NAIST-IS-DD661014

Doctoral Dissertation

Statistical and Graph-based Approaches to Small Sample and High Dimensional Data

Ikumi Suzuki

March 16, 2012

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Ikumi Suzuki

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Kazushi Ikeda	(Co-supervisor)
Professor Masashi Shimbo	(Co-supervisor)

Statistical and Graph-based Approaches to Small Sample and High Dimensional Data*

Ikumi Suzuki

Abstract

In recent years, machine learning has become a popular tool for analyzing various types of data. The goal of machine learning is to construct a model from existing data to make predictions for new data. To build (or select) accurate predictive models, two aspects of data must be verified: (1) Is the amount of training data large enough to predict unseen test data? (2) When the data is represented by a vector, is the number of dimensions small enough not to be affected by so-called "curse of dimensionality"? If any of these is violated, learning a predictive model becomes much harder, but these are not necessarily satisfied in real situations. This thesis deals with how to alleviate the problems incurred in such situations. For issue (1), we address the task of selecting a predictive model using a small number of training samples. In particular, we focus on developing a cancer diagnosis system that requires an accurate prediction from gene expression profiling (microarray) data. We propose a "min-max" model selection method based on the bootstrap resampling to obtain a reliable classifier. We show that our method is less susceptible to variation in the assessment of the occurrence data, indicating the effectiveness of risk-averse as a model selection criterion. For issue (2), we focus on a problem related to the high dimensionality of the data called *hubness* phenomenon, which was discovered only recently. We show the family of kernels based on the graph Laplacian is less prone to make hubs when used as a similarity measure. We found that these kernels indeed reduce hubness phenomemon in some cases, and in these cases they work well in ranking and classification tasks. This result suggests that the amount of hubs, which can be readily computed in an unsupervised fashion, can be a yardstick of whether Laplacian-based kernels work effectively for a given data.

*Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD661014, March 16, 2012.

Keywords:

Hub, Laplacian-based kernel, Microarray, min-max model selection

少数サンプルと高次元データへの確率とグラフに基づくアプローチ*

鈴木　郁美

内容梗概

近年，機械学習は，さまざまな種類のデータの解析に使われている．機械学習の目的は，新しいデータに対する予測を行うために，既存データを用いてモデルを構築することである．精度の高いモデルを構築（あるいは選択）するためには，訓練データを以下の二つの観点から調べ，対策を立てる必要がある．(1) 訓練データは，どんなテストデータの予測にも十分対応できるだけの数が集められているか？(2) 訓練データがベクトル値で表現されている場合，その次元数は「次元の呪い」の影響を受ける程に大きくないか？(1) に関して本研究で取り上げる問題は，訓練サンプル数の小さい場合の予測モデルの選択である．実際例として，正確な予測が求められる，遺伝子発現プロファイル（マイクロアレイデータ）による癌の診断に注目した．癌など病気の診断に用いる遺伝子発現量プロファイルは，得られるサンプル数が限られるために，実用化には判別器の信頼性が問題になる．交差検定によって複数の判別器の性能を評価し, その評価が最大となる判別器を選ぶことが行われてきた. 本研究では, テスト性能の分散を考慮することで, 悪い判別器が得られるリスクを回避してモデル選択を行う min-max 法を提案する．提案手法がデータ出現の偏りに起因した性能評価のばらつきに影響を受けにくく，リスク回避型のモデル選択基準として有効であることを示す. (2) に関して，データの高次元性に関わる問題を扱う．最近，新しい「次元の呪い」として,「ハブの出現」が報告された．本研究では，通勤時間カーネルを始めとするラプラシアンベースのカーネルを類似度がハブを抑制できる可能性がある性質を持つことを示す．実験から，ラプラシアンカーネルは常にではないがハブを抑えることがわかった. ハブが減少したデータセットでは，分類精度も向上した．このことから，教師なしで測れるハブの出現度合いを調べることにより，与えられたデータセットに対するラプラシアンカーネルの有効性を予め評価できると考えられる．

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 博士論文, NAIST-IS-DD661014, 2012 年 3 月 16 日.

