality) the problem of estimating $P(\mathbf{x}_i | y = +1)$. First, we take all \mathbf{x}_i from the training set such that $y_i = +1$ and form a similarity graph using these examples. We then add the new example \mathbf{x}_i pretending that its label is $y = +1$. Let $G_{y=+1}$ be the graph we get (Figure 5). In the following we describe how we can use the stationary distribution of a random walk on $G_{y=+1}$ and use the local connectivity as an *approximation* for a density estimate [Lee and Wasserman, 2010]

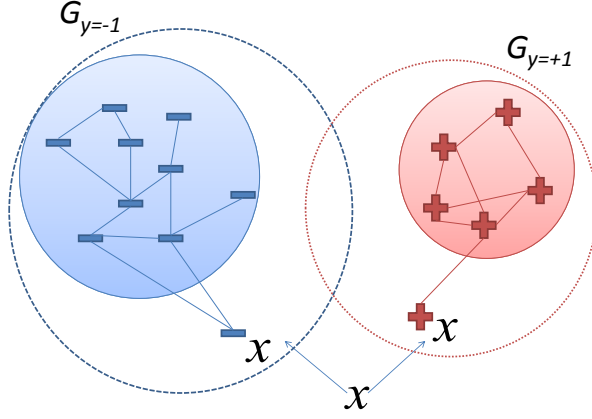


Figure 5: Estimating class-conditional probabilities from two similarity graphs

for $P(\mathbf{x}_i|y = +1)$. We do the same for $P(\mathbf{x}_i|y = -1)$ and plug the both estimates into (3.2) to get an estimate for $P(y \neq +1|\mathbf{x})$.

The equation (2.1) enables us to compute the stationary distribution in a closed form and ultimately allows us to compute (3.2) efficiently. Once we have the stationary distribution s of random walk on $G_{y=+1}$ we approximate $P(\mathbf{x}_i|y = +1)$ with s_i .

3.1.6 Regularized discriminative approach

We now describe how to avoid detecting the fringe and isolated points using regularization. Again, our approach considers both classes and y becomes an anomaly if its posterior probability given \mathbf{x} is small. We stress again that in this work we are not interested in isolated or fringe points. Let us consider the case of isolated points. Imagine the scenario that we get such an anomaly (\mathbf{x}_a, y_a) . If we take the approach we just described, \mathbf{x}_a will be far from $G_{y=c}$ for all c . Intuitively, the posterior (3.2) compares the weighted likelihoods of \mathbf{x}_a given the class, where weight is the class prior. If these likelihoods are estimated from the training data (and possibly from a small sample size), the estimates of $P(\mathbf{x}_a|y = -1)$ and $P(\mathbf{x}_a|y = +1)$ may become unreliable. Consequently, the relative difference between these likelihoods can

strongly favor one class. Our model would then become overly confident in that \mathbf{x}_a belongs to that class. We illustrate this behavior in Section 5.4.1.3.

To alleviate these problems, we propose a new discriminative approach that penalizes instances of \mathbf{x} that are anomalies themselves. We do it by regularizing the model as follows:

$$P(y \neq +1|\mathbf{x}) = \frac{P(\mathbf{x}|y = -1)P(y = -1)}{\lambda + \sum_{c \in \{-1, +1\}} P(\mathbf{x}|y = c)P(y = c)} \quad (3.3)$$

Intuitively, λ is a placeholder for “everything else” class. We point out that this is different from the Laplace correction which is used to smooth out probability estimates derived from the empirical counts¹. First, this regularization is applied directly to Bayes theorem and not to a probability estimate. Second, this regularization only changes the denominator of the Bayes theorem and effectively creates the aforementioned “everything else” class.

3.1.6.1 Regularized Random Walk CAD We will refer to just proposed algorithm as λ -regularized random walk CAD algorithm (λ -RWCAD). Algorithm 1 displays the pseudo-code of the λ -RWCAD algorithm. Notice that $\text{vol}(W^+)$ and $\text{vol}(W^-)$ are constants and can be pre-computed. One of the benefits of λ -RWCAD algorithm is that it does not require us to store the whole $n \times n$ similarity matrix. Moreover, the method requires only a nearest neighbor type of a computation, and therefore it has $O(n^2)$ time and $O(n)$ space requirements. For sparse representations of the graph the time is reduced to $O(|E|)$, where $|E|$ is the number of edges in the graph. On the other hand, many other graph-based algorithms require quadratic space and their time complexity is related to the computation of the inverse of $n \times n$ matrix which is $\Omega(n^2)$ and $O(n^{2.807})$ in most practical implementations². Finally, modeling the data distribution with a graph can be extended to online learning [Kivinen et al., 2002]. Unlike the label propagation methods that us require to store the whole $O(n^2)$ weight matrix ($O(|E|)$ when it is sparse) for the future computations, our method requires only a summary statistic for each vertex which is $O(n)$.

¹ $P(y = k) = (N_k + \lambda) / (\sum_k N_k + K\lambda)$, where K is the number of classes and N_k are the corresponding counts

²The complexity can be improved to $O(n^{2.376})$ by using the Coppersmith-Winograd algorithm.

Algorithm 1 RWCAD that calculates the anomaly score

Inputs:

new example (\mathbf{x}_e, y_e)
similarity metric $K(\cdot, \cdot)$
 $\text{vol}(W^+) = \sum_{y_i=y_j=+1} W_{ij}$
 $\text{vol}(W^-) = \sum_{y_i=y_j=-1} W_{ij}$
regularization coefficient λ

Algorithm:

$W_{i\mathbf{x}_e}^+ = K(\mathbf{x}_i, \mathbf{x}_e), \forall i \text{ positive}$
 $W_{i\mathbf{x}_e}^- = K(\mathbf{x}_i, \mathbf{x}_e), \forall i \text{ negative}$
 $P(\mathbf{x}_e | y = +1) = \sum_i W_{i\mathbf{x}_e}^+ / \left(\text{vol}(W^+) + 2 \times \sum_i W_{i\mathbf{x}_e}^+ \right)$
 $P(\mathbf{x}_e | y = -1) = \sum_i W_{i\mathbf{x}_e}^- / \left(\text{vol}(W^-) + 2 \times \sum_i W_{i\mathbf{x}_e}^- \right)$
 $P(y \neq y_e | \mathbf{x}_e) = \frac{P(\mathbf{x}_e | y \neq y_e) P(y \neq y_e)}{\lambda + \sum_{c \in \{-1, +1\}} P(\mathbf{x}_e | y=c) P(y=c)}$

Outputs:

$P(y \neq y_e | \mathbf{x}_e)$

3.1.6.2 Conditional Anomaly Detection with Soft Harmonic Functions In this section we show how to solve the CAD problem using label propagation on the data similarity graph and how to compute the anomaly score. In particular, we will build on the harmonic solution approach (Section 2.2.2.1) and adopt it for CAD in the following ways: 1) show how to compute the confidence of mislabeling, 2) add a regularizer to address the problem of isolated and fringe points, 3) use soft constraints to account for a fully labeled setting, and 4) describe a compact computation of the solution from a quantized backbone graph.

The label propagation method described in Section 2.2.2.1 can be applied to CAD by considering all observed data as labeled examples with no unlabeled examples. The setting for matrix C is dependent on the quality of the past observed data. If the labels of the past observed data (or any example from the recent sample) are guaranteed to be correct, we set the corresponding diagonal elements of C to a large value to make their labels fixed. Notice that specific domain techniques can be utilized to make sure that the collected examples from the past observed data have correct labels. We assume that we do not have the access to such prior knowledge and therefore, the observed data are also subject to label noise.

We now propose a way to compute the anomaly score from (2.6). The output ℓ^* of (2.5)

for the example i can be rewritten as:

$$\ell_i^\star = |\ell_i^\star| \times \text{sgn}(\ell_i^\star) \quad (3.4)$$

SSL methods use $\text{sgn}(\ell_i^\star)$ in (3.4) as the predicted label for i . For an unlabeled example, the closer the value of ℓ_i is to ± 1 , the more consistent labeling information was propagated to it. Typically, that means that the example is close to the labeled examples of the respective class. The key observation, which we exploit here, is that we can interpret $|\ell_i^\star|$ as a confidence of the label. Our situation differs from SSL as all our examples are labeled and we aim to assess the confidence of *already labeled* example. Therefore, we define the *anomaly score* as the absolute difference between the actual label y_i and the inferred soft label ℓ_i :

$$s_i = |\ell_i^\star - y_i|. \quad (3.5)$$

We will now address the problems illustrated in Figure 4. Recall that the isolated points are the examples that are far (with respect to some metric) from the majority of the data. Consequently, they are surrounded by few or no nearby points. Therefore, no matter what their label is, we do not want to report them as conditional anomalies. In other words, we want CAD methods to assign them a low anomaly score. Even when the isolated points are far from the majority data, they still can be orders of magnitudes closer to the data points with the opposite label. This can make a label propagation approach falsely confident about that example being a conditional anomaly. In the same way we do not want to assign a high anomaly score to fringe points just because they lie on the distribution boundary. To tackle these problems we set $K = L + \gamma_g I$, where we diagonally regularize the graph Laplacian. Intuitively, such a regularization lowers the confidence value $|\ell^\star|$ of all examples; however it reduces the confidence score of far outlier points relatively more. To see this, notice (Section 5.2.5) that the similarity weight metric is an exponentially decreasing function of the Euclidean distance. In other words, such a regularization can be interpreted as a label propagation on the graph with an extra sink. The sink is an extra node in G with label 0 and every other node connected to it with a small but same weight γ_g . The edge weight of γ_g affects the isolated points more than other points because their connections to other nodes are small.

In the fully labeled setting, the *hard* harmonic solution degenerates to the weighted k -NN. In particular, the hard constraints of the harmonic solution do not allow the labels spread beyond other labeled examples. However, despite the fully labeled case, we still want to take the advantage of the manifold structure. To alleviate this problem we allow labels to spread on the graph by using soft constraints in the unconstrained regularization problem (2.5). In particular, instead of $c_l = \infty$ we set c_l to a finite constant and we set $C = c_l I$. With such a setting of K and C , we can solve (2.5) using (2.6) to get:

$$\ell^\star = ((c_l I)^{-1} (L + \gamma_g) + I)^{-1} \mathbf{y} = \left(c_l^{-1} L + \left(1 + \frac{\gamma_g}{c_l} \right) I \right)^{-1} \mathbf{y}. \quad (3.6)$$

To avoid computation of the inverse,³ we calculate (3.6) using the following system of linear equations:

$$\left(c_l^{-1} L + \left(1 + \frac{\gamma_g}{c_l} \right) I \right) \ell^\star = \mathbf{y} \quad (3.7)$$

We then plug the output of (3.7) into (3.5) to get the anomaly score. We will refer to this score as SoftHAD score. Intuitively, when the confidence is high but $\text{sign}(\ell_i^\star) \neq y_i$, we will consider the label y_i of the case (\mathbf{x}_i, y_i) conditionally anomalous.

Backbone graph The computation of the system of linear equations (3.7) scales with complexity⁴ $O(n^3)$. This is not feasible for a graph with more than several thousands of nodes. To address the problem, we use *data quantization* [Gray and Neuhoff, 1998] and sample a set of nodes from the training data to create G . We then substitute the nodes in the graph with a smaller set of $k \ll n$ distinct centroids which results in $O(k^3)$ computation.

We improve the approximation of the original graph with the backbone graph, by assigning different weights to the centroids. We do it by computing the multiplicities (i.e. how many nodes each centroid represents). In the following we will describe how to modify (3.7) to allow for the computation with multiplicities.

Let V be the diagonal matrix of multiplicities with v_{ii} being the number of nodes that centroid \mathbf{x}_i represents. We will set the multiplicities according to the empirical prior.

³due to numerical instability

⁴The complexity can be further improved to $O(n_u^{2.376})$ with the Coppersmith-Winograd algorithm.

Let W^V be the compact representation of the matrix W on G , where each node \mathbf{x}_i is replicated v_{ii} times. Let L^V and K^V be the graph Laplacian and regularized graph Laplacian of W^V . Finally, let C^V be the C in (2.5) with the adjustment for the multiplicities. C^V accounts for the fact that we care about “fitting” to train data according to the train data multiplicities. Then:

$$\begin{aligned} W^V &= VWV \\ L^V &= L(W^V) \\ K^V &= L^V + \gamma_g V \\ C^V &= V^{1/2} C V^{1/2} \end{aligned}$$

The unconstrained regularization (2.5) now becomes:

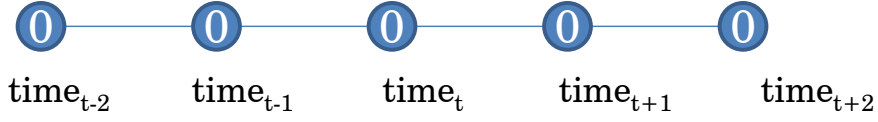
$$\ell^{V\star} = \min_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top C^V (\ell - \mathbf{y}) + \ell^\top K^V \ell \quad (3.8)$$

and subsequently (3.6) becomes:

$$\begin{aligned} \ell^{V\star} &= \left((C^V)^{-1} K^V + I \right)^{-1} \mathbf{y} \\ &= \left(V^{-1/2} C^{-1} V^{-1/2} (L^V + \gamma_g V) + I \right)^{-1} \mathbf{y} \\ &= \left((c_l V)^{-1} (L^V + \gamma_g V) + I \right)^{-1} \mathbf{y} \\ &= \left(1/c_l V^{-1} L^V + c_l \gamma_g + I \right)^{-1} \mathbf{y} \end{aligned}$$

With these adjustments the anomaly score that accounts for the multiplicities is equal to $|\ell^{V\star} - \mathbf{y}|$.

“negative” patient:



“positive” patient:



Figure 6: Unlabeled orders for ‘positive’ patients

3.2 SEMI-SUPERVISED LEARNING

Semi-supervised learning (SSL) is a field of machine learning that studies learning from both labeled and unlabeled examples. This learning paradigm is suitable for real-world problems, where data is often abundant but the resources to label them are limited. As a result, many semi-supervised learning algorithms have been proposed in the past years [Zhu, 2008]. The closest to this work are semi-supervised support vector machines (S3VMs) [Bennett and Demiriz, 1999], manifold regularization of support vector machines (SVMs) [Belkin et al., 2006], and harmonic function solutions on data adjacency graphs [Zhu et al., 2003]. Manifold regularization of SVMs essentially combines the ideas of harmonic function solutions and semi-supervised SVMs in a single convex objective.

In general, taking the advantage from unlabeled data can enhance any CAD method. Many AD methods are related to classification and regression and in some conditions [Singh et al., 2008], taking into account unlabeled data can improve the performance. Since the CAD in medicine is the main focus of this thesis, we discuss the unlabeled data in this domain.

- **missing data:** We look for the anomalies in the medical orders (conditioned on the patient state). Naturally, the patient states with the orders are the input of our algorithms.

The hospital data tend to come from multiple systems (pharmacy records, lab reports, visit data) with fair amount of missing orders (labels).

- **unlabeled patient states:** We illustrate this in Figure 6. Let us consider a rarely ordered lab test (such as HIT test for heparin induced thrombocytopenia) and 2 patients observed over the period of 5 time points. If for a patient that test was never ordered we may assume negative labels for this order throughout all time points. Say, on the other hand that we have a patient with a test order at time t (shown as ‘positive’ patient in Figure 6). We again assume that the patient state at time t is similar to the ones at time $t - 1, t + 1, \dots$ etc. Even though the test was not given at $t - 1, t + 1, \dots$ etc., we should not assume that this patient state is a negative example for the test order. Therefore, we treat this patient states as unlabeled.
- **manifold assumption:** Whatever is the reason for unlabeled orders, including unlabeled data may help us track the data manifold better and lead to a better performance.

In the rest of this section we propose two algorithms for SSL.

3.2.1 Max-Margin Graph Cuts

In this part we present our algorithm, that combines harmonic solution with max-margin learning. Our semi-supervised learning algorithm involves two steps. First, we obtain the regularized harmonic function solution ℓ^* (2.4). The solution is computed from the system of linear equations $(L_{uu} + \gamma_g I)\ell_u = W_{ul}\ell_l$. This system of linear equations is sparse when the data adjacency graph W is sparse. Second, we learn a max-margin discriminator, which is conditioned on the labels induced by the harmonic solution. The optimization problem is given by:

$$\begin{aligned}
& \min_{f \in \mathcal{H}_K} \sum_{i: |\ell_i^*| \geq \varepsilon} V(f, \mathbf{x}_i, \text{sgn}(\ell_i^*)) + \gamma \|f\|_K^2 \\
& \text{s.t. } \ell^* = \arg \min_{\ell \in \mathbb{R}^n} \ell^\top (L + \gamma_g I) \ell \\
& \text{s.t. } \ell_i = y_i \text{ for all } i \in l;
\end{aligned} \tag{3.9}$$

where $V(f, \mathbf{x}, y) = \max\{1 - yf(\mathbf{x}), 0\}$ denotes the *hinge loss*, f is a function from some *reproducing kernel Hilbert space (RKHS)* \mathcal{H}_K , and $\|\cdot\|_K$ is the norm that measures the complexity of f .

Training examples \mathbf{x}_i in our problem are selected based on our confidence into their labels. When the labels are highly *uncertain*, which means that $|\ell_i^*| < \varepsilon$ for some small $\varepsilon \geq 0$, the examples are excluded from learning. Note that as the regularizer γ_g increases, the values $|\ell_i^*|$ decrease towards 0 (Figure 3), and the ε thresholding allows for smooth interpolations between supervised learning on labeled examples and semi-supervised learning on all data. The trade-off between the regularization of f and the minimization of hinge losses $V(f, \mathbf{x}_i, \text{sgn}(\ell_i^*))$ is controlled by the parameter γ .

Due to the representer theorem [Wahba, 1999], the optimal solution f^* to our problem has a special form:

$$f^*(\mathbf{x}) = \sum_{i: |\ell_i^*| \geq \varepsilon} \alpha_i^* k(\mathbf{x}_i, \mathbf{x}),$$

where $k(\cdot, \cdot)$ is a Mercer kernel associated with the RKHS \mathcal{H}_K . Therefore, we can apply the kernel trick and optimize rich classes of discriminators in a finite-dimensional space of $\alpha = (\alpha_1, \dots, \alpha_n)$. Finally, note that when $\gamma_g = \infty$, our solution f^* corresponds to supervised learning with SVMs.

3.2.2 Joint Quantization and Label Propagation

Graph-based semi-supervised learning methods do not scale up well to large data sets mainly because their inference procedures require the computation of the inverse of an $n \times n$ matrix, where n is the size of the underlying graph that is equal to the size of the dataset. A typical solution to address this problem is to downsize the graph to a smaller *backbone* graph and perform the inference on this reduced representation. The key challenge is to decide on what elements should be included in the backbone graph. Typical solutions include sub-sampling, clustering or a Nyström approximation. However, these techniques do not consider the quality of semi-supervised learning inferences for this backbone graph. We introduce a new objective function that lets us incorporate the quality of inferences into the construction of the backbone graph.

To reduce the computational complexity of (2.5), we replace all n nodes of the similarity graph G with a set $C = [\mathbf{c}_1, \dots, \mathbf{c}_m, \dots, \mathbf{c}_{m+k}]^\top$ of $(m+k) \ll n$ representative nodes to create a backbone graph \tilde{G} . Notice that $\mathbf{c}_i = \mathbf{x}_i$ for $i = 1, \dots, m$. We want to find \tilde{G} such that it is a good representation of G in constructing the manifold. Let us assume for a moment that we do know the best set of examples \tilde{G} . Then, Equation (2.5) becomes:

$$\ell^\star = \operatorname{argmin}_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top F^C (\ell - \mathbf{y}) + \ell^\top L^C \ell. \quad (3.10)$$

In general, $C \in \mathbb{R}^{(m+k) \times d}$ can be obtained by fixing the first m labeled examples and choosing k unlabeled points by subsampling the dataset, clustering or other means of quantization. As mentioned earlier, the common approach is to select the set C first and only then perform the inference (3.10). In this work, we will perform both the quantization and the inference jointly, by adding the quantization penalty of the Elastic nets to the objective function in (3.10). As we will see in Section 4.3, this simple joint approach will produce interesting properties. The new objective function is:

$$[\ell^\star, \{c_j\}_{j=m+1}^{m+k}] = \operatorname{argmin}_{\ell \in \mathbb{R}^n, \{c_j\}_{j=m+1}^{m+k}} (\ell - \mathbf{y})^\top F^C (\ell - \mathbf{y}) + \ell^\top L^C \ell + \gamma_q \left(\frac{(m+k)^2}{n} \sum_{x_i \in K_j} \|c_j - x_i\|^2 \right) \quad (3.11)$$

where K_j is the set of examples for which c_j is the nearest centroid and γ_q is a cost parameter for the quantization penalty. We emphasize that we automatically consider all labeled examples as a fixed part of C and the optimization to learn the representing centroids are affected by the position of labeled examples. As we will see in the remainder of this paper, the above objective function has the interesting property: when optimized to find the centroids, it learns the principle manifold.

Adding the quantization penalty makes the objective function non-convex and hence difficult to optimize. To minimize (3.11), we propose to use an alternating optimization approach [Bezdek and Hathaway, 2002], where we alternate between 1) *label propagation* – inferring labels l on \tilde{G} , and 2) *quantization* – selecting the set C for \tilde{G} . Starting with a random seeds of unlabeled examples (or the output of k -means algorithm) as the initial centroids, we iterate the following steps.

3.2.2.1 Label Propagation: Once C is fixed, the labels can be computed by solving the following convex optimization problem:

$$\ell^\star = \operatorname{argmin}_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top F^C (\ell - \mathbf{y}) + \ell^\top L^C \ell$$

The solution to this problem is obtained in closed form as $\ell^\star = ((F^C)^{-1} L^C + I)^{-1} \mathbf{y}$ (see Section 2.2.2).

3.2.2.2 Quantization: To learn the centroids C when ℓ is fixed, first notice that:

$$\ell^\top L^C \ell = \sum_{i,j} \left(\frac{l_i}{n_i} - \frac{l_j}{n_j} \right)^2 W_{ij}^C$$

where $(n_i = 1)_{i=1}^{m+k}$ for unnormalized graph Laplacian $L = D^C - W^C$ [Luxburg, 2007] and $(n_i = \sqrt{d_i})_{i=1}^{m+k}$ for normalized graph Laplacian $L = I - D^{-1/2} W D^{-1/2}$ [Zhou et al., 2004]. Considering that $(\ell - \mathbf{y})^\top F^C (\ell - \mathbf{y})$ in (3.11) is not dependent on C , we have the following optimization problem to learn C if we use the widely used Gaussian kernel⁵ as the similarity function W :

$$\{c_j\}_{j=m+1}^{m+k} = \operatorname{argmin}_{\{c_j\}_{j=m+1}^{m+k}} \sum_{i,j} \left(\frac{l_i}{n_i} - \frac{l_j}{n_j} \right)^2 \exp \left(\frac{-\|c_j - c_i\|^2}{2\sigma^2} \right) + \gamma_q \left(\frac{(m+k)^2}{n} \sum_{i \in K_j} \|c_j - x_i\|^2 \right) \quad (3.12)$$

To learn the centers by optimizing (3.12), we first approximate the exponential function using Taylor expansion⁶:

$$\exp \left(\frac{-\|c_j - c_i\|^2}{2\sigma^2} \right) \approx 1 - \frac{\|c_j - c_i\|^2}{2\sigma^2},$$

This results in the following optimization problem:

$$\{c_j\}_{j=m+1}^{m+k} = \operatorname{argmin}_{\{c_j\}_{j=m+1}^{m+k}} \frac{-1}{(m+k)^2} \sum_{i,j} \left(\frac{(l_i - l_j)^2}{2\sigma^2} \right) \|c_j - c_i\|^2 + \frac{\gamma_q}{n} \sum_{i \in K_j} \|c_j - x_i\|^2 \quad (3.13)$$

⁵It is straightforward to apply similar derivation for other similarity functions.

⁶Notice we could also use the convexity of the exponential function to obtain an upper bound and have a more rigorous derivation. However, the results are very similar and we omit the details to simplify the description.

Taking derivatives of (3.13) with respect to $(c_j)_{j=m+1}^{m+k}$ and setting them to zero, we obtain the following system of k linear equations for $j = m + 1, \dots, m + k$:

$$\sum_i c_i \frac{(l_i - l_j)^2}{(m+k)^2 \sigma^2} + c_j \left(2\gamma_q \frac{|K_j|}{n} - \sum_i \frac{(l_i - l_j)^2}{(m+k)^2 \sigma^2} \right) = \frac{2\gamma_q}{n} \sum_{i \in K_j} x_i, \quad (3.14)$$

where $|K_j|$ is the number of examples assigned to center c_j . In order to optimize the system of linear equations in (3.14), we iterate between optimizing the centroids and the assignment of the examples to the centroids, a strategy similar to k -means.

Notice that the labeled examples c_1, \dots, c_m affect learning the centroids by

1. absorbing some of the unlabeled examples that are close to the labeled examples and do not need an unlabeled examples representative. In other words, in the quantization process, we remove those examples that are very close to the labeled examples;
2. controlling the position of unlabeled centers through the first term in (3.14).

Algorithm 2 outlines the *elastic-joint* algorithm.

Algorithm 2 Quantized semi-supervised learning with principal manifolds

Inputs:

examples $\{\mathbf{x}_i\}_{i=1}^n$
labels l , such that $l_i = \pm 1$ for labeled and $l_i = 0$ for unlabeled examples
 k : number of centroids and size of the \tilde{G}
regularizer γ_q for the quantization

Algorithm (elastic-joint):

randomly initialize the set of k centroids C

do until convergence

infer labels on the graph:

build a quantized data similarity graph \tilde{G} on C

compute L^C as the graph Laplacian of \tilde{G}

$\ell^* = \operatorname{argmin}_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top F^C (\ell - \mathbf{y}) + \ell^\top L^C \ell$

perform quantization

calculate C by solving the following system of linear equations for $j = m + 1, \dots, m + k$:

$$\sum_i c_i \frac{(l_i - l_j)^2}{(m+k)^2 \sigma^2} + c_j \left(2\gamma_q \frac{|K_j|}{n} - \sum_i \frac{(l_i - l_j)^2}{(m+k)^2 \sigma^2} \right) = \frac{2\gamma_q}{n} \sum_{i \in K_j} x_i$$

Outputs:

predictions $\hat{y} = |\ell^*|$

3.2.2.3 Approximation After solving the objective function in (3.11) using Algorithm 2, we need to infer the labels of unlabeled examples from the labels of the centroids. The

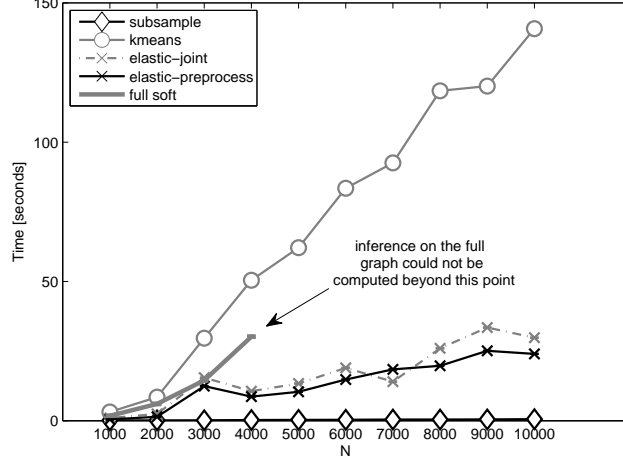


Figure 7: Running time for different methods on the SecStr dataset

common approach in the literature [Chapelle et al., 2006, Delalleau et al., 2005] is to use the weighted nearest neighbors. The label of any new example x (including the unlabeled examples) is computed as follows:

$$\hat{y} = \frac{\sum_{i=1}^{m+k} W'(x, c_i) \ell_i}{\sum_{i=1}^{m+k} W'(x, c_i)} \quad (3.15)$$

where W' is the symmetric edge weighting function such as Gaussian kernel [Chapelle et al., 2006]. Here, we only use 1-NN for the inference as we found that it produces the best results for the proposed method and the baselines.

3.2.2.4 Time Complexity Suppose Algorithm 2 takes T iterations to converge. Each iteration has two optimization steps: I) running SSL, and II) Constructing the backbone graph. Each run of SSL algorithm needs the computation of the inverse of a matrix of size $n + k$ that takes $O((m + k)^3)$. Each run of the backbone graph construction iterates between two steps 1) assigning examples to centroids, and 2) solving the system of linear equations (3.13). The second step is the major step and takes $O(k^3)$ which results in $O(tk^3)$ time complexity for t iterations. Since $(m + k)^3 \geq tk^3$ for even a small number of labeled examples, the complexity of the proposed method is $O(T(m + k)^3)$. In our experiments, we

found that T is usually very small; i.e. less than 10. Figure 7 shows the running time of different methods on SecStr dataset [Chapelle et al., 2006] by changing the total number of unlabeled examples from 1000 to 10000. Different quantization approaches are described in Section 5.3.2. We fixed number of labeled examples to $m = 10$, number of centroids to $k = 100$ and varied the number of sampled points N from the original 83679 examples. This plot clearly shows that the proposed method scales very well with the large number of examples. Also note that we used the k -means function in MATLAB which seems extremely slow.

3.3 ONLINE SEMI-SUPERVISED LEARNING WITH QUANTIZED GRAPHS

The regularized HS (Section 2.2.2) is an offline learning algorithm. This algorithm can be made naïvely online, by taking each new example, connecting it to its neighbors, and re-computing the HS. Unfortunately, this naïve implementation has computational complexity $O(t^3)$ at step t , and computation becomes infeasible as more examples are added to the graph.

To address the problem, we use *data quantization* [Gray and Neuhoff, 1998] and substitute the vertices in the graph with a smaller set of k distinct centroids. The resulting $t \times t$ similarity matrix W has many identical rows/columns. We will show that the *exact* HS using W may be reconstructed from a much smaller $k \times k$ matrix \tilde{W}^q , where \tilde{W}_{ij}^q contains the similarity between the i^{th} and j^{th} centroids, and a vector \mathbf{v} of length k , where v_i denotes to number of points collapsed into the i^{th} centroid. To show this, we introduce the matrix $W^q = V\tilde{W}^qV$ where V is a diagonal matrix containing the counts in \mathbf{v} on the diagonal.

Proposition 1. *The harmonic solution (2.3) using W can be computed compactly as*

$$\ell^q = (L_{uu}^q + \gamma_g V)^{-1} W_{ul}^q \ell_l,$$

where L^q is the Laplacian of W^q .

Proof: Our proof uses the electric circuit interpretation of a random walk [Zhu et al., 2003]. More specifically, we show that W and W^q represent identical electric circuits and therefore, their harmonic solutions are the same.

In the electric circuit formulation of W , the edges of the graph are resistors with the conductance w_{ij} . If two vertices i and j are identical, then by symmetry, the HS must assign the same value to both vertices and we may replace them with a single vertex. Furthermore, they correspond to ends of resistors in parallel. The total conductance of two resistors in parallel is equal to the sum of their conductances. Therefore, the two resistors can be replaced by a single resistor with the conductance of the sum. A repetitive application of this rule gives $W^q = V\tilde{W}^qV$, which yields the same HS as W . In Section 2.2.2, we showed that the regularized HS can be interpreted as having an extra sink in a graph. Therefore, when two vertices i and j are merged, we also need to sum up their sinks. A repetitive application of this rule yields the term $\gamma_g V$ in our closed-form solution. ■

We note that Proposition 1 may be applied whenever the similarity matrix has identical rows/columns, not just when quantization is applied. However, when the data points are quantized into a fixed number of centroids k , it shows that we may compute the HS for the t^{th} point in $O(k^3)$ time. Since the time complexity of computation on the quantized graph is independent of t , it gives a suitable algorithm for online learning.

We now describe how to perform incremental quantization with provably nearly-optimal distortion.

3.3.1 Incremental k-centers

We make use of *doubling algorithm* for incremental k -center clustering [Charikar et al., 1997] which assigns points to centroids in a near optimal way. In particular, it is a $(1 + \epsilon)$ -approximation with cost measured by the maximum quantization error over all points. In Section 4.4.3, we show that under reasonable assumptions, the quantization error goes to zero as the number of centroids increases.

The algorithm of [Charikar et al., 1997] maintains a set of centroids $C_t = \{\mathbf{c}_1, \mathbf{c}_2, \dots\}$ such that the distance between any two vertices in C_t is at least R and $|C_t| \leq k$ at the end of each

iteration. For each new point \mathbf{x}_t , if its distance to some $\mathbf{c}_i \in C_t$ is less than R , the point is assigned to \mathbf{c}_i . Otherwise, the distance of \mathbf{x}_t to $\mathbf{c}_i \in C_t$ is at least R and \mathbf{x}_t is added to the set of centroids C_t . If adding \mathbf{x}_t to C_t results in $|C_t| > k$, the scalar R is doubled and C_t is greedily repartitioned such that no two vertices in C_t are closer than R . The doubling of R also ensures that $|C_t| < k$.

Pseudocode of our algorithm is given in Algorithm 3. We make a small modification to the original quantization algorithm in that, instead of doubling R , we multiply it with some $m > 1$. This still yields a $(1+\epsilon)$ -approximation algorithm as it still obeys the invariants given in Lemma 3.4 in [Charikar et al., 1997]. We also maintain a vector of multiplicities \mathbf{v} which contains the number of vertices that each centroid represents. At each time step, the HS is calculated using the updated quantized graph, and a prediction is made.

The incremental k -centers algorithm also has the advantage that it provides a variable R which may be used to bound the maximum quantization error. In particular, at any point in time t , the distance of any example from its centroid is at most $Rm/(m-1)$. To see this, consider the following. As the new data arrive we keep increasing R by multiplying it by some $m > 1$. But for any point at any time, the centroid assigned to a vertex is at most R apart from the previously assigned centroid, which is at most R/m apart from the centroid assigned before, etc. Summing up, at any time, any point is at most

$$R + \frac{R}{m} + \frac{R}{m^2} + \dots = R \left(1 + \frac{1}{m} + \frac{1}{m^2} + \dots \right) = \frac{Rm}{m-1}$$

apart from its assigned centroid, where R is the most recent one.

3.4 PARALLEL MULTI-MANIFOLD LEARNING

Most of the SSL methods that exploit the manifold assumption (such as graph-based SSL methods) assume that the data lie on a single manifold. More plausible setting, however, is that the data lie on a mixture of manifolds [Goldberg et al., 2009]. For example, in digit recognition each digit lies on its own manifold in the feature space [Goldberg et al., 2009].

Algorithm 3 Online quantized harmonic solution

Inputs:

an unlabeled example \mathbf{x}_t
a set of centroids C_{t-1}
vertex multiplicities \mathbf{v}_{t-1}

Algorithm:

if $(|C_{t-1}| = k + 1)$
 $R \leftarrow mR$
 greedily repartition C_{t-1} into C_t such that:
 no two vertices in C_t are closer than R
 for any $\mathbf{c}_i \in C_{t-1}$ exists $\mathbf{c}_j \in C_t$ such that $d(\mathbf{c}_i, \mathbf{c}_j) < R$
 update \mathbf{v}_t to reflect the new partitioning
else
 $C_t \leftarrow C_{t-1}$
 $\mathbf{v}_t \leftarrow \mathbf{v}_{t-1}$
if \mathbf{x}_t is closer than R to any $\mathbf{c}_i \in C_t$
 $\mathbf{v}_t(i) \leftarrow \mathbf{v}_t(i) + 1$
else
 $\mathbf{v}_t(|C_t| + 1) \leftarrow 1$
 $C_t(|C_t| + 1) \leftarrow \mathbf{x}_t$
build a similarity matrix \tilde{W}_t^q over the vertices C_t
build a matrix V_t whose diagonal elements are \mathbf{v}_t
 $W_t^q = V_t \tilde{W}_t^q V_t$
compute the Laplacian L^q of the graph W_t^q
infer labels on the graph:
 $\ell^q[t] \leftarrow \operatorname{argmin}_{\ell} \ell^\top (L^q + \gamma_g V_t) \ell$
 s.t. $\ell_i = y_i$ for all labeled examples up to time t
make a prediction $\hat{y}_t = \operatorname{sgn}(\ell_t^q[t])$

Outputs:

a prediction \hat{y}_t
a set of centroids C_t
vertex multiplicities \mathbf{v}_t

In this work, we use the multi-manifold idea from a different perspective. We assume no or little interactions between the manifolds and learn the manifolds *in parallel* to achieve a speedup in computation. The speedup is accomplished in two ways:

1. Assuming independence between the manifolds, we can solve several smaller problems instead. For example, in the ideal case (Section 4.5), the similarity matrix will consist of b block-diagonal blocks of the equal size. Therefore to approximate the harmonic solution (HS) on a graph with n nodes which takes $\mathcal{O}(n^3)$ time, we can instead solve b HS problems on b graph with n/b nodes, each taking only $\mathcal{O}((n/b)^3)$ and achieve a polynomial speedup.
2. Using multi-core and/or multi-processor architectures, we can solve the smaller problems in parallel and achieve additional, potentially linear speedup, up to the number of cores.

Assuming the independence of manifolds, when the manifolds are not independent, may come with a cost in accuracy. We study this theoretically in Section 4.5 and empirically in Section 5.3.4 where we measure the trade-off between the computational speedup and decrease in prediction accuracy.

4.0 THEORETICAL ANALYSIS

In this chapter, we analyze the methods proposed in Chapter 3. We will be mostly with:

- **generalization** errors induced by harmonic solutions on the graph,
- errors induced by **quantization** of the graph to accommodate online learning, and
- errors due to the **online** setting.

4.1 ANALYSIS OF CONDITIONAL ANOMALY DETECTION

4.1.1 RWCAD and its Relationship to the weighted k -NN

In this part we show that weighed k -NN is a special case of λ -RWCAD for $\lambda = 0$ and $n \rightarrow \infty$.

Rewriting (3.2) we get:

$$P(y \neq +1 | \mathbf{x}_i) = \frac{1}{1 + \frac{P(\mathbf{x}_i | y=+1)P(y=+1)}{P(\mathbf{x}_i | y=-1)P(y=-1)}} \quad (4.1)$$

Let us estimate $P(\mathbf{x}_i | y = +1)/P(\mathbf{x}_i | y = -1)$ – *conditional likelihood* – in (4.1) also from a stationary distribution of a random walk shown in Figure 8, where we connect the node representing \mathbf{x}_i with all examples in the training set (from all classes) and define the likelihood ratio as the ratio between the time spent in the nodes with the respective labels:

$$\frac{P(\mathbf{x}_i | y = +1)}{P(\mathbf{x}_i | y = -1)} = \frac{\#T(y = +1)}{\#T(y = -1)}, \quad (4.2)$$

where $\#T(y = c)$ is the time spent in the nodes of class c during the random walk. Let W, W^+, W^- be the weight matrices for all, just the positive and just the negative nodes respectively.

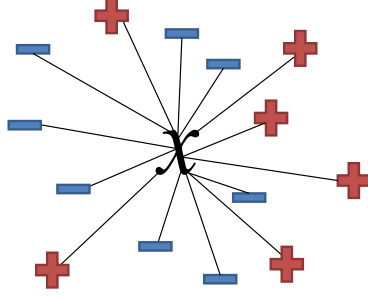


Figure 8: Estimating likelihood ratio from a single graph

Combining (4.2) and (2.1), we get:

$$\frac{P(\mathbf{x}_i|y = +1)}{P(\mathbf{x}_i|y = -1)} = \frac{\sum_j W_{j\mathbf{x}_i}^+}{\sum_j W_{j\mathbf{x}_i}^-} \quad (4.3)$$

which is equal to weighted k -NN method. Now, let $T^+ = \text{vol}(W^+)$ and $T^- = \text{vol}(W^-)$ be the sums of all weights in W^+ and W^- . Moreover, let $T_{\mathbf{x}_i}^+$, $T_{\mathbf{x}_i}^-$ be the total edge sums of the respective graphs including the node \mathbf{x}_i . The conditional likelihood of the λ -RWCAD for $\lambda = 0$ can be derived combining (3.2) and (2.1) to get:

$$\frac{P(\mathbf{x}_i|y = +1)}{P(\mathbf{x}_i|y = -1)} = \frac{\sum_j W_{j\mathbf{x}_i}^+}{\sum_j W_{j\mathbf{x}_i}^-} \times \frac{T_{\mathbf{x}_i}^-}{T_{\mathbf{x}_i}^+} \quad (4.4)$$

Equations (4.3) and (4.4) are the conditional likelihoods for weighted k -NN and RWCAD for $\lambda = 0$ respectively. Notice that as the number of nodes increases, $T_{\mathbf{x}_i}^-/T_{\mathbf{x}_i}^+$ approaches T^-/T^+ which is a constant. Therefore, the influence of one node (\mathbf{x}_i) in the ratio becomes negligible. In that case, both methods will yield comparable results.

4.1.2 Soft Harmonic Solution

One of the most challenging part of anomaly detection (both conditional and unconditional) is the evaluation. The expert knowledge is often necessary to evaluate the performance. That is also why we have conducted several evaluations with field experts (Section 5.4.2).

When the human expert knowledge is not available and in theoretical analysis we often rely on some surrogate evaluation measures.

In this section we prove a bound on the generalization error of our transductive learner. The generalization error of the solution to the problem (2.6) (and also (2.5)) is bounded in Lemma 1.

Lemma 1. *Let ℓ^* be a solution to the problem:*

$$\min_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top C(\ell - \mathbf{y}) + \ell^\top Q \ell,$$

where $Q = L + \gamma_g I$ and all labeled examples l are selected i.i.d. Then the inequality:

$$\begin{aligned} R_P^W(\ell^*) &\leq \underbrace{\widehat{R}_P^W(\ell^*) + \beta + \sqrt{\frac{2 \ln(2/\delta)}{n_l}}(n_l \beta + 4)}_{\text{transductive error } \Delta_T(\beta, n_l, \delta)} \\ \beta &\leq 2 \left[\frac{\sqrt{2}}{\gamma_g + 1} + \sqrt{2n_l} \frac{1 - \sqrt{c_u}}{\sqrt{c_u}} \frac{\lambda_M(L) + \gamma_g}{\gamma_g^2 + 1} \right] \end{aligned}$$

holds with probability $1 - \delta$, where:

$$\begin{aligned} R_P^W(\ell^*) &= \frac{1}{n} \sum_i (\ell_i^* - y_i)^2 \\ \widehat{R}_P^W(\ell^*) &= \frac{1}{n_l} \sum_{i \in l} (\ell_i^* - y_i)^2 \end{aligned}$$

are risk terms for all and labeled vertices, respectively, and β is the stability coefficient of the solution ℓ^* .