キーワード

ハブ, ラプラシアンカーネル，マイクロアレイ，min-max モデル選択

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, machine learning techniques have been applied to various data analysis, such as building a prediction model or data mining. For example, in the field of natural language processing, document classification (or document categorization) is the task of assigning a document to one or more classes or categories, and machine learning is used to automate classification. In image processing, there is a large volume of research in identifying a person from his/her fingerprints/veins/iris, diagnosing diseases from x-ray images, and recognizing handwritten characters. For time-series data, such as variation of blood pressure, monthly river flow and daily share price of stocks, constructing statistical models is important to predict future values. To this end, machine learning techniques have been applied, not only in the field of engineering but also in medicine, biological science, and economics.

The purpose of (supervised) learning is to construct a model to predict test samples from training samples. To build high-performance models in practical settings, however, we recognize two issues.

(1) Are there enough training samples to predict test samples?

(2) When a sample is represented in a form of a feature vector, is the vector dimension not too high to be affected by curse of dimensionality?

1.2 Contribution

This thesis addresses the issue (1) by selecting a predictive model for small number of training samples. As a practical example, we focus on developing a cancer diagnosis system that requires an accurate prediction from gene expression profiling (microarray) data. Recently, microarray-based cancer diagnosis systems have been increasingly developed. However, cost reduction and reliability assurance of such diagnosis systems are still remaing problems in real clinical scenes. To reduce the cost, we need a supervised classifier involving the smallest number of genes, as long as the classifier is sufficiently reliable. To achieve a reliable classifier, we should assess candidate classifiers and select the best one. In the selection process of the best classifier, however, the assessment criterion must involve a large variance because of limited number of samples and non-negligible observation noise. Therefore, even if a classifier with a very small number of genes exhibited the smallest leave-one-out cross-validation (LOO) error rate, it would not necessarily be reliable because classifiers based on a small number of genes tend to show a large variance. We propose a robust model selection criterion, the min-max criterion, based on a resampling bootstrap simulation to assess the variance of estimation of classification error rates. We applied our assessment framework to four published real gene expression datasets and one synthetic dataset. We found that a state of-the-art procedure, weighted voting classifiers with LOO criterion, had a non-negligible risk of selecting extremely poor classifiers and, on the other hand, that the new min-max criterion could eliminate that risk. These finding suggests that our criterion presents a safer procedure to design a practical cancer diagnosis system.

To address the issue (2), we focus on the *hubness* phenomenon, which is a problem related to the high dimensionality of the data, and is only recently discovered by Radovanović et al. A "hub" is an object closely surrounded by, or very similar to, many other objects in the dataset. Recent studies by Radovanović et al. have indicated that in high dimensional spaces, objects close to the data centroid tend to become hubs. We show that the family of kernels based on the graph Laplacian makes all objects in the dataset equally similar to the centroid, and thus they are expected to make less hubs when used as a similarity measure. We investigate this hypothesis using both synthetic and real-world data. It turns out that these kernels suppress hubs in some cases but not always, and the results seem to be affected by the size of the data—a factor not discussed previously. However, for the datasets in which hubs are indeed reduced by the Laplacian-based kernels, these kernels work well in classification and information retrieval tasks. This result suggests that the amount of hubs, which can be readily

computed in an unsupervised fashion, can be a yardstick of whether Laplacian-based kernels work effectively for a given data.

1.3 Structure of the Thesis

To deal with the problem of high-dimensional data, Laplacian-based kernels are studied. As preliminary work, in Chapter 2, we experimentally show that a Laplacian-based kernel, specifically, the Laplacian diffusion kernel depreciates pivotal vertices having many links to surrounding vertices. In Chapter 3, we investigate the effect of Laplacian-based kernels on the recently-reported hubness phenomenon, which is a new type of curse of dimensionality that affects high-dimensional dataset.

To overcome the problem of small sample-size data, in Chapter 4, a robust model selection method is proposed for cancer diagnosis.