Proof: To simplify the proof, we assume that $c_l = 1$ and $c_l > c_u$. Our risk bound follows from combining Theorem 1 of [Belkin et al., 2004] with the assumptions $|y_i| \leq 1$ and $|\ell_i^*| \leq 1$. The coefficient β is derived based on Section 5 of [Cortes et al., 2008]. In particular, based on the properties of the matrix C and Proposition 1 [Cortes et al., 2008], we conclude:

$$\beta = 2 \left[\frac{\sqrt{2}}{\lambda_m(Q) + 1} + \sqrt{2n_l} \frac{1 - \sqrt{c_u}}{\sqrt{c_u}} \frac{\lambda_M(Q)}{(\lambda_m(Q) + 1)^2} \right],$$

where $\lambda_m(Q)$ and $\lambda_M(Q)$ refer to the smallest and largest eigenvalues of Q , respectively, and can be further rewritten as $\lambda_m(Q) = \lambda_m(L) + \gamma_g$ and $\lambda_M(Q) = \lambda_M(L) + \gamma_g$. Our final claim directly follows from applying the lower bounds $\lambda_m(L) \geq 0$ and $(\lambda_m(L) + \gamma_g + 1)^2 \geq \gamma_g^2 + 1$. ■

Lemma 1 is practical when the error $\Delta_T(\beta, n_l, \delta)$ decreases at the rate of $O(n_l^{-\frac{1}{2}})$. This is achieved when $\beta = O(1/n_l)$, which corresponds to $\gamma_g = \Omega(n_l^{\frac{3}{2}})$. Thus, when the problem (2.5) is sufficiently regularized, its solution is stable, and the generalization error of the solution is bounded.

4.2 ANALYSIS OF MAX-MARGIN GRAPH CUTS

4.2.1 When Manifold Regularization Fails

The major difference between manifold regularization (2.8) and the regularized harmonic function solution (2.3) is in the space of optimized parameters. In particular, manifold regularization is performed on a class of functions \mathcal{H}_K . When this class is severely restricted, such as linear functions, the minimization of $\mathbf{f}^\top L \mathbf{f}$ may lead to results, which are significantly worse than the harmonic function solution.

This issue can be illustrated on the problem from Figure 3, where we learn a linear decision boundary $f(\mathbf{x}) = \alpha_1 x_1 + \alpha_2 x_2$ through manifold regularization of linear SVMs:

$$\min_{\alpha_1, \alpha_2} \sum_{i \in l} V(f, \mathbf{x}_i, y_i) + \gamma[\alpha_1^2 + \alpha_2^2] + \gamma_u \mathbf{f}^\top L \mathbf{f}. \quad (4.5)$$

The structure of our problem simplifies the computation of the regularization term $\mathbf{f}^\top L \mathbf{f}$. In particular, since all edges in the data adjacency graph are either horizontal or vertical, the term $\mathbf{f}^\top L \mathbf{f}$ can be expressed as a function of α_1^2 and α_2^2 . Therefore, for this particular problem

we have:

$$\begin{aligned}
\mathbf{f}^\top L \mathbf{f} &= \frac{1}{2} \sum_{i,j} w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 \\
&= \frac{1}{2} \sum_{i,j} w_{ij} (\alpha_1 (\mathbf{x}_{i1} - \mathbf{x}_{j1}) + \alpha_2 (\mathbf{x}_{i2} - \mathbf{x}_{j2}))^2 \\
&= \frac{\alpha_1^2}{2} \underbrace{\sum_{i,j} w_{ij} (\mathbf{x}_{i1} - \mathbf{x}_{j1})^2}_{\Delta=218.351} + \\
&\quad \frac{\alpha_2^2}{2} \underbrace{\sum_{i,j} w_{ij} (\mathbf{x}_{i2} - \mathbf{x}_{j2})^2}_{\Delta=218.351}.
\end{aligned} \tag{4.6}$$

After we incorporate (4.6) to our objective function (4.6), we get (4.6) as an additional weight at the regularizer $[\alpha_1^2 + \alpha_2^2]$:

$$\min_{\alpha_1, \alpha_2} \sum_{i \in l} V(f, \mathbf{x}_i, y_i) + \left(\gamma + \frac{\gamma_u \Delta}{2} \right) [\alpha_1^2 + \alpha_2^2] = \min_{\alpha_1, \alpha_2} \sum_{i \in l} V(f, \mathbf{x}_i, y_i) + \gamma^* [\alpha_1^2 + \alpha_2^2], \tag{4.7}$$

where $\gamma^* = \left(\gamma + \frac{\gamma_u \Delta}{2} \right)$. Thus, manifold regularization of linear SVMs on our problem can be viewed as supervised learning with linear SVMs with a varying weight at the regularizer. In other words, in this particular problem, the unlabeled examples only influence the solution through the regularizer γ^* on $f(\mathbf{x})$. That means, we can get the same $f(\mathbf{x})$ for a different γ^* if the unlabeled examples were not present at all. Since the problem involves only two labeled examples, changes in the weight γ^* do not affect the direction of the discriminator $f^*(\mathbf{x}) = 0$, because the margin is maximized by the hyperplane between them. Therefore, different settings of regularizer only change the slope of f^* (Figure 9, second row). The above analysis shows that the discriminator $f^*(\mathbf{x}) = 0$ does not change with γ_u . As a result, all discriminators are equal to the discriminator for $\gamma_u = 0$, which can be learned by linear SVMs, and none of them solves our problem optimally. Max-margin graph cuts solve the problem optimally for small values of γ_g . If we included more unlabeled examples, we could get the error arbitrarily large, assuming our problems would consist of two coherent square-shaped classes as in 3. Figure 9 shows linear, cubic, and RBF decision boundaries obtained by manifold regularization of SVMs (MR) and max-margin graph cuts (GC) on the problem from Figure 3. The regularization parameter $\gamma_g = \gamma/\gamma_u$ is set as suggested in Section 2.4.1.2,

$\gamma = 0.1$, and $\varepsilon = 0.01$. The pink and blue colors denote parts of the feature space \mathbf{x} where the discriminators f are positive and negative, respectively. The yellow color marks regions where $|f(\mathbf{x})| < 0.05$.

A similar line of reasoning can be used to extend our results to polynomial kernels. Figure 9 indicates that max-margin learning with the cubic kernel exhibits similar trends to the linear case.

The notion of algorithmic stability can be used to bound the generalization error of many learning algorithms [Bousquet and Elisseeff, 2002]. In this section, we discuss how to make the harmonic function solution stable and prove a bound on the generalization error of max-margin cuts (3.9). Our bound combines existing transductive [Belkin et al., 2004, Cortes et al., 2008] and inductive [Vapnik, 1995] bounds.

4.2.2 Generalization Error

Our objective is to show that the *risk* of our solutions f :

$$R_P(f) = \mathbb{E} P(\mathbf{x}) \mathcal{L}(f(\mathbf{x}), y(\mathbf{x})) \quad (4.8)$$

is bounded by the *empirical risk* on graph-induced labels:

$$\frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) \quad (4.9)$$

and error terms, which can be computed from training data. The function $\mathcal{L}(y', y) = 1\{\text{sgn}(y') \neq y\}$ computes the zero-one loss of the prediction $\text{sgn}(y')$ given the ground truth y , and $P(\mathbf{x})$ is the distribution of our data. For simplicity, we assume that the label y is a deterministic function of \mathbf{x} . Our proof starts by relating $R_P(f)$ and graph-induced labels ℓ_i^* .

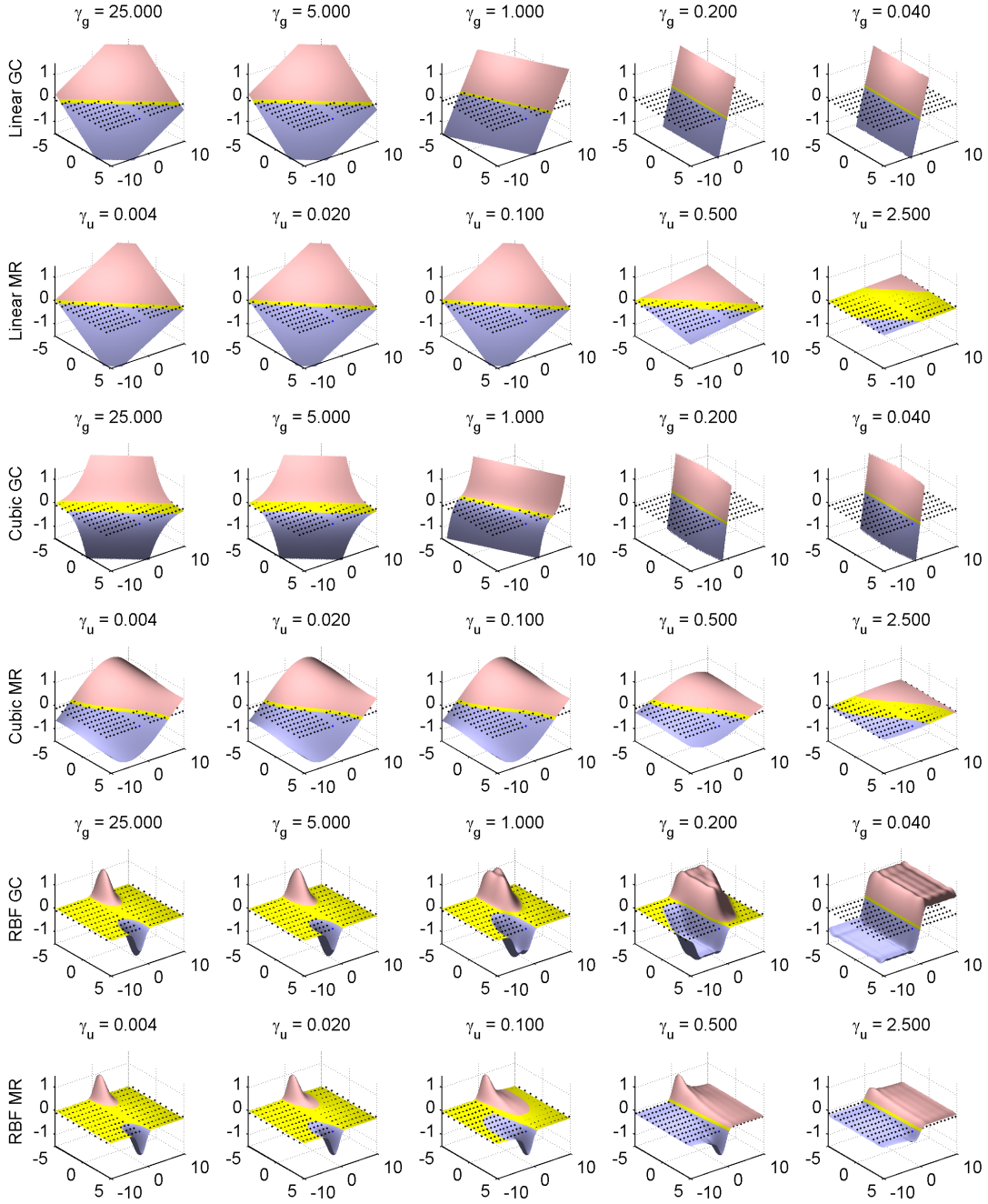


Figure 9: Linear, cubic, and RBF decision boundaries for different methods.

Lemma 2. *Let f be from a function class with the VC dimension h and \mathbf{x}_i be n examples, which are sampled i.i.d. with respect to the distribution $P(\mathbf{x})$. Then the inequality:*

$$R_P(f) \leq \frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \frac{1}{n} \sum_i (\ell_i^* - y_i)^2 + \underbrace{\sqrt{\frac{h(\ln(2n/h) + 1) - \ln(\eta/4)}{n}}}_{\text{inductive error } \Delta_I(h, n, \eta)}$$

holds with probability $1 - \eta$, where y_i and ℓ_i^ represent the true and graph-induced soft labels, respectively.*

Proof: Based on Equations 3.15 and 3.24 [Vapnik, 1995], the inequality:

$$R_P(f) \leq \frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), y_i) + \Delta_I(h, n, \eta)$$

holds with probability $1 - \eta$. Our final claim follows from bounding all terms $\mathcal{L}(f(\mathbf{x}_i), y_i)$ as:

$$\mathcal{L}(f(\mathbf{x}_i), y_i) \leq \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + (\ell_i^* - y_i)^2.$$

The above bound holds for any $y_i \in \{-1, 1\}$ and ℓ_i^* . ■

It is hard to bound the error term $\frac{1}{n} \sum_i (\ell_i^* - y_i)^2$ when the constraints $\ell_i = y_i$ (2.3) are enforced in a hard manner. Thus, in the rest of our analysis, we consider a relaxed version of the harmonic function solution (Section 2.2.2.1). Lemma 1 and its proof can be found in Section 4.1.2. Lemmas 1 and 2 can be combined using the union bound.

Proposition 2. *Let f be from a function class with the VC dimension h . Then the inequality:*

$$R_P(f) \leq \frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \hat{R}_P^w(\ell^*) + \Delta_T(\beta, n_l, \delta) + \Delta_I(h, n, \eta)$$

holds with probability $1 - (\eta + \delta)$.

The above result can be viewed as follows. If both n and n_l are large, the sum of $\frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*))$ and $\hat{R}_P^w(\ell^*)$ provides a good estimate of the risk $R_P(f)$. Unfortunately, our bound is not practical for setting γ_g because it is hard to find γ_g that minimizes both $\hat{R}_P^w(\ell^*)$ and $\Delta_T(\beta, n_l, \delta)$. The same phenomenon was observed by [Belkin et al., 2004] in a similar context. To solve our problem, we suggest setting γ_g based on the validation set. This methodology is used in the experimental section.

4.2.3 Threshold epsilon

Finally, note that when $|\ell_i^*| < \varepsilon$, where ε is a small number, $|\ell_i^* - y_i|$ is close to 1 irrespective of y_i , and a trivial upper bound $\mathcal{L}(f(\mathbf{x}_i), y_i) \leq 1$ is almost as good as $\mathcal{L}(f(\mathbf{x}_i), y_i) \leq \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + (\ell_i^* - y_i)^2$ for any f . This allows us to justify the ε threshold in the problem (3.9). In particular, note that $\mathcal{L}(f(\mathbf{x}_i), y_i)$ is bounded by $1 - (\ell_i^* - y_i)^2 + (\ell_i^* - y_i)^2$. When $|\ell_i^*| < \varepsilon$, $1 - (\ell_i^* - y_i)^2 < 2\varepsilon - \varepsilon^2$, and we conclude the following.

Proposition 3. *Let f be from a function class with the VC dimension h and n_ε be the number of examples such that $|\ell_i^*| < \varepsilon$. Then the inequality:*

$$R_P(f) \leq \frac{1}{n_{i:|\ell_i^*| \geq \varepsilon}} \sum \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \frac{2\varepsilon n_\varepsilon}{n} + \hat{R}_P^w(\ell^*) + \Delta_T(\beta, n_l, \delta) + \Delta_I(h, n, \eta)$$

holds with probability $1 - (\eta + \delta)$.

Proof: The generalization bound is proved as:

$$\begin{aligned}
R_P(f) &\leq \widehat{R}_P(f) + \Delta_I(h, n, \eta) \\
&= \frac{1}{n_{i:|\ell_i^*| \geq \varepsilon}} \sum \mathcal{L}(f(\mathbf{x}_i), y_i) + \frac{1}{n_{i:|\ell_i^*| < \varepsilon}} \sum \mathcal{L}(f(\mathbf{x}_i), y_i) + \\
&\quad \Delta_I(h, n, \eta) \\
&\leq \frac{1}{n_{i:|\ell_i^*| \geq \varepsilon}} \sum [\mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + (\ell_i^* - y_i)^2] + \\
&\quad \frac{1}{n_{i:|\ell_i^*| < \varepsilon}} \sum [1 - (\ell_i^* - y_i)^2 + (\ell_i^* - y_i)^2] + \\
&\quad \Delta_I(h, n, \eta) \\
&= \frac{1}{n_{i:|\ell_i^*| \geq \varepsilon}} \sum \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \\
&\quad \frac{1}{n_{i:|\ell_i^*| < \varepsilon}} \sum [1 - (\ell_i^* - y_i)^2] + \frac{1}{n} \sum_i (\ell_i^* - y_i)^2 + \\
&\quad \Delta_I(h, n, \eta) \\
&\leq \frac{1}{n_{i:|\ell_i^*| \geq \varepsilon}} \sum \mathcal{L}(f(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \frac{2\varepsilon n_\varepsilon}{n} + \\
&\quad \widehat{R}_P(\ell^*) + \Delta_T(\beta, n_l, \delta) + \Delta_I(h, n, \eta).
\end{aligned}$$

The last step follows from the inequality $1 - (\ell_i^* - y_i)^2 < 2\varepsilon$ and Lemma 1. ■

When $\varepsilon \leq n_l^{-\frac{1}{2}}$, the new upper bound is asymptotically as good as the bound in Proposition 2. As a result, we get the same convergence guarantees although highly-uncertain labels $|\ell_i^*| < \varepsilon$ are excluded from our optimization.

In practice, optimization of the thresholded objective often yields a lower risk

$$\frac{1}{n} \sum_{i:|\ell_i^*| \geq \varepsilon} \mathcal{L}(f^*(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \frac{2\varepsilon n_\varepsilon}{n},$$

and also lower training and test errors. This is a result of excluding the most uncertain examples $|\ell_i^*| < \varepsilon$ from learning. Figure 10 illustrates these trends on three learning problems. In particular it shows the thresholded empirical risk $\frac{1}{n} \sum_{i:|\ell_i^*| \geq \varepsilon} \mathcal{L}(f^*(\mathbf{x}_i), \text{sgn}(\ell_i^*)) + \frac{2\varepsilon n_\varepsilon}{n}$ of the optimal max-margin graph cut f^* (3.9), its training and test errors, and the percentage

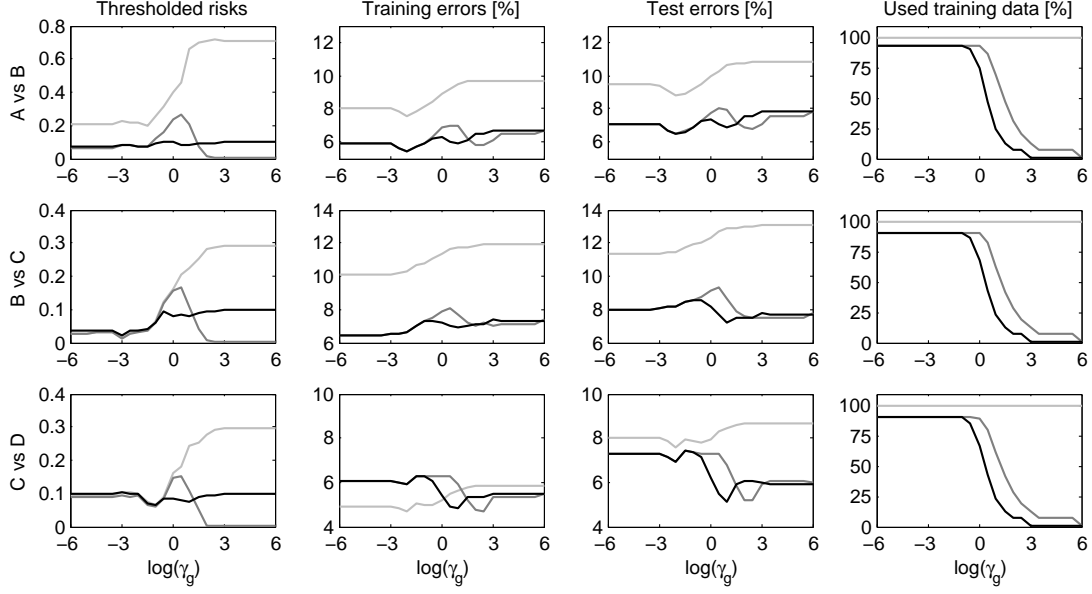


Figure 10: The thresholded empirical risk

of training examples such that $|\ell_i^*| \geq \varepsilon$, on 3 letter recognition problems from the UCI ML repository. The plots are shown as functions of the parameter γ_g , and correspond to the thresholds $\varepsilon = 0$ (light gray lines), $\varepsilon = 10^{-6}$ (dark gray lines), and $\varepsilon = 10^{-3}$ (black lines). All results are averaged over 50 random choices of 1 percent of labeled examples.

Note that the parameters γ_g and ε are redundant in the sense that the same result is often achieved by different combinations of parameter values. This problem is addressed in the experimental section by fixing ε and optimizing γ_g only.

4.3 ANALYSIS OF JOINT QUANTIZATION AND LABEL PROPAGATION

In this section we analyze our method of jointly optimizing for the backbone graph and the harmonic solution (Section 3.2.2) by showing its connection to principal manifold approach. One interesting property of the objective function in (3.13) for learning the centroids is that it has a similar form to the objective function of Elastic net model [Gorban and Zinovyev,

2009]. Elastic net is well-known technique based on an analogy between principle manifold and elastic membrane. It is a fast approximation of the principle manifolds and produces results similar to Kohonen's self-organized maps (SOM) [Haykin, 1994]. Given a set of initial centroids and a given connectivity between the centroids (just like SOM), Elastic net has the following form:

$$U = \gamma_q \sum_{i \in K_j} \|x_i - c_j\|^2 + \sum_{i,j \in \tilde{G}} \lambda_{ij} \|c_i - c_j\|^2 + \sum_{i,j,k \in \tilde{G}} \mu_{ijk} \|c_i + c_k - 2c_j\|^2 \quad (4.10)$$

where \tilde{G} is the graph connectivity between centroids and is assumed to be given. The objective function of the Elastic net model consist of three terms: the k-means term $U^Y = \gamma_q \sum_{i \in K_j} \|x_i - c_j\|^2$, the term $U^E = \sum_{i,j \in \tilde{G}} \lambda_{ij} \|c_i - c_j\|^2$ for stretching elasticity, and the term $U^R = \sum_{i,j,k \in \tilde{G}} \mu_{ijk} \|c_i + c_k - 2c_j\|^2$ for bending elasticity. λ_{ij} and μ_{ijk} are respectively, the coefficients of stretching elasticity of edge between nodes i and i and the coefficients of bending elasticity of edge between nodes i, j , and k .

Notice that U^Y is equivalent to the quantization penalty (3.11) for $\gamma_q = 1$. Moreover, if we set $\lambda_{ij} = -(l_i - l_j)^2 / 2\sigma^2$ then U^E approximates $\ell^\top L^C \ell$. Therefore, the objective function in (3.13) is the Elastic net with no bending term and with stretching coefficients dependent on the labels of the centroids; if the labels of two centroids are similar, the objective function tries to keep them close to each other and if the labels of two centroids are different, the objective function keeps them apart.

4.4 ANALYSIS OF ONLINE SSL ON QUANTIZED GRAPHS

In the rest of this section, W denotes full data similarity matrix, W_t^o its observed portion up to time t and W_t^q the quantized version of W_t^o . For simplicity, we do not consider the compact version of quantized matrix. In other words, W_t^q is $t \times t$ matrix with at most k distinct rows/columns. The Laplacians and regularized Laplacians of these matrices are denoted as L, L^o, L^q and K, K^o, K^q respectively. Similarly, we use ℓ^* , $\ell^o[t]$, and $\ell^q[t]$ to refer to the harmonic solutions on W , W_t^o , and W_t^q respectively. Finally, ℓ_t^* , $\ell_t^o[t]$, and $\ell_t^q[t]$ refer

to the predicted label of the example \mathbf{x}_t .

In this section, we use a stability argument to bound quality of the predictions. We note that the derived bounds are not tight. Our online learner (Figure 3) solves an online regression problem. As a result, it should ideally minimize the error of the form $\sum_t (\ell_t^q[t] - y_t)^2$, where $\ell_t^q[t]$ is the prediction at the time step t (again, time is denoted in the square brackets). In the following proposition we decompose this error into three terms. The first term (4.11) corresponds to the generalization error of the HS and is bounded by the algorithm stability argument. The second term (4.12) appears in our online setting because the similarity graph is only partially revealed. Finally, the third term (4.13) quantifies the error introduced due to quantization of the similarity matrix.

Proposition 4. *Let $\ell_t^q[t]$, $\ell_t^o[t]$, ℓ_t^* be the predictions as defined above and let y_t be the true labels. Then the error of our predictions $\ell_t^q[t]$ is bounded as*

$$\frac{1}{n} \sum_{t=1}^n (\ell_t^q[t] - y_t)^2 \leq \frac{9}{2n} \sum_{t=1}^n (\ell_t^* - y_t)^2 \quad (4.11)$$

$$+ \frac{9}{2n} \sum_{t=1}^n (\ell_t^o[t] - \ell_t^*)^2 \quad (4.12)$$

$$+ \frac{9}{2n} \sum_{t=1}^n (\ell_t^q[t] - \ell_t^o[t])^2. \quad (4.13)$$

Proof: Our bound follows from the inequality

$$(a - b)^2 \leq \frac{9}{2} [(a - c)^2 + (c - d)^2 + (d - b)^2],$$

which holds for $a, b, c, d \in [-1, 1]$. ■

We continue by bounding all the three sums in the Proposition 4. These sums can be bounded if the constraints $\ell_i = y_i$ are enforced in a soft manner [Cortes et al., 2008]. One way of achieving this is by solving a related problem

$$\min_{\ell \in \mathbb{R}^n} (\ell - \mathbf{y})^\top C(\ell - \mathbf{y}) + \ell^\top K \ell,$$

where $K = L + \gamma_g I$ is the regularized Laplacian of the similarity graph, C is a diagonal matrix such that $C_{ii} = c_l$ for all labeled examples, and $C_{ii} = c_u$ otherwise, and \mathbf{y} is a vector of pseudo-targets such that y_i is the label of the i -th example when the example is labeled, and $y_i = 0$ otherwise.

4.4.1 Bounding Transduction Error (4.11)

The following proposition bounds the generalization error of the solution to the problem (2.5). We then use it to bound the HS part (4.11) of Proposition 4.

Proposition 5. *Let ℓ^* be a solution to the problem (2.5), where all labeled examples l are selected i.i.d. If we assume that $c_l = 1$ and $c_l \gg c_u$, then the inequality*

$$R(\ell^*) \leq \underbrace{\widehat{R}(\ell^*) + \beta + \sqrt{\frac{2\ln(2/\delta)}{n_l}}(n_l\beta + 4)}_{\text{transductive error } \Delta_T(\beta, n_l, \delta)}$$

$$\beta \leq 2 \left[\frac{\sqrt{2}}{\gamma_g + 1} + \sqrt{2n_l} \frac{1 - \sqrt{c_u}}{\sqrt{c_u}} \frac{\lambda_M(L) + \gamma_g}{\gamma_g^2 + 1} \right]$$

holds with probability $1 - \delta$, where

$$R(\ell^*) = \frac{1}{n} \sum_t (\ell_t^* - y_t)^2 \text{ and } \widehat{R}(\ell^*) = \frac{1}{n_l} \sum_{t \in l} (\ell_t^* - y_t)^2$$

are risk terms for all and labeled vertices, respectively, and β is the stability coefficient of the solution ℓ^* .

The proof can be found in Section 4.2.2. Proposition 5 shows that when $\Delta_T(\beta, n_l, \delta) = o(1)$, the true risk is not much different from the empirical risk on the labeled points which bounds the generalization error. This occurs when $\beta = o(n_l^{-1/2})$, which corresponds to setting $\gamma_g = \Omega(n_l^{1+\alpha})$ for any $\alpha > 0$.

4.4.2 Bounding Online Error (4.12)

In the following, we will bound the difference between the online and offline HS and use it to bound (4.12) of the Proposition 4. The idea is that when Laplacians L and L^0 are regularized enough by γ_g , resulting harmonic solutions are close to zero and therefore close to each other. We first show that any regularized HS can be bounded as follows:

Lemma 3. *Let ℓ be a regularized harmonic solution, i.e. $\ell = (C^{-1}K + I)^{-1}\mathbf{y}$ where $K = L + \gamma_g I$.*

Then

$$\|\ell\|_2 \leq \frac{\sqrt{n_l}}{\gamma_g + 1}.$$

Proof: If $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix and $\lambda_m(A)$ and $\lambda_M(A)$ are its smallest and largest eigenvalues, then for any $\mathbf{v} \in \mathbb{R}^{n \times 1}$, $\lambda_m(A)\|\mathbf{v}\|_2 \leq \|A\mathbf{v}\|_2 \leq \lambda_M(A)\|\mathbf{v}\|_2$. Then

$$\|\ell\|_2 \leq \frac{\|\mathbf{y}\|_2}{\lambda_m(C^{-1}K + I)} = \frac{\|\mathbf{y}\|_2}{\frac{\lambda_m(K)}{\lambda_M(C)} + 1} \leq \frac{\sqrt{n_l}}{\gamma_g + 1}. \quad \blacksquare$$

The straightforward implication of Lemma 3 is that any 2 regularized harmonic solutions can be bounded as in the following proposition:

Proposition 6. *Let $\ell^o[t]$ be the predictions of the online HS, and ℓ^* be the predictions of the offline HS. Then*

$$\frac{1}{n} \sum_{t=1}^n (\ell_t^o[t] - \ell^*[t])^2 \leq \frac{4n_l}{(\gamma_g + 1)^2}. \quad (4.14)$$

Proof: We use the fact that $\|\cdot\|_2$ is an upper bound on $\|\cdot\|_\infty$. Therefore, for any t

$$\begin{aligned} (\ell_t^o[t] - \ell_t^*)^2 &\leq \|\ell^o[t] - \ell^*\|_\infty^2 \leq \|\ell^o[t] - \ell^*\|_2^2 \\ &\leq \left(\frac{2\sqrt{n_l}}{\gamma_g + 1} \right)^2, \end{aligned}$$

where in the last step we used Lemma 3 twice. By summing over n and dividing by n we get (4.14). \blacksquare

From Proposition 6 we see that we can achieve convergence of the term (4.12) at the rate of $O(n^{-1/2})$ with $\gamma_g = \Omega(n^{1/4})$.

4.4.3 Bounding Quantization Error (4.13)

In this section, we show in Proposition 7 a way to bound the error for the HS between the full and quantized graph, and then use it to bound the difference between the *online* and *online quantized* HS in (4.13). Let us consider the perturbed version of the problem (2.5), where we replace the regularized Laplacian K^o with K^q ; i.e., K^q corresponds to the regularized Laplacian of the quantized graph. Let ℓ^o and ℓ^q minimize (2.5) and its perturbed version respectively. Their closed-form solutions are given by $\ell^o = (C^{-1}K^o + I)^{-1}\mathbf{y}$ and $\ell^q = (C^{-1}K^q + I)^{-1}\mathbf{y}$ respectively. We now follow the derivation of [Cortes et al., 2008] that derives stability coefficient of unconstrained regularization algorithms. Instead of considering perturbation

on C , we consider the perturbation on K^0 . Our goal is to derive a bound on a difference in HS when we use K^q instead of K^0 .

Lemma 4. *Let ℓ^0 and ℓ^q minimize (2.5) and its perturbed version respectively. Then*

$$\|\ell^q - \ell^0\|_2 \leq \frac{\sqrt{n_l}}{c_u \gamma_g^2} \|K^q - K^0\|_F.$$

Proof: Let $Z^q = C^{-1}K^q + I$ and $Z^0 = C^{-1}K^0 + I$. By definition

$$\begin{aligned} \ell^q - \ell^0 &= (Z^q)^{-1}\mathbf{y} - (Z^0)^{-1}\mathbf{y} = (Z^q Z^0)^{-1}(Z^0 - Z^q)\mathbf{y} \\ &= (Z^q Z^0)^{-1}C^{-1}(K^0 - K^q)\mathbf{y}. \end{aligned}$$

Using the eigenvalue inequalities from the proof of Lemma 3 we get

$$\|\ell^q - \ell^0\|_2 \leq \frac{\lambda_M(C^{-1})\|(K^q - K^0)\mathbf{y}\|_2}{\lambda_m(Z^q)\lambda_m(Z^0)}. \quad (4.15)$$

By the compatibility of $\|\cdot\|_F$ and $\|\cdot\|_2$ and since \mathbf{y} is zero on unlabeled points, we have

$$\|(K^q - K^0)\mathbf{y}\|_2 \leq \|K^q - K^0\|_F \cdot \|\mathbf{y}\|_2 \leq \sqrt{n_l}\|K^q - K^0\|_F.$$

Furthermore,

$$\lambda_m(Z^0) \geq \frac{\lambda_m(K^0)}{\lambda_M(C)} + 1 \geq \gamma_g \quad \text{and} \quad \lambda_M(C^{-1}) \leq c_u^{-1},$$

where c_u is a small constant as defined in (2.5). By plugging these inequalities into (4.15) we get the desired bound. ■

Proposition 7. *Let $\ell_t^q[t]$ be the predictions of the online harmonic solution on the quantized graph at the time step t , $\ell_t^0[t]$ be predictions of the online harmonic solution at the time step t . Then*

$$\frac{1}{n} \sum_{t=1}^n (\ell_t^q[t] - \ell_t^0[t])^2 \leq \frac{n_l}{c_u^2 \gamma_g^4} \|L^q - L^0\|_F^2. \quad (4.16)$$

Proof: Similarly as in Proposition 6, we get

$$\begin{aligned} (\ell_t^q[t] - \ell_t^o[t])^2 &\leq \|\ell_t^q[t] - \ell_t^o[t]\|_\infty^2 \leq \|\ell_t^q[t] - \ell_t^o[t]\|_2^2 \\ &\leq \left(\frac{\sqrt{n_l}}{c_u \gamma_g^2} \|K^q - K^o\|_F \right)^2, \end{aligned}$$

where we used (4.15) the last step. We also note that

$$K^q - K^o = L^q + \gamma_g I - (L^o + \gamma_g I) = L^q - L^o,$$

which gives us $(\ell_t^q[t] - \ell_t^o[t])^2 \leq \|L^q - L^o\|_F^2 \cdot n_l / (c_u^2 \gamma_g^4)$. By summing over n and dividing by n we get (4.16). ■

If $\|L^q - L^o\|_F^2 = O(1)$, the left-hand side of (4.16) converges to zero at the rate of $O(n^{-1/2})$ with $\gamma_g = \Omega(n^{1/8})$. We show this condition is achievable whenever the Laplacian is scaled appropriately. Specifically, we demonstrate that normalized Laplacian achieves this bound when the quantization is performed using incremental k -center clustering in Section 3.3, and when the weight function obeys a Lipschitz condition (e.g. the Gaussian kernel). We also show that this error goes to zero as the number of center points k goes to infinity. This result is directly applicable to unnormalized Laplacian used in previous sections, with details being omitted due to space limitation.

Suppose the data $\{\mathbf{x}_i\}_{i=1,\dots,n}$ lie on a smooth d -dimensional compact manifold \mathcal{M} with boundary of bounded geometry embedded in \mathbb{R}^b . We first demonstrate that the distortion introduced by quantization is small, and then show that small distortion gives small error in the Frobenius norm.

Proposition 8. *Using incremental k -center clustering for quantization has maximum distortion $Rm/(m-1) = \max_{i=1,\dots,n} \|\mathbf{x}_i - \mathbf{c}\|_2 = O(k^{-1/d})$, where \mathbf{c} is the closest centroid to \mathbf{x}_i .*

Proof: Consider a sphere packing with k centers contained in \mathcal{M} and each with radius r . Since the manifold is compact and the boundary has bounded geometry, it has finite volume V and finite surface area A . The maximum volume that the packing can occupy obeys the inequality $k c_d r^d \leq V + A c_{\mathcal{M}} r$ for some constants $c_d, c_{\mathcal{M}}$ that only depend on the dimension and the manifold. Trivially, if k is sufficiently large, then $r < 1$, and we have an upper bound $r < ((V + A c_{\mathcal{M}})/(k c_d))^{1/d} = O(k^{-1/d})$. An r -packing is a $2r$ -covering, so we have an upper

bound on the distortion of the optimal k -centers solution. Since the incremental k -centers algorithm is a $(1 + \epsilon)$ -approximation algorithm [Charikar et al., 1997], it follows that the maximum distortion returned by the algorithm is $Rm/(m - 1) = 2(1 + \epsilon)O(k^{-1/d})$. ■

We now show that with appropriate normalization, the error $\|L^q - L^o\|_F^2 = O(k^{-2/d})$. If L^q and L^o are normalized Laplacians, then this bound holds if the underlying density is bounded away from 0. Note that since we use the Gaussian kernel, the Lipschitz condition is satisfied.

Proposition 9. *Let W_{ij}^o be a weight matrix constructed from $\{x_i\}_{i=1,\dots,n}$ and a bounded, Lipschitz function $\omega(\cdot, \cdot)$ with Lipschitz constant M . Let D^o be the corresponding degree matrix and $L_{ij}^o = (D_{ij}^o - W_{ij}^o)/c_{ij}^o$ be the normalized Laplacian. Suppose $c_{ij}^o = \sqrt{D_{ii}^o D_{jj}^o} > c_{\min} n$ for some constant $c_{\min} > 0$ that does not depend on k . Likewise define W^q, L^q, D^q on the quantized points. Let the maximum distortion be $Rm/(m - 1) = O(k^{-1/d})$. Then $\|L^q - L^o\|_F^2 = O(k^{-2/d})$.*

Proof: Since ω is Lipschitz, we have that $|W_{ij}^q - W_{ij}^o| < 2MRm/(m - 1)$ and $|c_{ij}^q - c_{ij}^o| < 2nMRm/(m - 1)$. The error of a single off-diagonal entry of the Laplacian matrix is

$$\begin{aligned} L_{ij}^q - L_{ij}^o &= \frac{W_{ij}^q}{c_{ij}^q} - \frac{W_{ij}^o}{c_{ij}^o} \\ &\leq \frac{W_{ij}^q - W_{ij}^o}{c_{ij}^q} + \frac{W_{ij}^q(c_{ij}^q - c_{ij}^o)}{c_{ij}^o c_{ij}^q} \\ &\leq \frac{4MRm}{(m - 1)c_{\min} n} + \frac{4M(nMRm)}{((m - 1)c_{\min} n)^2} \\ &= O\left(\frac{R}{n}\right). \end{aligned}$$

The error on the diagonal entries is 0 since the diagonals of L^q and L^o contain all 1. Thus $\|L^q - L^o\|_F^2 \leq n^2 O(R^2/n^2) = O(k^{-2/d})$. ■

Here we showed the asymptotic behavior $\|L^q - L^o\|_F$ in term of the number of vertices used in the quantized graph. In Section 5.3.3.1, we empirically show that $\|L^q - L^o\|_F$ vanishes quickly as the number of vertices increases (Figure 20). Moreover, with fixed number of vertices, $\|L^q - L^o\|_F$ quickly flattens out even when the data size (time) keeps increasing (Figure 19).

4.4.4 Discussion

Our goal in this section is to show how much of regularization γ_g is needed for error of our predictions to reasonably decrease over time. We point out that in Proposition 1 the lower bound for γ_g for reasonable convergence is a function of n_l labeled examples. On the other hand, in Propositions 6 and 7 those lower bounds are the functions of all n examples.

In particular, Proposition 1 requires $\gamma_g = \Omega(n_l^{1+\alpha})$, $\alpha > 0$ for the true risk not to be much different from the empirical risk on the labeled points. Next, Propositions 6 and 7 require $\gamma_g = \Omega(n^{1/4})$ and $\gamma_g = \Omega(n^{1/8})$ respectively for the terms (4.12) and (4.13) to be $O(n^{-1/2})$.

For many applications of online SSL, a small set of n_l labeled example is given in advance, the rest of the examples are unlabeled. That means we usually expect $n \gg n_l$. Therefore, if we regard n_l as a constant, we need to regularize as much as $\gamma_g = \Omega(n^{1/4})$. For such a setting of γ_g we have that for n approaching infinity the error of our predictions is getting close to the empirical risk on labeled examples with the rate of $O(n^{-1/2})$.

4.5 PARALLEL MULTI-MANIFOLD LEARNING

In this section we analyze the approximation proposed in Section 3.4, when instead of computing the harmonic solution (HS) on the whole graph, we

1. decompose the graph into several smaller subgraphs,
2. compute the HSs on the smaller graphs *in parallel*, and
3. aggregate the partial HSs.

In the ideal case, the similarity matrix has a block-diagonal (BD) structure, which corresponds to the graph with disconnected components. In this case, such an approximation is exact. Since the harmonic solution for n nodes of the graph has computational complexity of $\mathcal{O}(n^3)$, the time savings can be significant (Section 4.5).

In the rest of this section we analyze the general case, when the similarity matrix does not have BD structure. Intuitively, the closer the similarity matrix resembles BD structure, the smaller decrease in prediction accuracy we expect.

Again, if similarity and its Laplacian are BD, then HS calculated per block and as a whole are identical (even with the regularization), because it can be rewritten as solving two independent systems of linear equations. On the other hand, an *impurity* of BD structure can change HS a lot (think of the case when we merge blocks with labeled examples from different classes). We continue by extending the analysis in Section 4.4 and follow Proposition 7:

Lemma 5. *Let ℓ^o and ℓ^q minimize (2.5) and its perturbed version respectively. Then*

$$\|\ell^q - \ell^o\|_2 \leq \frac{\sqrt{n_l}}{c_u \gamma_g^2} \|K^q - K^o\|_F.$$

Proof is in Section 4.4.3. The question now is how to bound $\|K^q - K^o\|_F$ or $\|L^q - L^o\|_F$ if same regularization is used. Let L^{bd} denotes general block-diagonal approximation of L^q where the entries outside the BD structure are ignored (ie. are assumed to be zero). Then

$$\|L^{\text{bd}} - L^o\|_F \leq \|L^{\text{bd}} - L^q\|_F + \|L^q - L^o\|_F. \quad (4.17)$$

Let d_{\max} is the value of the maximum entry in K^q which is ignored when the approximation is performed. In general case for BD setting, we can have $n^2/2$ to n^2 ignored entries. Therefore,

$$n \sqrt{d_{\max}/2} \leq \|L^{\text{bd}} - L^q\|_F \leq n \sqrt{d_{\max}}. \quad (4.18)$$

This approximation adds factor of $\Theta(n)$ to the quantization bound (Section 4.4.3). To maintain the overall convergence of $O(n^{-1/2})$ we need to have $\gamma_g = \Omega(n^{3/8})$, along with the discussion in Section 4.4.4.

5.0 EXPERIMENTS

This chapter presents the set of experiments we performed for semi-supervised learning (SSL) and conditional anomaly detection (CAD). In Section 5.1, we describe the data we used in our experiments. We used medical, vision, UCI ML repository and synthetic datasets. In Section 5.2, we describe the competing algorithms we used for the comparison and base lines. Finally, we present our SSL results in Section 5.3 and our CAD results in Section 5.4.

5.1 DATASETS

5.1.1 Post-surgical cardiac patients (PCP)

Post-surgical cardiac patients (PCP) database is a database of de-identified records for 4486 post-surgical cardiac patients treated at one of the University of Pittsburgh Medical Center (UPMC) teaching hospitals. The entries in the database were populated from data from the MARS¹ system, which serves as an archive for much of the data collected at UPMC. The records for individual patients included discharge records, demographics, progress notes, all labs and tests (including standard and all special tests), two medication databases, microbiology labs, EKG, radiology and special procedures reports, and a financial charges database. The data in PCP database were cleaned, cross-mapped and stored in a local MySQL database with protected access. In this thesis, we worked with subset of these data: a cohort of 4486 patients that underwent cardiac surgery from 2002 to 2007.