In Chapter 5, we conclude this thesis.

Chapter 2

A Graph-based Approach for Biomedical Thesaurus Expansion

2.1 Introduction

Biomedical thesauri using controlled vocabularies are a fundamental resource for information retrieval from biomedical literature. One successful example is the MeSH thesaurus[1] administrated by the National Library of Medicine (NLM). This thesaurus absorbs into its hierarchical structure the synonymous variations of PubMed[2] search queries, so that MeSH terms appearing in a query are augmented to include all the descendants of original query terms, making search results more comprehensive.

In parallel with the progress of biomedical research and development, new biomedical terms are constantly appearing. Hence the addition of new terms to current thesauri is an important task and indeed human specialists update the MeSH thesaurus annually by hand. In this chapter, we aim to build a system that supports human judgment regarding where to add a new term in a thesaurus. Using a large scale corpus of biomedical articles, we construct a graph whose vertices correspond to biomedical terms occurring in the corpus. We then compute the similarity scores between existing thesaurus terms and new terms by employing the Laplacian diffusion kernel matrix calculation. Finally, existing terms are ranked according to their similarity scores with a new term. We consider that such systems help editors map new terms to a thesaurus (e.g., editors can attach or insert a new term near the top ranked terms in the thesaurus tree structure).

[1] http://www.nlm.nih.gov/mesh/

[2] http://www.ncbi.nlm.nih.gov/pubmed/

Previous work [24] shows that cosine similarity works well for synonym acquisition, a task related to thesaurus expansion. This technique represents a term as a feature vector whose elements are values of feature functions determined using contexts surrounding the term in the corpus. The cosine similarity is the cosine of the angle between two vectors.

Our research is motivated by a concern regarding the application of cosine similarity to thesaurus expansion. New terms (i.e., targets to add to the thesaurus) can be considered to occur in narrower contexts in a corpus because they are newcomers in biomedical text. In the annual MeSH update for the 2009 version[3], more than 70% of newly added terms were attached as leaf nodes in the thesaurus tree structure, indicating that the majority of new terms had more specific senses.

On the other hand, general terms express broader concepts that reside near the top level of the thesaurus hierarchy, and thus occur in a variety of contexts in the corpus. As a result, general terms are likely to share contextual information with new terms, resulting in higher cosine similarity values by chance. When attempting to add highly specific terms to a thesaurus, a system that inherently returns general terms with high ranks is not beneficial, but this situation appears to be common in practice.

In order to adapt to this situation, we explore an approach using the Laplacian of a graph to represent a biomedical thesaurus structure. In our graph, general terms are equal to pivotal vertices that have many links to surrounding vertices. The problematic phenomena of topic drift [4] caused by pivotal vertices are well known in link analysis. Fortunately, it is reported that the approach based on graph Laplacian successfully solves the problem in tasks such as co-citation analysis, collaborative recommendation, and word sense disambiguation [12, 16, 32]. We therefore expect the approach to be advantageous also for thesaurus expansion, and attempt to confirm this supposition in this chapter.

2.2 Related work

There have been several studies for synonym acquisition, which is a task related to thesaurus expansion. Hagiwara et al. [24] attempt to use cosine similarity to find effective acquisition features by investigating what contextual information (e.g., word proximity or dependency between words) is useful in experimentation. They also apply Probabilistic Latent Semantic Indexing (PLSI) techniques to cope with data sparse-

[3] http://www.nlm.nih.gov/mesh/newd.html

ness problems caused by using word surface information as features [23]. Apart from feature selection and smoothing, Hagiwara [22] focuses on discriminating between similar and dissimilar word pairs by employing Support Vector Machines (SVMs), and Shimizu et al. [41] apply a method to tune feature weights as parameters of the Mahalanobis distance.

The above researchers focus solely on the distributional hypothesis [26], which states that similar terms occur in similar contexts. In contrast, the work of Blondel et al. [6] is similar to our own in that it takes into account graph structure. Vertices corresponding to terms are connected with directed links when they have a particular relation, for example in a dictionary when one appears in the definition of another. Using graph structure, Blondel et al. propose an algorithm that can be regarded as a generalization of Kleinberg's HITS for synonym acquisition.