¹MARS stands for Medical Archival System is medical record system that has been storing clinical and financial information from UMPC since 1980.

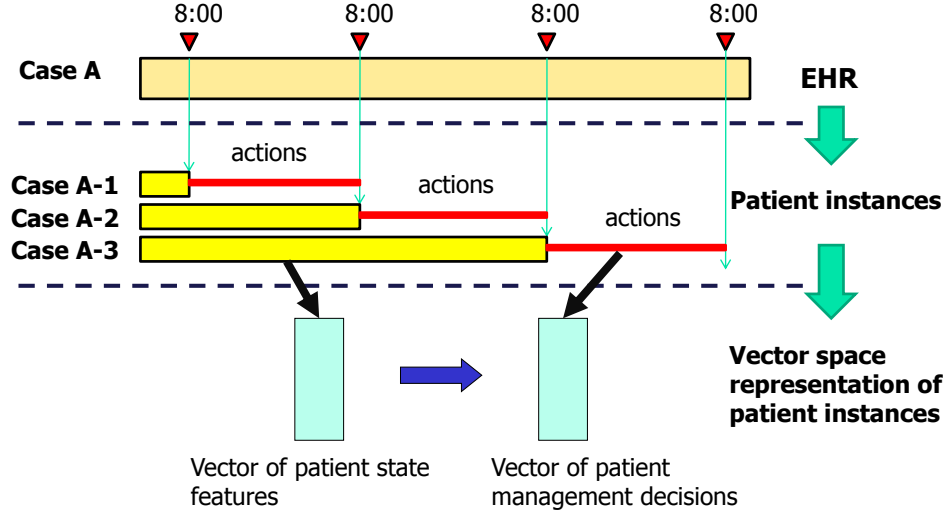


Figure 11: Processing of data in the electronic health record

The database is very heterogeneous and has many variables in different formats. It has also a fair amount of missing data.

The EHRs were first divided into two groups: a training set that included 2646 cases, and a test set that included 1840 cases. We used the time-stamped data in each EHR to segment the record at 8:00am every day to obtain multiple patient case instances, as illustrated in Figure 11: (1) segmentation of an EHR into multiple patient-state/decision instances, (2) transformation of these instances into a vector space representation of patient states and their follow-up decisions. Figure 12 illustrates a subset of features generated for labs. Figure 12 displays following features for lab test. Last value: A , Last value difference = $B - A$, Last % change = $(B - A)/B$, Last slope = $(B - A)/(t_B - t_A)$, Nadir = D , Nadir difference = $A - D$, Nadir % difference = $(A - D)/D$. Lab tests will be described in more detail in Section 5.1.1.2. The segmentation led to 51,492 patient-state instances, such that 30,828 were used for training the model and 20,664 were used in the evaluation.

To represent a patient state we have adopted a vector space representation that is suitable for machine learning approaches. In this representation a patient state is represented by a set of features characterizing the patient at a specific point in time and their corre-

sponding feature values. Features represent and summarize the information in the medical record such as last blood glucose measurement, last glucose trend, or the time the patient is on heparin.

The features used in our experiment were generated from time series associated with different clinical variables, such as blood glucose measurement, platelet measurement, Amiodarone medication. The clinical variables used in this study were grouped into five categories:

1. Laboratory tests (LABs)
2. Medications (MEDs)
3. Visit features/demographics
4. Procedures
5. Heart support devices

Altogether, our dataset consists of 9,223 different features. We now briefly describe the features generated for clinical variables in each of these categories.

5.1.1.1 Visit/Demographic Features We only have 3 features in this category: age, sex and race. These are static and same for every time point we generate.

5.1.1.2 Lab Features For the categorical labs, for example the ones with POS/NEG results we use the following features: Last value; second last value; first value; time since last order; is the order pending; is the value know; is the trend known. For the labs with continuous or ordinal values we use a richer set of features including features as difference between last two values, slope of last 2 values, and their percentage drop/increase. We used the same kind of features for the following pairs of lab values (last value, first value), (last value, nadir value), (last value, horizon value). Nadir and horizon value are the lab values with the smallest and the greatest value recorded up to that point. Figure 12 illustrates a subset of features generated for labs with continuous values. The total number of features generated for such a lab is 40. We list some of the features here:

- Last value: A

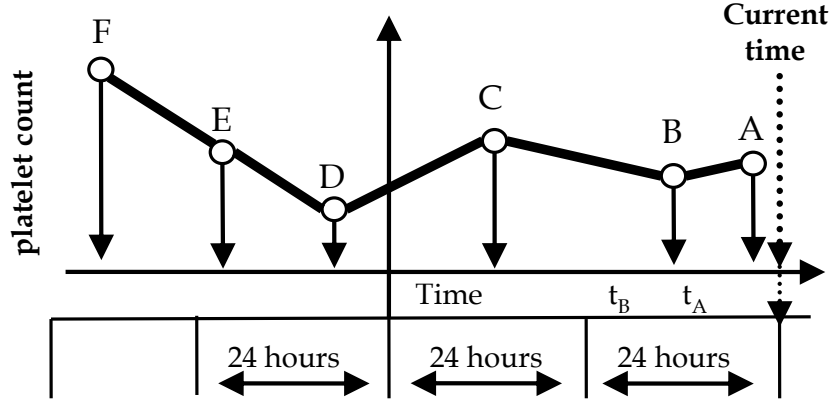


Figure 12: Examples of temporal features for continuous lab values

- Last value difference = $B-A$
- Last percentage change = $(B-A)/B$
- Last slope = $(B-A) / (t_B-t_A)$
- Nadir = D
- Nadir difference = $A-D$
- Nadir percentage difference = $(A-D)/D$
- Baseline = F
- Drop from baseline = $F-A$
- Percentage drop from baseline = $(F-A)/F$
- 24 hour average = $(A+B)/2$

5.1.1.3 Medications For each medication we used four features: 1) indicator if the patient is currently on the medication, 2) time since the patient was put on that medication for the first time, 3) time since the patient was last on that medication, and 4) time since last change in the order of the medication.

5.1.1.4 Procedure Features Procedure feature capture the information about procedures such as *Heart valve repair* that were performed either in operating room (OR) or

at the bedside. In our data we distinguish 36 different procedures that are performed on cardiac patients. We record four features per procedure: 1) time since the procedure was done last time 2) time since the procedure was done first time 3) indicator of whether the procedure was done in last 24 hours and 4) indicator of if the procedure was done.

5.1.1.5 Heart Support Device Features Finally we describe the status of 4 different heart support devices: extra-corporeal membrane oxygenation (ECMO), balloon counter pulsation, pacemaker and other heart assist device. For each of them we record a single feature which describes whether the device is currently used to support patient's heart function.

5.1.1.6 Orders/Labels Labels in this case correspond to patient-management decisions. In addition to feature generation, every patient state example in the dataset that was generated by the above segmentation process was linked to lab order and medication decisions that were made for the patient within next 24 hours. Patient management decisions considered were:

- lab order decisions with (true/false) values reflecting whether the lab was ordered within the next 24 hours or not
- medication decisions with (true/false) values reflecting if the patient was given a medication within the next 24 hours or not.

A total of 335 lab order and 407 medication decision values were recorded and linked to every patient state example in the dataset.

5.1.2 UCI ML Datasets

We used the following datasets from UCI ML Repository [[Asuncion and Newman, 2011](#)]:

5.1.2.1 Digit recognition This dataset was preprocessed by programs made available by NIST to extract normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are

counted in each block. This generates an input matrix of 8×8 where each element is an integer in the range 0..16. This reduces dimensionality and gives invariance to small distortions.

5.1.2.2 Letter recognition The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15.

5.1.2.3 Image segmentation The Segmentation dataset, created in 1990 by the Vision Group, University of Massachusetts, consists of 2310 instances. Each instance was drawn randomly from a database of seven outdoor images. The image, a 3×3 region, was hand-segmented to create a classification for each pixel. The seven classes are brickface, sky, foliage, cement, window, path, and grass. Each of the 7 images is represented by 330 instances. The extracted features are 19 continuous attributes that describe the position of extracted image, line densities, edges, and color values.

5.1.2.4 Coil The Columbia object image library (COIL-100) is a set of color images of 100 different objects taken from different angles (in steps of 5 degrees) at a resolution of 128×128 pixels [Nene et al., 1996]. We used the binary version of this dataset as preprocessed by [Chapelle et al., 2006].

5.1.2.5 Car Car evaluation data set classifying cars into four categories using 6 features including buying price, number of doors etc. We converted the Car dataset to a binary problem to classify first two vs. second two car categories.

5.1.2.6 SecStr SecStr is a benchmark data set designed by [Chapelle et al., 2006] to investigate how far current methods can cope with large-scale application. The task is to

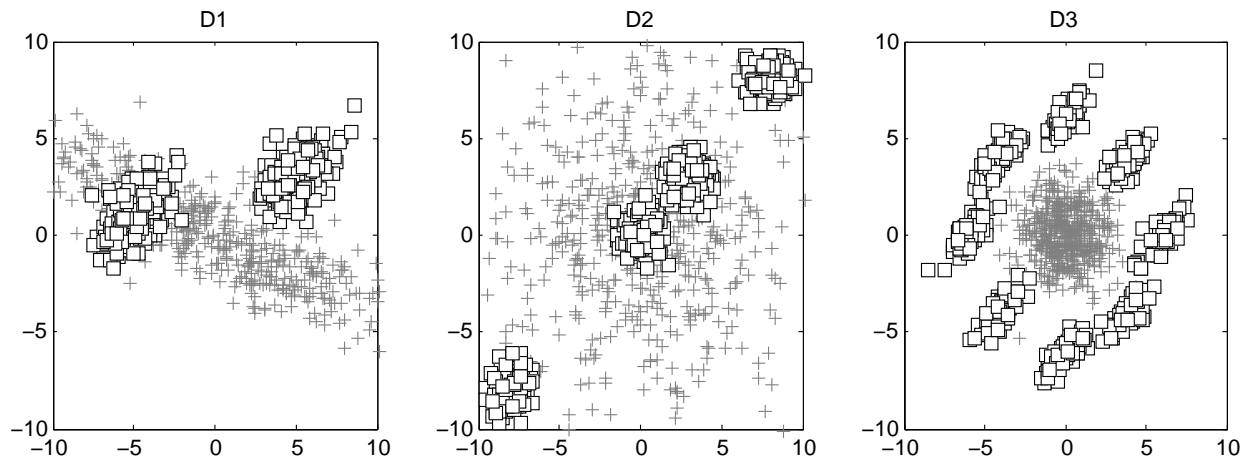


Figure 13: The three synthetic datasets with known underlying distributions

predict the secondary structure of a given amino acid in a protein based on a sequence window centered around that amino acid.

For the multi-class datasets we sometimes transformed them to a set of binary problems.

5.1.3 Synthetic Datasets

5.1.3.1 Core Dataset Inspired by [Papadimitriou and Faloutsos, 2003], we generated a synthetic *Core* dataset which consists of two overlapping squares from two uniform distributions. We extended this dataset with two tiny squares (Figure 24, top left). These 2 tiny squares may be considered anomalous but not conditionally anomalous. The goal is to detect 12 conditional anomalies which are located in the middle square (Figure 24, top middle). We also used this dataset to demonstrate the challenges for conditional anomaly detectors, namely fringe and isolated points.

5.1.3.2 Mixtures of Gaussians We generated three synthetic datasets (D1, D2, and D3) with known underlying distributions that let us compute the true anomaly scores.

We show the three datasets we used in our experiments in Figure 13. Each dataset

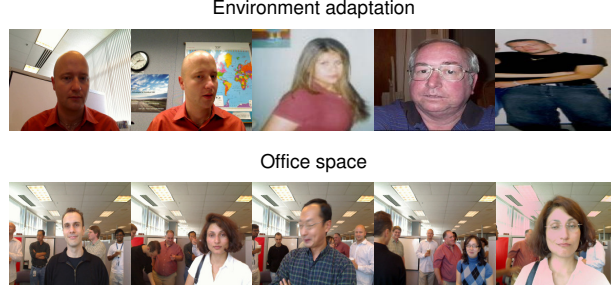


Figure 14: Snapshots from the environment adaptation and office space datasets

consists of equal number of samples from the class +1 and class -1. The class densities we used to generate these datasets are modeled with mixtures of multivariate Gaussians and vary in locations, shapes and mutual overlaps.

5.1.4 Vision Datasets

The *environment adaptation* dataset consists of faces of a single person, which are captured at various locations, such as a cubicle, a conference room, and the corner with a couch (Figure 14). The first four faces in the cubicle are labeled and we want to learn a face recognizer for all locations. To test the sensitivity of the recognizer to outliers, we appended the dataset by random faces. The *office space* dataset (Figure 14) is multi-class, and involves 8 people who walk in front of a camera and make funny faces. When a person shows up on the camera for the first time, we label four faces of the person. Our goal is to learn good face recognizers for all 8 people.

Another vision dataset is a *face-based authentication* dataset of 16 people (Figure 15). The people try to log into a tablet PC with their face while being recorded by its embedded camera. The data are collected at 10 indoor locations, which differ by backgrounds and lighting conditions. In short, we recorded 20 10-second videos per person, each at 10 fps. Therefore, our face-based authentication dataset contains a total of $16 \times 20 = 32000$ images. Faces in the images are detected using OpenCV [Bradski, 2000], converted to grayscale,

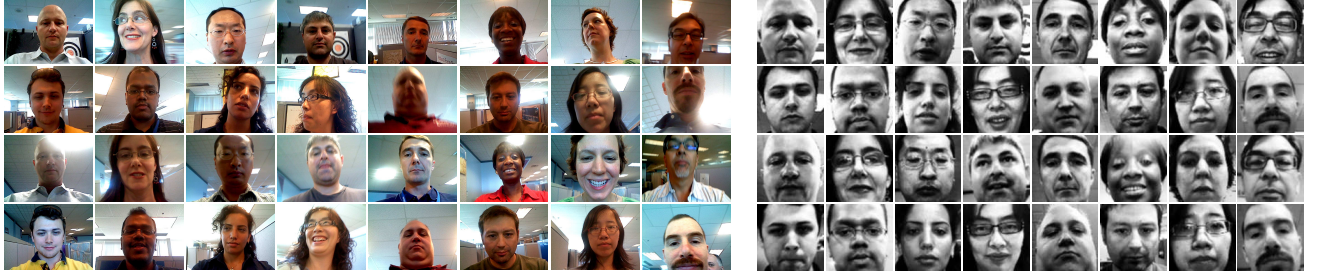


Figure 15: Face-based authentication dataset (left) and examples of labeled faces (right).

resized to 96×96 , smoothed using the 3×3 Gaussian kernel, and equalized by the histogram of their pixel intensities.

5.2 ALGORITHMS FOR COMPARISON

In this section we review the algorithms chosen for the comparison.

5.2.1 Discriminative SVM Anomaly Detection

For the baseline method we use an SVM based method [Valko et al., 2008, Hauskrecht et al., 2010], that computes anomaly score from the distance from the hyperplane. SVM [Vapnik, 1995, Burges, 1998] is a discriminative method that learns the decision boundary as

$$\mathbf{w}^T \mathbf{x} + w_0 = \sum_{i \in SV} \hat{\alpha}_i y_i (\mathbf{x}_i^T \mathbf{x}) + w_0,$$

where only samples in the support vector set (SV) contribute to the computation of the decision boundary. To support classification tasks, the projection defining the decision boundary is used to determine the class of a new example. That is, if the value

$$\mathbf{w}^T \mathbf{x} + w_0 \geq 0$$

is positive then $C(\mathbf{x})$ belong to one class, if it is negative it belongs to the other class. However, in our for conditional anomaly detection we use the projection itself for the positive class and the negated projection for the negative class to measure the deviation:

$$d(y|\mathbf{x}) = y(\mathbf{w}^T \mathbf{x} + w_0), \text{ where } y \in \{-1, 1\}$$

In other words, the smaller the projection is the more likely is the example anomalous. We note that the negative projections correspond to misclassified examples.

5.2.2 One-Class SVM

As an example of a classical anomaly detection method converted to the conditional anomaly detection method we compare to the one-class SVM [Manevitz and Yousef, 2002]. Originally proposed in [Scholkopf et al., 1999], the method only needs positive examples to learn the margin. The idea is that the space origin (zero) is treated as the only example as the ‘negative’ class. In that way the learning essentially estimates the support of the distribution. The data that do not fall into this support have negative projections and can be considered anomalous. In our scenario, we will learn one one-class SVM for each of the classes and based on the test label (which is known) we calculate the anomaly score. The more negative the score the higher the rank of the anomaly.

5.2.3 Quadratic discriminant analysis

In quadratic discriminant analysis (QDA) model [Hastie et al., 2001], where we model each class by a multivariate Gaussian, and the anomaly score is the class posterior of the opposite class.

5.2.4 Weighted NN

We also used weighted k -NN approach [Hastie et al., 2001] that uses the same weight metric W as SoftHAD, but relies on only on the labels in the local neighborhood and does not account for the manifold structure.

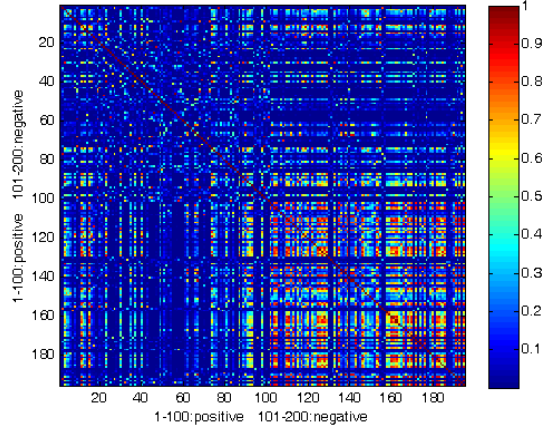


Figure 16: The weight matrix for 100 negative and 100 positive cases of HPF4 order

5.2.5 Parameters for the Graph-Based Algorithms

The similarity weights are computed as

$$w_{ij} = \exp \left[-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_{2,\psi}^2}{p\sigma^2} \right],$$

where p is the number of features, $\psi = (p \times 1)$ is a weighing of the features based on their discriminative power. Including p in the weight metric allows us to control the connectivity of the graph. Next, σ is chosen so that the graph is reasonably sparse [Luxburg, 2007]. We followed [Valizadegan and Tan, 2007] and chose σ as 10% of the mean of empirical standard deviations of all features. Based on the experiments, our algorithm is not sensitive to the small perturbations of σ , what is important is that the graph does not become disconnected by having all edges of several nodes with weights close to zero.

For the feature weights (ψ) for PCP data we used the univariate Wilcoxon (ROC) score [Hanley and Mcneil, 1982]. Since this score ranges from 0.5 to 1, we modify the score by subtracting 0.5 and raising it to the power of 5 to make the differences between the weights larger. We used the same metric for the weighted NN anomaly detection from Section 4.1.1. We varied the regularization parameter as $\lambda \in \{10^{-10}, 10^{-9}, \dots, 10^{-1}\}$ and chose the best one

on the held out validation set. Figure 16 illustrates this metric on a binary classification task for the heparin induced thrombocytopenia (a life threatening condition that may occur with prolonged heparin treatments). One hundred negative and one hundred positive cases and their mutual similarities are shown. We can see that positive cases are much closer to each other (bottom right corner) than negatives. For the other datasets, we used uniform (ψ) for all features.

5.3 EVALUATIONS OF SEMI-SUPERVISED LEARNING MODELS

In this section we evaluate the predictive performance of our graph-based model on semi-supervised tasks. Our goal is to demonstrate that graph-based method can yield good predictors. Having good predictive accuracy is important for CAD because we are typically more confident about anomalies from better predictive models.

5.3.1 Max-margin Graph Cuts Experiments

The experiments with max-margin graph cuts is divided into two parts. The first part compares max-margin graph cuts to manifold regularization of SVMs on the problem from Figure 3. The second part compares max-margin graph cuts, manifold regularization of SVMs, and supervised learning with SVMs on three UCI ML repository datasets [Asuncion and Newman, 2011]. Manifold regularization of SVMs is evaluated based on the implementation of [Belkin et al., 2006]. Max-margin graph cuts and SVMs are implemented using LIBSVM [Chang and Lin, 2001].

5.3.1.1 Synthetic Problem The first experiment (Figure 9) illustrates linear, cubic, and RBF graph cuts (3.9) on the synthetic problem from Figure 3. The cuts are shown for various settings of the regularization parameter γ_g . As γ_g decreases, note that the cuts gradually interpolate between supervised learning on just two labeled examples and semi-supervised learning on all data. The resulting discriminators are max-margin decision boundaries that

separate the corresponding colored regions in Figure 3.

Figure 9 also shows that manifold regularization of SVMs (2.8) with linear and cubic kernels cannot perfectly separate the two clusters in Figure 3 for any setting of the parameter γ_u . The reason for this problem is discussed in Section 4.2.1. Finally, note the similarity between max-margin graph cuts and manifold regularization of SVMs with the RBF kernel. This similarity was suggested in Section 2.4.1.2.

Dataset	L	Misclassification errors [%]								
		Linear kernel			Cubic kernel			RBF kernel		
		SVM	MR	GC	SVM	MR	GC	SVM	MR	GC
Letter recognition	1	18.90	30.94	15.79	20.54	25.96	17.45	20.06	17.61	16.01
	2	12.92	28.45	10.79	12.18	18.34	10.90	13.52	13.10	11.83
	5	8.21	27.13	5.65	5.49	18.77	4.80	6.81	8.06	5.65
	10	6.51	25.45	3.96	4.17	14.03	2.96	4.95	6.14	3.32
Digit recognition	1	7.06	9.59	6.88	9.62	5.29	8.55	8.22	6.36	7.65
	2	4.87	7.97	4.60	6.06	5.06	5.09	6.17	4.21	5.61
	5	2.97	3.68	2.29	3.04	2.27	2.36	2.74	2.29	2.19
	10	1.70	2.86	1.59	1.87	1.60	1.74	1.68	1.75	1.35
Image segmentation	1	14.02	11.81	10.27	23.30	12.02	14.10	14.02	11.60	9.51
	2	8.54	10.87	7.69	14.28	13.07	7.73	9.06	8.93	7.34
	5	4.73	7.83	4.49	8.32	8.79	7.17	5.87	5.43	5.31
	10	3.30	6.26	3.28	3.65	6.64	3.60	3.84	4.81	3.73

Figure 17: Comparison of SVMs, GC and MR on 3 datasets from the UCI ML repository

5.3.1.2 UCI ML Repository Datasets The second experiment (Figure 17) shows that max-margin graph cuts (3.9) typically outperform manifold regularization of SVMs (2.8) and supervised learning with SVMs. In particular it shows the comparison of SVMs, max-margin graph cuts (GC), and manifold regularization of SVMs (MR) on three datasets from the UCI ML repository. The fraction of labeled examples L varies from 1 to 10 percent.

The experiment is done on three UCI ML repository datasets: letter recognition, digit recognition, and image segmentation. The datasets are multi-class and thus, we transform each of them into a set of binary classification problems. The digit recognition and image segmentation datasets are converted into 45 and 15 problems, respectively, where all classes are discriminated against every other class. The letter recognition dataset is turned into 25

problems that involve pairs of consecutive letters. Each dataset is divided into three folds. The first fold is used for training, the second one for selecting the parameters $\gamma \in [0.01, 0.1]n_l$, $\gamma_u \in [10^{-3}, 10^3]\gamma$, and $\gamma_g = \gamma/\gamma_u$, and the last fold is used for testing.² The fraction of labeled examples in the training set is varied from 1 to 10 percent. All examples in the validation set are labeled and its size is limited to the number of labeled examples in the training set.

In all experiments, we use 5-nearest neighbor graphs whose edges are weighted as $w_{ij} = \exp[-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / (2K\sigma^2)]$, where K is the number of features and σ denotes the mean of their standard deviations. The width of radial basis functions (RBFs) is set accordingly to $\sqrt{K}\sigma$, and the threshold ε for choosing training examples (3.9) is 10^{-6} .

Test errors of all compared algorithms are averaged over all binary problems within each dataset and shown in Figure 17. Max-margin graph cuts outperform manifold regularization of SVMs in 29 out of 36 experiments. Note that the lowest errors are usually obtained for linear and cubic kernels, and our method improves the most over manifold regularization of SVMs in these settings.

5.3.2 Joint Quantization and Label Propagation Experiments

In this part, we evaluate the method we proposed in Section 3.2.2 that combines the creation of backbone graph with label propagation. The benefit of our algorithm comes when the data lies on a low dimensional manifold. In this section, we showcase the 2 data sets when this is the case. For data sets without a manifold structure or for the data sets where a cluster assumption holds, we expect the performance of our method be comparable to the case when k -means is used as a preprocessing step. We compare our algorithm to several quantization approaches:

1. *random subsampling*: We randomly sample k examples from the unlabeled data. Then we apply SSL method on the selected samples.
2. *k-means*: We cluster the unlabeled data using k -means [Hastie et al., 2001] to get k cluster centers and then apply SSL algorithms to get their labels.

²Alternatively, the regularization parameters γ , γ_u , and γ_g can be set using leave-one-out cross-validation on labeled examples.

3. *elastic nets*: We use elastic net [Gorban and Zinovyev, 2009] as a preprocessing to get k cluster centers. We then apply SSL to get their labels.
4. *elastic-joint*: We apply the proposed algorithm in this thesis to get both the centroids and their labels.
5. *full-soft*: We apply SSL algorithm on the full set of examples as a reference point.

After obtaining the labels of the centroids using items 1-4 above, we apply the approximation in Section 3.2.2.3 to get the labels for unlabeled examples.

5.3.2.1 Experimental setup We used a small subset of examples as labeled examples. To see the sensitivity of the method on different number of labeled examples, we tried $m = 2, 10, 20$ and 50 as the number of labeled examples. To allow for the fair comparison between the methods, we run all the algorithms on the same set of the labeled examples. Moreover, all the approximation methods are initialized with the same cluster centers as the ones that were drawn by the random subsampling.

Finally we fixed all the parameters for the semi-supervised prediction in Equation (2.5) to the same settings as follows. We created a 3-nearest neighbors similarity graph and we used the Gaussian kernel with the kernel width σ equal to the 10% of the standard deviation of the distances as suggested in [Luxburg, 2007].

For each of the method we computed the regularized graph Laplacian, where we added $\gamma_g = 10^{-6}$ to the diagonal. For the diagonal matrix F of empirical weights we set $f_l = 10$ for the labeled and $f_u = 0.1$ for the unlabeled examples. We set parameter γ_q in our method to 10^5 . Finally, we varied the number of cluster centers as $k = 15, 20, 25, 30, 60$ and 90 .

5.3.2.2 Results The result are shown in Figure 18 for varying number of labeled examples m and centroids k . Error bars show the 95% confidence intervals over 50 runs. The 5 compared methods are 1) subsample - random subsampling 2) k -means as a preprocessing 3) our method: elastic-joint 4) elastic net as a preprocessing 5) full soft – harmonic solution using all unlabeled examples to create the full graph. For the Car dataset and $m = 2$ unlabeled examples, our method outperformed other baselines for different number of cluster centers up-to $k = 60$ when all the methods achieved the performance of the full non-approximated

graph. For $m = 10, 20$ and 50 , all the subsampling methods are comparable. For the COIL dataset, $m = 2$ of labeled examples was not sufficient for the learning as the classes are perfectly balanced and all the methods produced a trivial classifier comparable to a random one, including the SSL on the full graph with all the examples. For $m = 10, 20$ and 50 our method significantly outperformed all the other approximation methods. The result for SecStr (Figure 7) is similar for all the baselines. We utilize this data set to show the time complexity of different methods. Notice that the same observation and setup has been used in [Chapelle et al., 2006].

5.3.3 Online Quantized SSL Experiments

The experimental section is divided into two parts. In the first part, we evaluate our online learner (Figure 3) on UCI ML repository datasets (Section 5.1.2). In the second part, we apply our learner to solve two face recognition problems. In all experiments, the multiplicative parameter m of the k -centers algorithm is set to 1.5.

5.3.3.1 UCI ML Repository Datasets In the first experiment, we study the online quantization error $\|L_t^q - L_t^o\|_F$ and its relation to the HS on the quantized graphs W_t^q . This experiment is performed on two datasets from the UCI ML repository: letter and optical digit recognition. The datasets are converted into a set of binary problems, where each class is discriminated against every other class. The similarity weights computed as $w_{ij} = \exp[-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / (2p\sigma^2)]$, where p is the number of features and σ denotes the mean of their standard deviations. Our results are averaged over 10 problems from each dataset, and shown in Figures 19 and 20.

In Figure 19, we fix the number of centroids at $k = 200$ and study how the quality of our solution changes with the learning time t . The upper plots show the difference between the normalized Laplacian L_t^o and its approximation L_t^q at time t . The bottom plots show the cumulative accuracy of the harmonic solutions on W (light gray lines), W_t^o (dark gray lines), and W_t^q (black lines) for various times t . Two trends are apparent. First, as time t increases, the error $\|L_t^q - L_t^o\|_F$ slowly levels off. Second, the accuracy of the harmonic

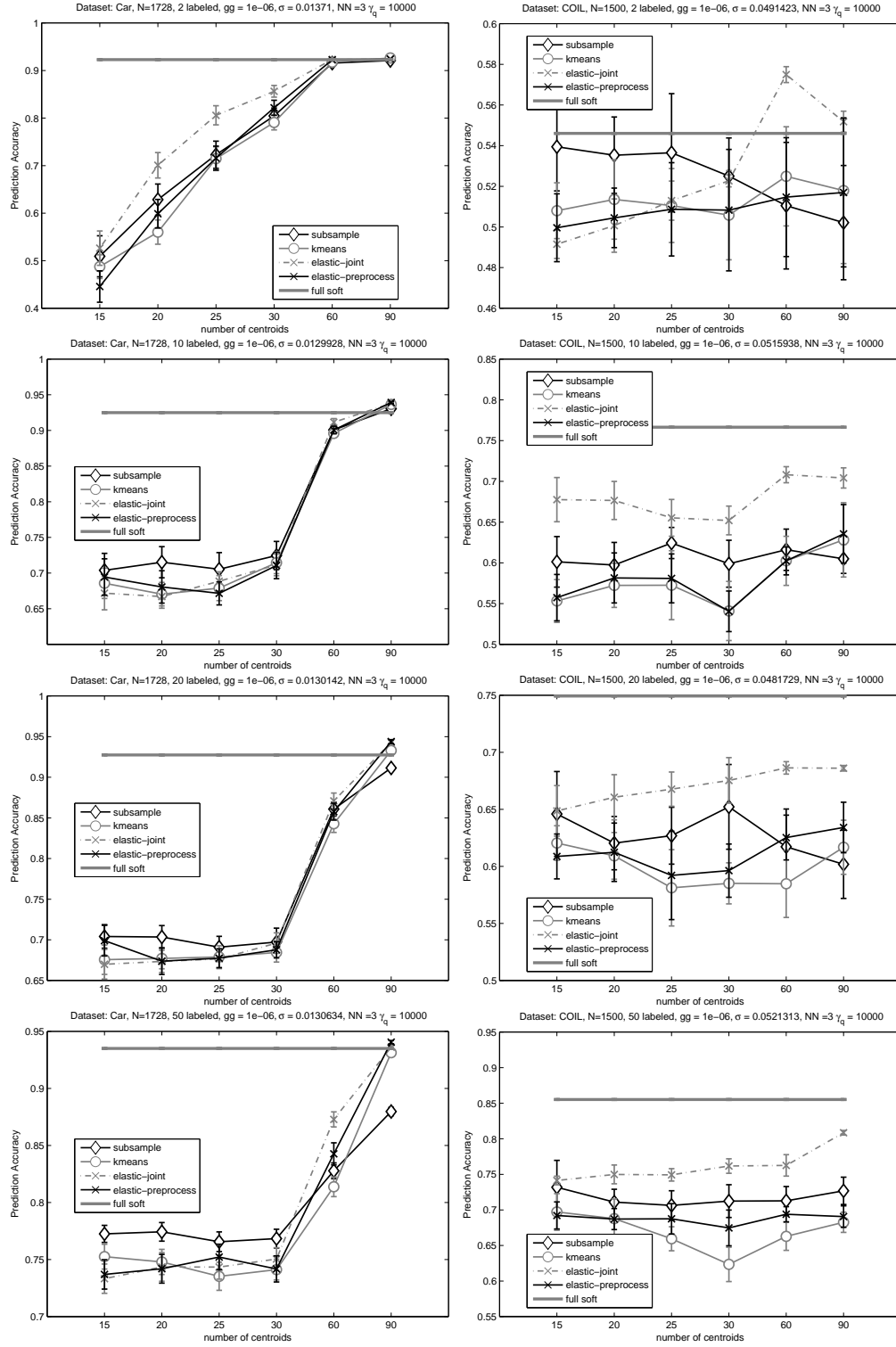


Figure 18: Coil and Car datasets from UCI ML Repository

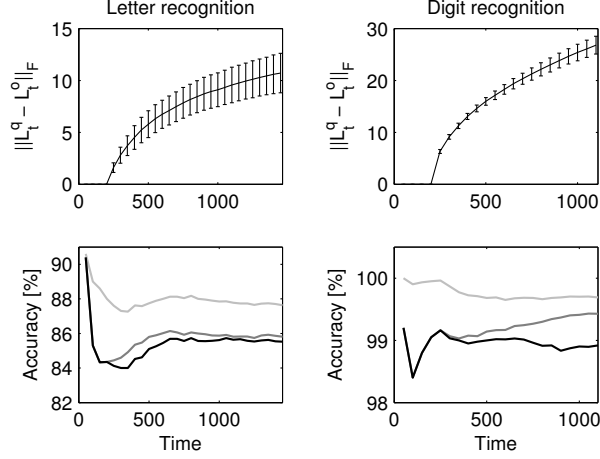


Figure 19: UCI ML: Quality of approximation as a function of time

solutions on W_t^q changes little with t . These trends indicate that a fixed number of centroids k may be sufficient for quantizing similarity graphs that grow with time. In Figure 20, we fix the learning time at $t = n$ and vary the number of centroids k . The upper plots show the difference between the normalized Laplacian L and its approximation L_n^q . The difference is plotted as a function of the number of centroids k . The bottom plots compare the cumulative accuracy of the harmonic solutions up to time n on W (light gray lines), W_t^o (dark gray lines), and W_t^q (black lines). Note that as k increases, the quantization error decreases and the quality of the solutions on W_t^q improves. This trend is consistent with the theoretical results in our work.

5.3.3.2 Face Recognition In the second experiment, we evaluate our learner on 2 face recognition datasets: office space and environment adaptation. (Section 5.1.4).

The similarity of faces \mathbf{x}_i and \mathbf{x}_j is computed as $w_{ij} = \exp[-d(\mathbf{x}_i, \mathbf{x}_j)^2/2\sigma^2]$, where σ is a heat parameter, which is set to $\sigma = 0.025$, and $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance of the faces in the feature space. To make the graph W sparse, we treat it as an ε -neighborhood graph and set w_{ij} to 0 when $w_{ij} < \varepsilon$. The scalar ε is set as $\varepsilon = 0.1\gamma_g$. As a result, the lower the regularization parameter γ_g , the higher the number of edges in the graph W and our learner

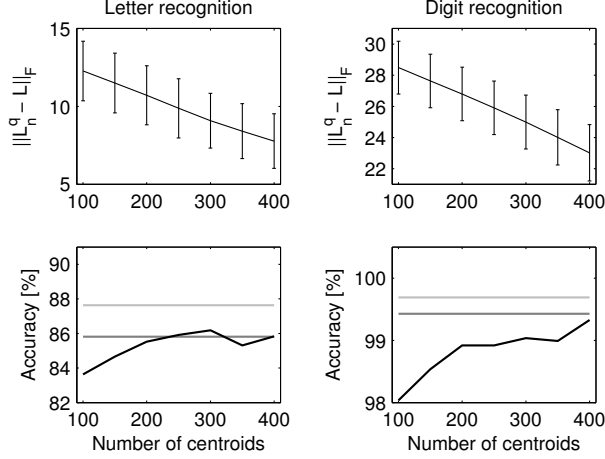


Figure 20: UCI ML: Quality of approximation as a function of number of centroids

extrapolates to more unlabeled examples. If an example is disconnected from the rest of the graph W , we treat it as an outlier, and neither predict the label of the example nor use it to update the quantized graph. This setup makes our algorithm robust to outliers, and allows for controlling its precision and recall by a single parameter γ_g . In the rest of the section, the number of centroids k is fixed at 500. More details are provided in Section 5.3.1.2.

In Figure 21, we compare our online algorithm to online semi-supervised boosting [Grabner et al., 2008] and a nearest-neighbor (NN) classifier, which is trained on all labeled faces. The recognizers are trained by a NN classifier (gray lines with circles), online semi-supervised boosting (thin gray lines), and our online learner (black lines with diamonds). The plots are generated by varying the parameters ε and γ_g . From left to right, the points on the plots correspond to decreasing values of the parameters. Online semi-supervised boosting is performed on 500 weak NN learners, which are sampled at random from the whole environment adaptation dataset (solid line), and its first and last quarters (dashed line). The algorithm of [Grabner et al., 2008] is modified to allow for a fair comparison to our method. First, all weak learners have the nearest-neighbor form $h_i(\mathbf{x}_t) = \mathbb{1}\{w_{it} \geq \varepsilon\}$, where ε is the radius of the neighborhood. Second, outliers are modeled implicitly. The new algorithm learns a regressor $H(\mathbf{x}_t) = \sum_i \alpha_i h_i(\mathbf{x}_t)$, which yields $H(\mathbf{x}_t) = 0$ for outliers and $H(\mathbf{x}_t) > 0$

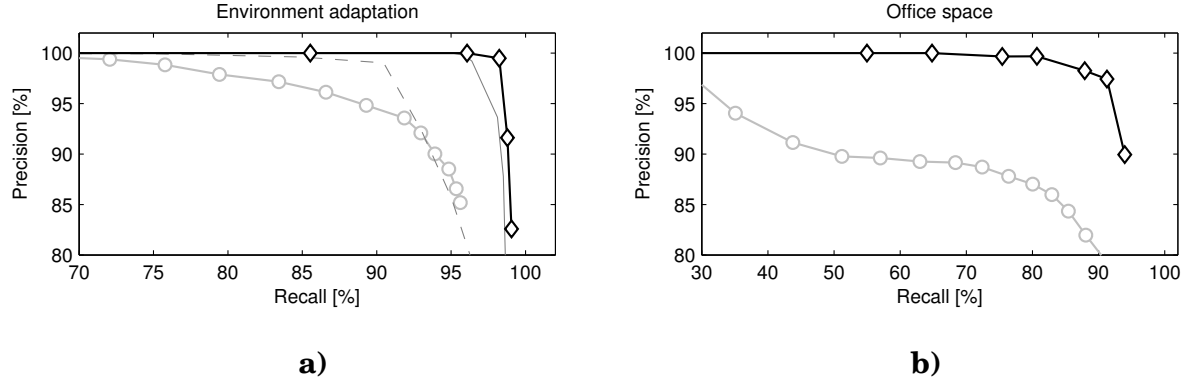


Figure 21: Comparison of 3 face recognizers on 2 face recognition datasets

when the detected face is recognized.

Figure 21a clearly shows that our learner is better than the nearest-neighbor classifier. Furthermore, note that online semi-supervised boosting yields as good results as our method when given a good set of weak learners. However, future data are rarely known in advance, and when the weak learners are chosen using only a part of the dataset, the quality of the boosted results degrades significantly (Figure 21a). In comparison, our algorithm constantly adapts its representation of the world. How to incorporate a similar adaptation step in online semi-supervised boosting is not obvious.

In Figure 21b, we evaluate our learner on an 8-class face recognition problem. Despite the fact that only 4 faces of each person are labeled, we can identify people with 95 percent precision and 90 percent recall. In general, our precision is 10 percent higher than the precision of the NN classifier at the same recall level.

5.3.4 Parallel SSL

In this experiment, we demonstrate how to speed up the online HS on a graph using an additional structure and parallelization (Section 3.4). Therefore, we perform our experiments on an Intel Xeon workstation with six cores. The experimental setup is the same as in Section 5.3.3. The number of labeled examples used for training models of Person 1 and 13

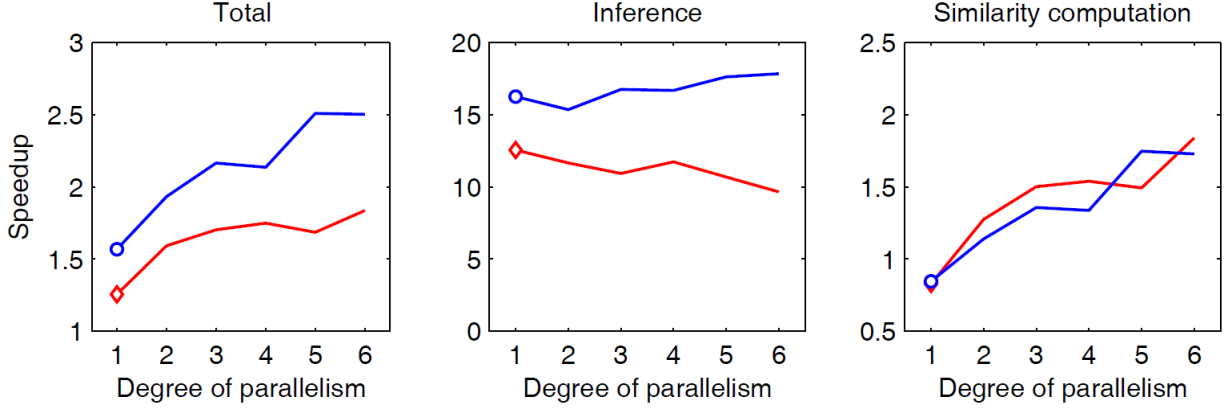


Figure 22: Speedups in the total, inference, and similarity computation times

(from Figure 15) is 5 and 6, respectively. Figure 22 reports speedups due to decomposing the online HFS on 300 vertices into n_l smaller graphs of 50 vertices. The plots correspond to Person 1 (red lines) and 13 (blue lines) in our dataset. The diamonds and circles mark speedups that are obtained by the decomposition alone. We observe two main trends. First, the decomposition alone yields a modest speedup of 35% on average. The speedup is due to 15 times faster inference, which is a result of solving n_l smaller systems of linear equations, each with 50 variables, instead of a bigger one with 300. Second, we parallelize the online HS on the n_l smaller graphs using OpenMP [OpenMP, 2008]. The problem is trivially parallelizable because the graphs can be updated independently. Figure 22 shows that as the number of used cores increases, the online HFS can be sped up more than two times on average. The speedup is due to parallelizing the computation of similarities w_{ij} , which at this point consumes much more time than inference. Finally, note that the proposed decomposition has almost no impact on the quality of our solutions. For Person 1 and Person 13, the loss in accuracy is 2.5% and 1%, respectively.