Both synonym acquisition and thesaurus expansion aim to rank terms according to their associated similarity scores calculated for a query term, but the tasks differ importantly in evaluation. No canonical criteria are available with respect to what extent two terms can be presupposed to be synonymous in a synonym acquisition task. In contrast, in a thesaurus expansion task, the thesaurus being expanded inherently gives canonical relationships or distances between terms. For this reason, we evaluate performance of tested methods by examining whether a set of ranked terms obtained from each method successfully includes terms that actually reside in the neighborhood of a query term in the thesaurus.

2.3 Laplacian-based Kernels for Graph Vertices

We present a brief review of Laplacian-based kernels.

Let $\mathscr{G}$ be an undirected graph with n vertices, and let $\mathbf{A}$ be its adjacency matrix. The edges of $\mathscr{G}$ may have positive weights representing the degree of similarity between vertices. In this case, $\mathbf{A}$ is an affinity matrix holding the edge weights as its components. The *(combinatorial) Laplacian* $\mathbf{L}$ of $\mathscr{G}$ is an $n \times n$ matrix defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{2.1}$$

where $\mathbf{D}$ is a diagonal matrix with diagonals $[\mathbf{D}]_{ii} = \sum_j [\mathbf{A}]_{ij}$. $\mathbf{L}$ is positive semidefinite and has n orthogonal eigenvectors $\mathbf{u}_i$ and n corresponding eigenvalues λ_i. We assume that the indices for eigenvalues/eigenvectors are arranged in ascending order of eigen-

values, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. A well-known property of $\mathbf{L}$ is that $\lambda_1 = 0$ and $\mathbf{u}_1 = \mathbf{1}$ (a vector of all 1's).

2.3.1 Laplacian-based Kernels

In machine learning community, graph Laplacian has been used as the building block of various kernels defining inner products between vertices. Below are the most popular of such Laplacian-based kernels.

The commute-time kernels [40]

$$\mathbf{L}_{\mathrm{CT}} = \mathbf{L}^{+} \quad \text{(pseudo-inverse of } \mathbf{L}\text{)}, \tag{2.2}$$

Regularized Laplacian [10, 42]

$$\begin{aligned} \mathbf{L}_{\mathrm{RL}} &= (\mathbf{I} + \beta\mathbf{L})^{-1} \\ &= \sum_{k=0}^{\infty} \beta^k (-\mathbf{L})^k \\ &= I + \beta(-\mathbf{L}) + \beta^2(-\mathbf{L})^2 + \beta^3(-\mathbf{L})^3 + \cdots, \end{aligned} \tag{2.3}$$

(Laplacian) diffusion kernels [33]

$$\begin{aligned} \mathbf{L}_{\mathrm{DF}} &= \exp(-\beta\mathbf{L}) \\ &= \sum_{k=0}^{\infty} (\frac{\beta}{k!})^k (-\mathbf{L})^k \\ &= I + \beta(-\mathbf{L}) + \frac{\beta^2}{2!}(-\mathbf{L})^2 + \frac{\beta^3}{3!}(-\mathbf{L})^3 + \cdots, \end{aligned} \tag{2.4}$$

where $\beta(\geq 0)$ is a parameter of the regularized Laplacian and the (Laplacian) diffusion kernels. Note that while we do not discuss them in this thesis, variations of these kernels exist which use the *normalized Laplacian* $\mathscr{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ [11] in place of $\mathbf{L}$ in their definition.