5.3.5 Conclusions

In this section, we evaluated our algorithms for the semi-supervised learning tasks. Max-margin graph cuts algorithm learns max-margin graph cuts that are conditioned on the labels induced by the harmonic function solution. The approach is evaluated on a synthetic problem and three UCI ML repository datasets, and we showed that it usually outperforms manifold regularization of SVMs. Next, we evaluated our joint optimization approach for graph quantization and label propagation. We experimentally showed that this approach can lead to a significant gain in classification accuracy over the competing quantization approaches. In the online SSL experiments we approximated similarity graph for a harmonic solution. This algorithm significantly reduces the expense of the matrix computation in the harmonic solution, while retaining good control on the classification accuracy. Our evaluation shows that a significant speedup for semi-supervised learning can be achieved with little degradation in classification accuracy. We further approximated the computation by decomposing the graph into several smaller graphs, thereby performing parallel multi-manifold learning. With such a decomposition we were able to speed up the computation even more with almost no loss in accuracy.

5.4 EVALUATIONS OF CAD

In this section we present the experiments using the CAD methods from Chapter 3. For each method we first discuss the datasets from Section 5.1 used for comparison.

In all our experiments, we focus on the conditional anomalies in the class labels with respect to the features. In general in the whole field of anomaly detection and in medical domain especially, the evaluation is extremely challenging. Most of time, it is subjective. Most veracious evaluation would be to have human experts judging the goodness of the methods. Since this is a very expensive way, most researchers resort to some surrogate measures. In the area of mislabel detection, the most common surrogate measure is to change the labels of a fraction of the dataset and observe how many of those were detected

as mislabeled. The problem with this measure is that the anomalies in the real life datasets are really *sampled* randomly. However, if the model performs well in prediction we may assume that it models the data well and so it will be able to detect anomalies better. We will provide two kinds of evaluations, the evaluation when the ground truth is known or can be computed (Section 5.4.1) and then the evaluation with human experts (Section 5.4.2).

5.4.1 Evaluation of CAD with Known Ground Truth

5.4.1.1 CAD on Synthetic Datasets with Known Distribution The evaluation of a CAD is a very challenging task when the true model is not known. Therefore, we first evaluate and compare the results of different CAD methods on three synthetic datasets (D1, D2, and D3) with known underlying distributions that let us compute the true anomaly scores (Section 5.1.3.2). Then, we show the advantage of regularizing discriminative approach on a synthetic dataset. We will use the a 2D synthetic dataset, when we can demonstrate the ability to tackle fringe and isolated points outliers as described in Section 3.1.6.

For each experiment we sample the datasets 10 times. After the sampling, we randomly switch the class labels for three percent of examples. We then calculate the true anomaly score as $P(y \neq y_i | \mathbf{x}_i)$ reflecting how anomalous is the label of the example with respect to the true model.

Each of the methods outputs a score which orders the examples according to the belief of the anomalous labeling. For each of the CAD methods, we assessed how much this ordering is consistent with the ordering of the true anomaly score. In particular, we calculated the area under the receiver operating characteristic (AUROC), which is inversely proportional to the number of swaps between the ordering induced by the evaluated method and the true ordering.

Table 1 compares the AUROCs of the experiment for all methods for 1000 samples per dataset. The results demonstrate that our λ -RWCAD method outperforms weighted k -NN, 1-class SVM and discriminative SVM with RBF kernel³ and it is comparable to our label propagation SoftHAD algorithm on D2 and D3. SoftHAD seems to be the best choice overall

³We also evaluated the linear versions of SVM and 1-class SVM, but the results were inferior to the ones with the RBF kernel.

	Dataset D1	Dataset D2	Dataset D3
<i>SVM RBF</i>	58.4% (7.4)	49.3% (2.1)	51.7% (1.9)
<i>1cSVM RBF</i>	51.5% (0.8)	47.4% (0.6)	59.1% (0.6)
<i>SoftHAD</i>	82.8% (1.3)	63.9% (2.3)	63.5% (3.3)
<i>wk-NN</i>	64.3% (2.2)	45.6% (1.6)	62.5% (1.5)
<i>λ-RWCAD</i>	64.7% (0.8)	68.9% (1.1)	67.4% (1.9)

Table 1: Mean anomaly AUROC and variance on three synthetic datasets

because it takes advantage of both local and global consistency. However, is computationally more expensive.

In the next experiment we evaluated the scalability of the graph-based methods as we increase the number of examples. All of the graph methods were given the same graph (with the same weight matrix). Figure 23 compares the running times of these algorithm. We see that while the running time of the SoftHAD algorithm becomes prohibitive once the number of examples gets into thousands, our algorithm scales similarly to k -NN method. Figure 23 also shows the time spent in constructing the graph from the data, which is the same among all the graph-based methods. Observe that both weighted k -NN and our λ -RWCAD algorithm take very little time over the necessary graph construction time to do their calculations.

5.4.1.2 CAD on UCI ML Datasets with ordinal response variable We also evaluated our method on the three UCI ML datasets [Asuncion and Newman, 2011] for which an ordinal response variable was available to calculate the true anomaly score. In particular, we selected 1) *Wine Quality* dataset with the response variable *quality* 2) *Housing* dataset with the response variable *median value of owner-occupied homes* and 3) *Auto MPG* dataset the response variable *miles per gallon*. In each of the dataset we scaled the response variable y_r to the $[-1, +1]$ interval and set the class label as $y := y_r \geq 0$. As with the syn-

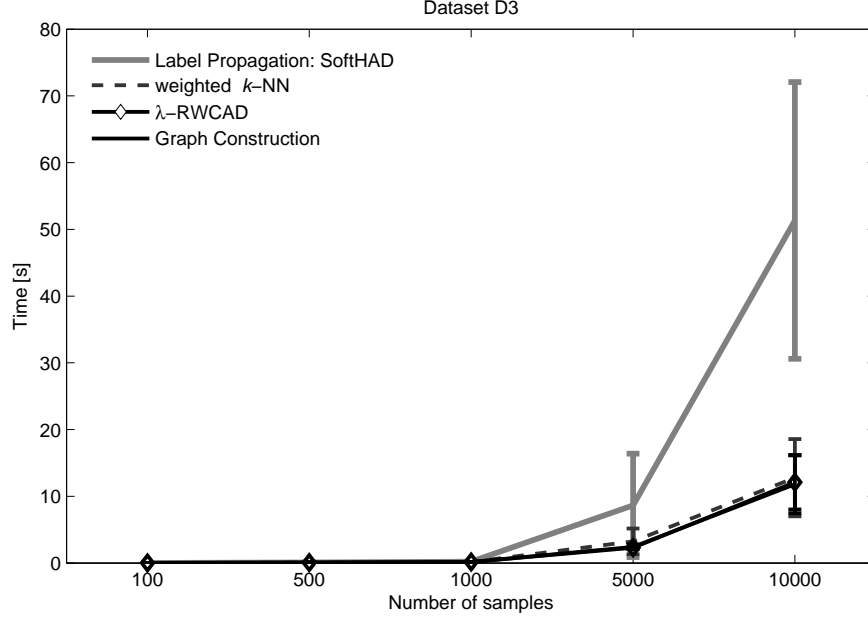


Figure 23: Computation time comparison for the three graph-based methods

thetic datasets, we randomly switched the class labels for three percent⁴ of examples. The true anomaly score was computed as the absolute difference between the original response variable y_r and the (possibly switched) label. Table 2 compares the agreement scores (over 100 runs) to the true score for all methods on (2/3, 1/3) train-test split. Again, we see that SoftHAD either performed the best or was close to the best method.

5.4.1.3 CAD on Core Dataset with Fringe Points In this part we tested our CAD method on the synthetic Core dataset (Section 5.1.3.1). Besides one-class SVM (Section 5.2.2), we also compared to the weighted NN described in Section 4.1.1 and to the Cross-Outlier method [Papadimitriou and Faloutsos, 2003] described in Section 2.1.2.

In Figure 24, the training data consists of a bigger square of 100 uniformly distributed points (blue ‘x’), a smaller square of 50 uniformly distributed points (red ‘+’) and 2 small groups of points (3 points from each class). Testing dataset is twice as big sampled from the same distribution. Big black dots display true conditional anomalies and top 12 highest

⁴We also performed the experiments with 1% to 10% of switched labels with the same trends.

	Wine Quality	Housing	Auto MPG
<i>QDA</i>	75.1% (1.3)	56.7% (1.5)	65.9% (2.9)
<i>SVM</i>	75.0% (9.3)	58.5% (4.4)	37.1% (8.6)
<i>1-class SVM</i>	44.2% (1.9)	27.2% (0.5)	50.1% (3.5)
<i>wk-NN</i>	67.6% (1.4)	44.4% (2.0)	61.4% (2.3)
<i>SoftHAD</i>	74.5% (1.5)	71.3% (3.2)	72.6% (1.7)

Table 2: Mean anomaly agreement score and variance on 3 UCI ML datasets

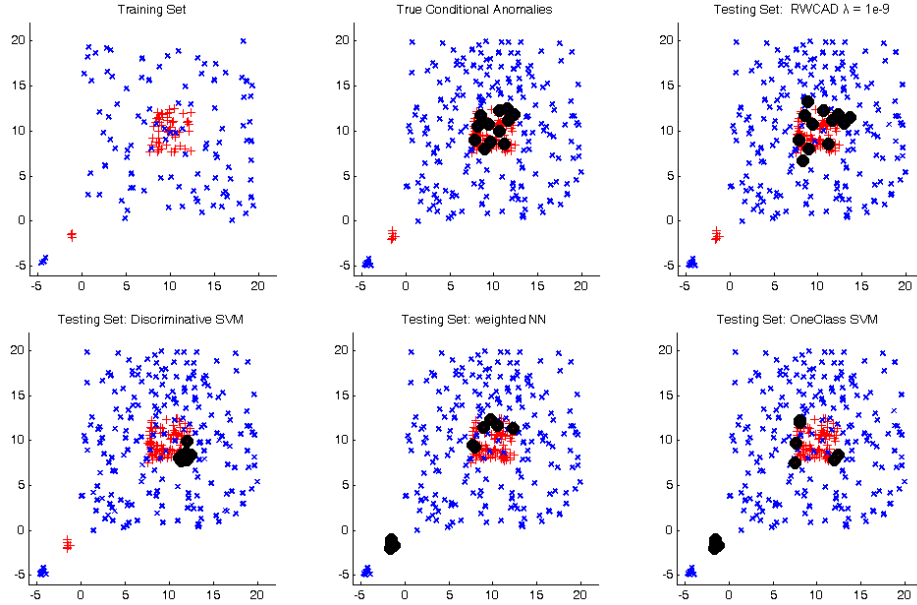


Figure 24: Conditional anomaly detection on a synthetic *Core* dataset

ranked conditional anomalies for 1) our λ -RWCAD method 2) discriminative SVM anomaly detection 3) weighted nearest neighbor and 4) one-class SVM learned for both of the classes.

The cross-outlier method [Papadimitriou and Faloutsos, 2003] was able to find all of the conditional anomalies in the middle square but also declared many *fringe* points (points at the outer boundary of the bigger square) as anomalous (see Figure 2, middle row in [Papadimitriou and Faloutsos, 2003]). Although the authors claim that the fringe points are “clearly different from the rest of the points” [Papadimitriou and Faloutsos, 2003], we prefer methods that only find anomalously labeled instances. In Figure 24, we show top 12 highest scored anomalies from 4 methods 1) our λ -RWCAD method 2) linear discriminative SVM anomaly detection (described in Section 5.2.1) 3) weighted nearest neighbor and 4) one-class SVM with RBF kernel learned for both of the classes.

Discriminative SVM anomaly detection (Figure 24, bottom left) could only detect the fringe points from the smaller square since the anomaly score there corresponds to the most wrongly classified testing points. Next, the objective of one-class SVM is to detect the points with the minimal support. In Figure 24, bottom right, we see that this way one-class SVM ranked with the highest score the fringe points of the smaller square and one of the tiny squares. Weighted NN (Figure 24, bottom middle) detects half of the true anomalies, but also falsely detects one of the tiny squares as anomalous. Our method (Figure 24, top right) avoids such a mistake due to the regularization. Although, the results of our method do not completely match with the truth, the 3 points detected outside the smaller square are in its vicinity.

5.4.1.4 Conclusions We showed how we use regularization to avoid detection of unconditional anomalies and the fringe points at the boundary of an distribution support. In general, the advantage of the CAD approach over knowledge-based error detection approaches is that the method is evidence-based, and hence requires no or minimum input from the domain expert.

5.4.2 Evaluation of Expert Assesed Clinically Useful Anomalies

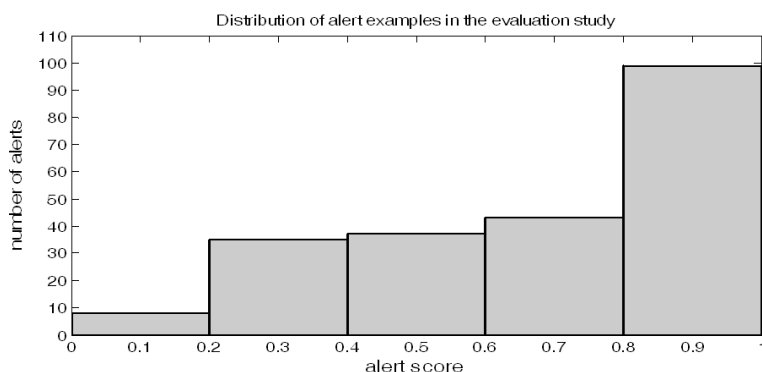


Figure 25: Histogram of alert examples in the study according to their alert score

5.4.2.1 Pilot study in 2009 The aim of the study [Hauskrecht et al., 2010] was to test the hypothesis that the clinical anomalies lead to good clinical alerts.

Learning anomaly detection models The training set was used to build three types of anomaly detection models: (1) models for detecting unexpected lab-order omissions, (2) models for detecting unexpected medication omissions, (3) models for detecting unexpected continuation of medications (commissions).

Selection of alerts for the study The alerts for the evaluation study were selected as follows. We first applied all the above anomaly detection models to matching patient instances in the test set. The following criteria were then applied. First, only models with AUC of 0.68 or higher were considered. This means that many predictive models built did not qualify and were never used. Second, the minimum anomaly score for all alert candidates had to be at least 0.15. Third, for each decision, only top 125 anomalies and top 20 alerts obtained from the test data were considered as alert candidates. This led to 3,768 alert candidates, from which we selected 222 alerts for 100 patients, such that 101 alerts were lab-omission alerts, 55 were medication-omission alerts, and 66 were medication-commission alerts. Figure 25 shows the distribution of alerts in the study according to the alert score.

Alert reviews. The alerts selected for the study were assessed by physicians with expertise in post-cardiac surgical care. The reviewers (1) were given the patient cases and model-

generated alerts for some of the patient management decisions, and (2) were asked to assess the clinical usefulness of these alerts. We recruited 15 physicians to participate in the study, of which 12 were fellows and 3 were faculty from the Departments of Critical Care Medicine or Surgery. The reviewers were divided randomly into five groups, with three reviewers per group, for the total of 15 reviewers. Overall, each clinician made assessments of 44 or 45 alerts generated on 20 different patients. The total number of alerts reviewed by all clinicians was 222 and included: 101 lab omission alerts, 55 medication omission alerts, and 66 medication commission alerts. The survey was conducted over the Internet using a secure web-based interface [[Post and Harrison, 2008](#)].

Alert assessments. The pairwise kappa agreement scores for the groups of three ranged from 0.32 to 0.56. We used the majority rule to define the gold standard. That is, an alert was considered to be useful if at least two out of three reviewers found it to be useful. Out of 222 alerts selected for the evaluation study, 121 alerts were agreed upon by the panel (via the majority rules) as an useful alert.

Analysis of clinical usefulness of alerts. We analyzed the extent to which the alert score from a model was predictive of it producing clinically important alerts. Figure [26](#) summarizes the results by binning the alert scores (in intervals of the width of 0.2) and presenting the true alert rate per bin. The true alert rates vary from 19% for the low alert scores to 72% for high alert scores, indicating that higher alert scores are indicative of higher true alert rates. This is also confirmed by a positive slope of the line in Figure [26](#) which is obtained by fitting the results via linear regression, and the results of the ROC analysis. All alerts reviewed were ordered according to their alert scores, from which we generated an ROC curve. The AUC for our alert score was 0.64. This is statistically significantly different from 0.5, which is the value one expects to see for random or non-informative orderings. Again, this supports that higher alert scores induce better true alert rates. Finally, we would like to note that alert rates in Figure 4 are promising and despite alert selection restrictions, they compare favorably to alert rates of existing clinical alert systems [[Schedlbauer et al., 2009](#), [Bates et al., 2003](#)].

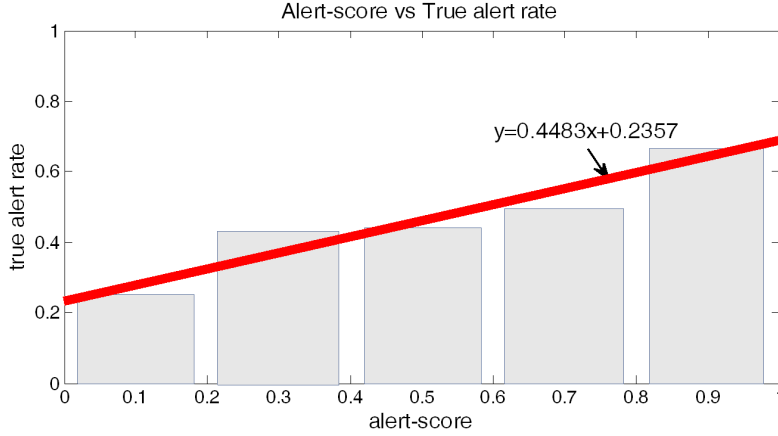


Figure 26: The relationship between the alert score and the true alert rate

5.4.2.2 Soft Harmonic Anomaly Detection For our experiment, we used PCP dataset (Section 5.1.1) and reused the human expert evaluations from Section 5.4.2.1. We computed the anomaly scores according to (Section 3.1.6.2)

Scaling for multi-task anomaly detection So far, we have described CAD only for a single task (anomaly in the single label). In this dataset, we have 749 binary tasks (or labels) that correspond to 749 different possible orders of lab tests or medications. In our experiments, we compute the CAD score for each task independently. Figure 27 shows the CAD scores for two of them. CAD scores close to 1 indicate that the order should be done, while the scores close to 0 indicate the opposite. The ranges for the anomaly scores can vary among the different labels/tasks as one can notice in Figure 27. The scores for the top and bottom task range from 0.1 to 0.9 and from 0.25 and 0.61 respectively. The arrow in both cases points to the scores of the evaluated examples, both with negative labels. Despite the score is lower for the bottom task, we may believe that it is more anomalous because it is more extreme within the scores for the same task. However, we want to output an anomaly score which is comparable among the different tasks/labels so we can, for example, set a unified threshold when the system is deployed in practice. Another reason for comparable scores is that we can have, for instance, 2 models each alerting on the fact that a certain

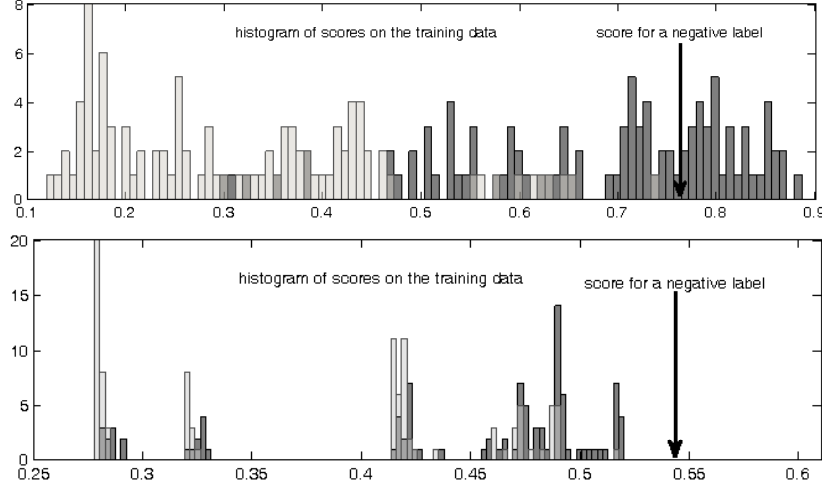


Figure 27: Histogram of anomaly scores for 2 different tasks

medication was omitted. Nevertheless, omitting one of medications can be more severe than the other (think Antibiotics vs. Vitamins). To achieve this score comparability, we propose a simple approach where we take the minimum and the maximum score obtained for the training set and scale all scores for the same task linearly so that the score after the scaling ranges from 0 to 1.

In Figure 28, we fixed $\gamma_g = 1$ and varied the number of examples we sampled from the training set to construct the similarity graph and compared it to the weighted k -NN. The error bars show the variances over 10 runs. Notice that the both of the methods are not too sensitive to the graph size. This is due to the multiplicity adjustment for the backbone graph (Section 3.1.6.2). Since we used the same graph both for SoftHAD and weighted k -NN, we anticipate that we are able to outperform weighted k -NN due to the label propagation over the data manifold and not only within the immediate neighborhood. In Figure 29, we compared SoftHAD to CAD using SVM with RBF kernel for different regularization settings. We sampled 200 examples to construct G (or train an SVM) and varied the γ_g regularizer (or cost c for SVM). We outperformed SVM approach over the range of regularizers. AUC for 1-class SVM with RBF was consistently below 55% and we do not show it in the figure.

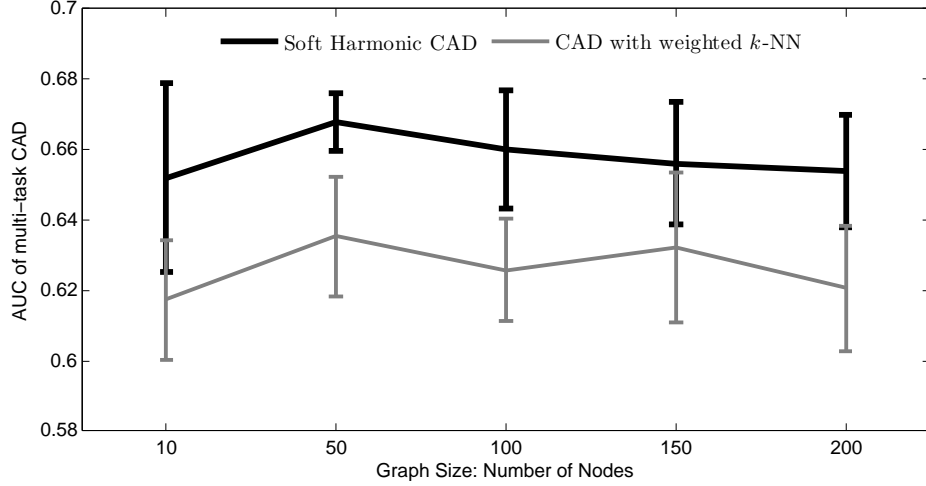


Figure 28: Medical Dataset: Varying graph size

We also compared the two methods with scaling adjustment for this multi-task problem (Figure 29). Scaling anomaly scores improved the performance of both methods and makes the methods less sensitive to the regularization settings.

5.4.2.3 Conclusions In the evaluations with human experts on the real-world data we showed we can indeed learn clinically useful alerts. The results reported here support that this is a promising methodology for raising clinically useful alerts. Moreover, we showed that with label propagation on the data similarity graph build from patient records, we can significantly outperform previously proposed SVM based anomaly detection in detecting conditional anomalies.

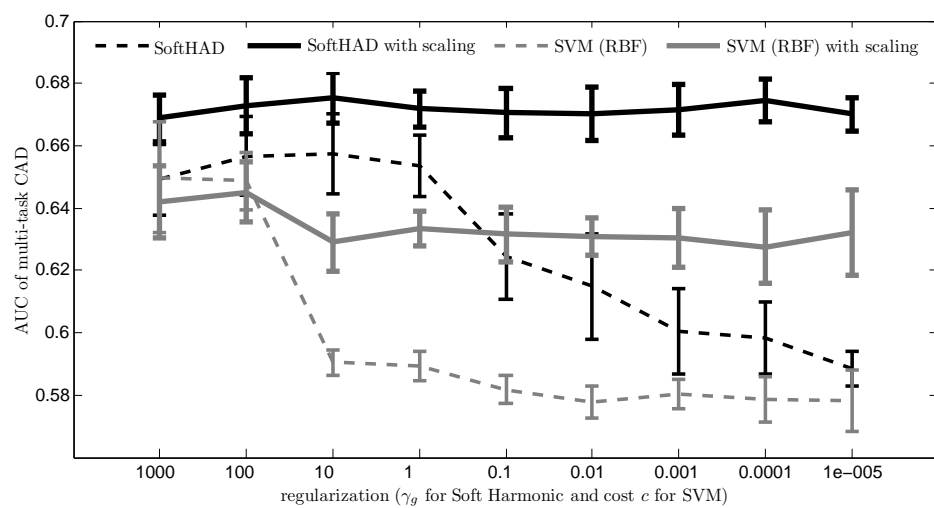


Figure 29: Medical Dataset: Varying regularization

6.0 DISCUSSIONS

We have presented several algorithms for semi-supervised learning and conditional anomaly detection. The algorithms are based on the label propagation on the similarity graph built from the examples in the dataset. Label propagation on graphs is a polynomial, but still computationally expensive method. Therefore, we focused on the approximation approaches for the cases with large datasets and when the data arrive in a stream. The main contributions of this thesis to the field of machine learning are summarized below.

- We presented one of the first works on *online semi-supervised learning*. Despite very natural scenario, this setting has not been extensively studied in the past. To our best knowledge this is the first work on online semi-supervised learning that comes with the theoretical guarantees. Moreover, we built a real-time system that works on noisy real-world data.
- We introduced a label propagation method for conditional anomaly detection and applied it to compute the anomaly score for the class labels. We presented a general framework where the discriminative models need to be regularized to decrease the effect caused by isolated and fringe points in the data.
- We presented a new semi-supervised learning algorithm based on max-margin graph cuts, which in some classes of learning functions can perform better than manifold regularization approach.
- We introduced a joint learning of the backbone graph and the label propagation and show its relationship to the elastic nets. This is one of the first works (besides [Zhu and Lafferty, 2005]) that relates propagated labels and the cluster centers.

We also made contributions to the area of the health informatics:

- The most important finding is that the existing rule-based alert systems currently used in hospitals, that require expensive human time to tune can be learned from the past patient data instead.
- We proposed a non-parametric method that can discover anomalies in the clinical actions. The common uses cases are: 1) discovery of the omitted order of a lab test 2) commission of a drug that has interactions with previously taken drugs 3) controlling overspending: execution of an expensive action that was not necessary and the resources could be used better.
- We conducted an extensive study with the human evaluation of the alerts on the real patient records showing that the higher anomaly score corresponded to the higher severity of the alerts.

There are, however, some assumptions and limitation of our methods:

- We assume that the data can be modeled with the pair-wise similarities between the nodes and that such a model is meaningful.
- The similarity function between the graph nodes needs to be given or learned.
- Our methods are expected to perform well when the manifold assumption holds.
- In the approximation settings, when we create a summary graph (both in online and a large scale setting), we assume that we can model the data with the reduced number of nodes well.

We now outline some related open questions and research opportunities.

- *Structured Anomaly Detection*

In this thesis we applied our conditional anomaly detection method to discover unusual clinical actions. However, we did it separately for each action. Nevertheless, these actions are not independent. For example, a clinician usually prescribe a set of drug such that:

- drugs with the same effect do not tend to be given at the same time.
- drugs with the opposite effect do not tend to be given at the same time.
- drugs with negative interactions do not tend to be given at the same time.

Therefore, we can form groups of drugs from which at most one is administered at the same time. This additional information could be given a priori or learned from the data.

- *Graph Parametrization:* Despite the research in this area, the graph constructions is still not well understood. There are some rules of thumb, such as $\log(n)$ for number of neighbors, but a problem-specific calibration is usually needed. In particular, the clinical data, could benefit from the similarity measures (kernels) that would measure the similarity of the conditions from the electronic health data.
- *Multi-manifold Learning* In our multi-manifold learning approach, we decomposed the graph and kept updating each of the components independently in parallel. There can be some benefit in accuracy if we allow the components to exchange some information.
- *Concept Drift* In this thesis we were concerned with adapting to the distribution in a short-term. The problem of concept drift is concerned with long-term changes, such as when a face of a person changes as they grow older or when the medical practices change. One possible extension of our methods can be the online graph-based learning with forgetting the history. For example, we can delete the nodes from the graph which were added long time ago and do not change the current prediction much if they are removed.

BIBLIOGRAPHY

- [Aggarwal et al., 2003] Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases - Volume 29*, VLDB '2003, pages 81–92. VLDB Endowment.
- [Aggarwal and Yu, 2001] Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 37–46, New York, NY, USA. ACM.
- [Akoglu et al., 2010] Akoglu, L., McGlohon, M., and Faloutsos, C. (2010). Oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II*, pages 410–421.
- [Aktolga et al., 2010] Aktolga, E., Ros, I., and Assogba, Y. (2010). Detecting outlier sections in us congressional legislation. In *Proceedings of SIGIR*.
- [Asuncion and Newman, 2011] Asuncion, A. and Newman, D. (2011). UCI machine learning repository.
- [Bates et al., 2003] Bates, D. W., Kuperman, G. J., Wang, S., Gandhi, T., Kittler, A., Volk, L., Spurr, C., Khorasani, R., Tanasijevic, M., and Middleton, B. (2003). Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inform Assoc*, 10(6):523–530.
- [Belkin et al., 2004] Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *Proceeding of the 17th Annual Conference on Learning Theory*, pages 624–638.
- [Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhvani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434.
- [Bennett and Demiriz, 1999] Bennett, K. and Demiriz, A. (1999). Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11*, pages 368–374.

- [Bezdek and Hathaway, 2002] Bezdek, J. C. and Hathaway, R. J. (2002). Some notes on alternating optimization. In *Proceedings of the 2002 AFSS International Conference on Fuzzy Systems. Calcutta: Advances in Soft Computing*, AFSS '02, pages 288–300, London, UK. Springer-Verlag.
- [Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2:499–526.
- [Bradski, 2000] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [Breunig et al., 2000] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104.
- [Brodley and Friedl, 1999] Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *J. Artif. Intell. Res. (JAIR)*, 11:131–167.
- [Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.
- [Chandola et al., 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58.
- [Chang and Lin, 2001] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [Chapelle et al., 2006] Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.
- [Charikar et al., 1997] Charikar, M., Chekuri, C., Feder, T., and Motwani, R. (1997). Incremental clustering and dynamic information retrieval. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 626–635.
- [Chawla et al., 2003] Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. In *PKDD*, pages 107–119.
- [Chung, 1997] Chung, F. (1997). *Spectral Graph Theory*. American Mathematical Society.
- [Cortes et al., 2008] Cortes, C., Mohri, M., Pechyony, D., and Rastogi, A. (2008). Stability of transductive regression algorithms. In *Proceedings of the 25th International Conference on Machine Learning*, pages 176–183.
- [Cramér, 1999] Cramér, H. (1999). *Mathematical methods of statistics*. Princeton landmarks in mathematics and physics. Princeton University Press.
- [Das, 2009] Das, K. (2009). *Detecting Patterns of Anomalies*. PhD thesis, Carnegie Mellon University.

- [Das et al., 2008] Das, K., Schneider, J., and Neill, D. B. (2008). Anomaly pattern detection in categorical datasets. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 169–176, New York, NY, USA. ACM.
- [Delalleau et al., 2005] Delalleau, O., Bengio, Y., and Roux, N. L. (2005). Efficient non-parametric function induction in semi-supervised learning. In *AISTAT*, pages 96–103.
- [Drineas and Mahoney, 2005] Drineas, P. and Mahoney, M. W. (2005). On the Nyström method for approximating a Gram matrix for improved kernel-based learning. In *Proceedings of COLT, 2005*.
- [Eskin, 2000] Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th International Conf. on Machine Learning*, pages 255–262. Morgan Kaufmann, San Francisco, CA.
- [Fergus et al., 2009] Fergus, R., Weiss, Y., and Torralba, A. (2009). Semi-supervised learning in gigantic image collections. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 522–530. NIPS Foundation (<http://books.nips.cc>).
- [Fine and Scheinberg, 2001] Fine, S. and Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264.
- [Fowlkes et al., 2004] Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the Nyström method. *IEEE Transactions on PAMI*, 26(2).
- [Gans et al., 2005] Gans, D., Kralewski, J., Hammons, T., and Dowd, B. (2005). Medical groups’ adoption of electronic health records and information systems. *Health Aff (Millwood)*, 24(5):1323–1333.
- [Goldberg et al., 2008] Goldberg, A., Li, M., and Zhu, X. (2008). Online manifold regularization: A new learning setting and empirical study. In *Proceeding of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.
- [Goldberg et al., 2009] Goldberg, A. B., Zhu, X., Singh, A., Xu, Z., and Nowak, R. (2009). Multi-manifold semi-supervised learning. *Journal of Machine Learning Research*, 5:169–176.
- [Goldberger et al., 2004] Goldberger, J., Roweis, S. T., Hinton, G. E., and Salakhutdinov, R. (2004). Neighbourhood components analysis. In *NIPS*.
- [Gorban and Zinovyev, 2009] Gorban, A. and Zinovyev, A. (2009). Principal graphs and manifolds. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*, pages 28 – 59. Information Science Reference.

- [Grabner et al., 2008] Grabner, H., Leistner, C., and Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision*, pages 234–247.
- [Gray and Neuhoff, 1998] Gray, R. and Neuhoff, D. (1998). Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383.
- [Hanley and Mcneil, 1982] Hanley, J. A. and Mcneil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.
- [Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer.
- [Hauskrecht et al., 2010] Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaram, S., and Cooper, G. (2010). Conditional outlier detection for clinical alerting. *Annual American Medical Informatics Association Symposium*.
- [Hauskrecht et al., 2007] Hauskrecht, M., Valko, M., Kveton, B., Visweswaram, S., and Cooper, G. (2007). Evidence-based anomaly detection. In *Annual American Medical Informatics Association Symposium*, pages 319–324.
- [Haykin, 1994] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.
- [He and Carbonell, 2008] He, J. and Carbonell, J. (2008). Nearest-neighbor-based active learning for rare category detection. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 633–640. MIT Press, Cambridge, MA.
- [He et al., 2007] He, J., Carbonell, J., and Liu, Y. (2007). Graph-based semi-supervised learning as a generative model. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2492–2497, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Heard et al., 2010] Heard, N. A., Weston, D. J., Platanioti, K., and Hand, D. J. (2010). Bayesian anomaly detection methods for social networks. *Annals of Applied Statistics*, 4:645–662.
- [Hendrickson and Leland, 1995] Hendrickson, B. and Leland, R. (1995). A multilevel algorithm for partitioning graphs. In *Proceedings of Supercomputing*.
- [Jiang and Zhou., 2004] Jiang, Y. and Zhou., Z.-H. (2004). Editing training data for knn classifiers with neural network ensemble. In *Lecture Notes in Computer Science 3173*, pages 356–361.
- [Karypis and Kumar, 1999] Karypis, G. and Kumar, V. (1999). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20:359–392.

- [Kivinen et al., 2002] Kivinen, J., Smola, A. J., and Williamson, R. C. (2002). Online learning with kernels. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.
- [Kolar et al., 2010] Kolar, M., Song, L., Ahmed, A., and Xing, E. P. (2010). Estimating time-varying networks. *Annals of Applied Statistics*, 4:94–123.
- [Kveton et al., 2010a] Kveton, B., Valko, M., Philipose, M., and Huang, L. (2010a). Online semi-supervised perception: Real-time learning without explicit feedback. In *Proceedings of the 4th IEEE Online Learning for Computer Vision Workshop*.
- [Kveton et al., 2010b] Kveton, B., Valko, M., Rahimi, A., and Huang, L. (2010b). Semi-supervised learning with max-margin graph cuts. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 421–428.
- [Lazarevic and Kumar, 2005] Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, New York, NY, USA. ACM.
- [Lee and Wasserman, 2010] Lee, A. B. and Wasserman, L. (2010). Spectral connectivity analysis. *Journal of the American Statistical Association*, 0(0):1–15.
- [Luxburg, 2007] Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.
- [Ma and Perkins, 2003] Ma, J. and Perkins, S. (2003). Online novelty detection on temporal sequences. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–618, New York, NY, USA. ACM.
- [Madigan et al., 2002] Madigan, D., Raghavan, I., Dumouchel, W., Nason, M., Posse, C., and Ridgeway, G. (2002). Likelihood-based data squashing: a modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6(2):173–190.
- [Manevitz and Yousef, 2002] Manevitz, L. M. and Yousef, M. (2002). One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154.
- [Markou and Singh, 2003a] Markou, M. and Singh, S. (2003a). Novelty detection: a review, part 1: statistical approaches. *Signal Process.*, 83(12):2481–2497.
- [Markou and Singh, 2003b] Markou, M. and Singh, S. (2003b). Novelty detection: a review, part 2: neural network based approaches. *Signal Process.*, 83(12):2499–2521.
- [McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman and Hall, London, 2nd edition.
- [Mitra et al., 2002] Mitra, P., Murthy, C. A., and Pal, S. K. (2002). Density-based multiscale data condensation. *IEEE Transactions on PAMI*, 24(6):1–14.

- [Moonesignhe and Tan, 2006] Moonesignhe, H. D. K. and Tan, P.-N. (2006). Outlier detection using random walks. In *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 532–539, Washington, DC, USA. IEEE Computer Society.
- [Nene et al., 1996] Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library (COIL-100). Technical report, Columbia University.
- [OpenMP, 2008] OpenMP (2008). *OpenMP Application Program Interface – Version 3.0*. OpenMP Architecture Review Board.
- [Papadimitriou and Faloutsos, 2003] Papadimitriou, S. and Faloutsos, C. (2003). Cross-outlier detection. In Hadzilacos, T., Manolopoulos, Y., Roddick, J. F., and Theodoridis, Y., editors, *Advances in Spatial and Temporal Databases, 8th International Symposium, SSTD 2003, Santorini Island, Greece, July 24-27, 2003, Proceedings*, volume 2750, pages 199–213.
- [Pelleg and Moore, 2005] Pelleg, D. and Moore, A. W. (2005). Active learning for anomaly and rare-category detection. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1073–1080. MIT Press, Cambridge, MA.
- [Post and Harrison, 2008] Post, A. R. and Harrison, J. H. (2008). Temporal data mining. *Clin Lab Med*, 28(1):83–100, vii.
- [Rubin et al., 2005] Rubin, S., Christodorescu, M., Ganapathy, V., Giffin, J. T., Kruger, L., Wang, H., and Kidd, N. (2005). An auctioning reputation system based on anomaly detection. In *Proceedings of the 12th ACM conference on Computer and communications security, CCS '05*, pages 270–279, New York, NY, USA. ACM.
- [Sanchez et al., 2003] Sanchez, J., Barandela, R., Marques, A. I., Alejo, R., and J., B. (2003). Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters* 24, pages 1015–1022.
- [Schedlbauer et al., 2009] Schedlbauer, A., Prasad, V., Mulvaney, C., Phansalkar, S., Stanton, W., Bates, D. W., and Avery, A. J. (2009). What evidence supports the use of computerized alerts and prompts to improve clinicians’ prescribing behavior? *J Am Med Inform Assoc*, 16(4):531–538.
- [Scholkopf et al., 1999] Scholkopf, B., Platt, J. C., Shawe-taylor, J., Smola, A. J., and Williamson, R. C. (1999). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:2001.
- [Singh et al., 2008] Singh, A., Nowak, R. D., and Zhu, X. (2008). Unlabeled data: Now it helps, now it doesn’t. In *Advances in Neural Information Processing Systems 21, 2008*, pages 1513–1520.

- [Smola and Kondor, 2003] Smola, A. and Kondor, R. (2003). Kernels and regularization on graphs. In Schölkopf, B. and Warmuth, M., editors, *Proceedings of the Annual Conference on Computational Learning Theory and Kernel Workshop*, Lecture Notes in Computer Science. Springer.
- [Song et al., 2007] Song, X., Wu, M., and Jermaine, C. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645. Fellow-Sanjay Ranka.
- [Syed and Rubinfeld, 2010] Syed, Z. and Rubinfeld, I. (2010). Unsupervised risk stratification in clinical datasets: Identifying patients at risk of rare outcomes. In Fürnkranz, J. and Joachims, T., editors, *ICML*, pages 1023–1030. Omnipress.
- [Valizadegan and Tan, 2007] Valizadegan, H. and Tan, P.-N. (2007). Kernel based detection of mislabeled training examples. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*.
- [Valko et al., 2008] Valko, M., Cooper, G., Seybert, A., Visweswaran, S., Saul, M., and Hauskrecht, M. (2008). Conditional anomaly detection methods for patient-management alert systems. In *Workshop on Machine Learning in Health Care Applications in The 25th International Conference on Machine Learning*.
- [Valko and Hauskrecht, 2008] Valko, M. and Hauskrecht, M. (2008). Distance metric learning for conditional anomaly detection. In *Twenty-First International Florida Artificial Intelligence Research Society Conference*. AAAI Press.
- [Valko and Hauskrecht, 2010] Valko, M. and Hauskrecht, M. (2010). Feature importance analysis for patient management decisions. In *13th International Congress on Medical Informatics MEDINFO 2010*.
- [Valko et al., 2010] Valko, M., Kveton, B., Huang, L., and Ting, D. (2010). Online semi-supervised learning on quantized graphs. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*.
- [Valko et al., 2011] Valko, M., Valizadegan, H., Kveton, B., Cooper, G. F., and Hauskrecht, M. (2011). Conditional anomaly detection using soft harmonic functions: An application to clinical alerting. In *The 28th International Conference on Machine Learning Workshop on Machine Learning for Global Challenges*.
- [Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- [Verbaeten and Assche., 2003] Verbaeten, S. and Assche., A. V. (2003). Ensemble methods for noise elimination in classification problems. In *Proceeding of 4th International Workshop on Multiple Classifier Systems*.

- [Wahba, 1999] Wahba, G. (1999). *Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV*, pages 69–88. MIT Press, Cambridge, MA.
- [Williams and Seeger, 2001] Williams, C. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In *Neural Information Processing Systems, 2001*.
- [Yan et al., 2009] Yan, D., Huang, L., and Jordan, M. (2009). Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [Zhou et al., 2004] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Scholkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–328.
- [Zhu, 2008] Zhu, X. (2008). Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison.
- [Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919.
- [Zhu and Lafferty, 2005] Zhu, X. and Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning, ICML '05*, pages 1052–1059, New York, NY, USA. ACM.