2.3.2 Transformation of Laplacian-based Kernels

The Laplacian-based kernels introduced in section 2.3.1 can be interpreted as transformations of Laplacian $\mathbf{L}$ through eigenvalue regularlization [42]. To be precise, all

the Laplacian-based kernels above (henceforth denoted by $\mathbf{K}$) can be decomposed as follows, using n pairs of eigenvalues and eigenvectors $\{(\lambda_i, \mathbf{u}_i)\}$ $(i = 1, \ldots, n)$ of $\mathbf{L}$.

$$\mathbf{K} = \sum_{i=1}^{n} r(\lambda_i)\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}, \tag{2.5}$$

where $r : [0, \infty) \rightarrow [0, \infty)$ is a *regularization operator*, which characterizes each Laplacian-based kernel. For the three kernels above,

The commute-time kernels

$$r(\lambda) = \begin{cases} 0, & \lambda = 0; \\ 1/\lambda & \lambda \neq 0, \end{cases} \tag{2.6}$$

Regularized Laplacian

$$r(\lambda) = 1/(1 + \beta\lambda), \tag{2.7}$$

(Laplacian) diffusion kernels

$$r(\lambda) = \exp(-\beta\lambda). \tag{2.8}$$

As Eq. (2.5) shows, Laplacian-based kernels have the same eigenvectors as Laplacian $\mathbf{L}$. Their eigenvalues, on the other hand, are transformed by function $r(\cdot)$. To suppress the contribution of large λ's, $r(\cdot)$ is in general a non-increasing function.

In the rest of the thesis, we focus on the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$ and the regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$. Laplacian diffusion kernels $\mathbf{L}_{\mathrm{DF}}$ show properties similar to $\mathbf{L}_{\mathrm{RL}}$.

2.3.3 Regularized Laplacian and parameter β

In this chapter, the Laplacian kernels are employed as similarity measures, so that the off-diagonal elements of the regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ and the commute-time kernel matrix $\mathbf{L}_{\mathrm{CT}}$ are used. The diagonal elements are not used since it is self similarity.

Notice that as parameter β of the regularized Laplacian approaches to zero, the off-diagonal elements of the regularized Laplacian become proportional to those of the original adjacency matrix, and as parameter β of the regularized Laplacian tends to infinity, the off-diagonal elements of the regularized Laplacian become proportional to those of the commute-time kernels matrix $\mathbf{L}_{\mathrm{CT}}$

The case of large β

According to the form of Eq. 2.5, the regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ is written as

$$\mathbf{L}_{\mathrm{RL}} = \sum_{i=1}^{N} \frac{1}{1+\beta\lambda_i} \mathbf{u}_i \mathbf{u}_i^T. \tag{2.9}$$

When parameter β takes large enough value, this Eq. 2.9 is then written as,

$$\begin{aligned} \mathbf{L}_{\mathrm{RL}} &= \mathbf{u}_1 \mathbf{u}_1^T + \sum_{i=2}^{N} \frac{1}{1+\beta\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \\ &\approx \mathbf{u}_1 \mathbf{u}_1^T + \frac{1}{\beta} \sum_{i=2}^{N} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \qquad (\beta\lambda_2 \gg 1), \end{aligned} \tag{2.10}$$

note that $\lambda_1 = 0$.

Also the commute-time kernel matrix $\mathbf{L}_{\mathrm{CT}}$ is written in the form of Eq. 2.5,

$$\mathbf{L}_{\mathrm{CT}} = \sum_{i=2}^{N} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T.$$

Substituting this into Eq. 2.10, we obtain

$$\mathbf{L}_{\mathrm{RL}} \approx \mathbf{u}_1 \mathbf{u}_1^T + \frac{1}{\beta} \mathbf{L}_{\mathrm{CT}} \qquad (\beta\lambda_2 \gg 1). \tag{2.11}$$

Note that $\mathbf{u}_1 = [1, \ldots, 1]^T$, therefore all components of $\mathbf{u}_1 \mathbf{u}_1^T$ are 1. Therefore, as value of β increases, the regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ becomes a matrix such that a constant value is added to the commute-time kernel matrix $\mathbf{L}_{\mathrm{CT}}$ multiplied by a constant.

The case of small β

The regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ is expressed as a power series shown in Eq. (2.3) when $\beta\lambda_N < 1$. When β is close to 0, latter part than square terms can be ignored, so that the regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ can be approximated as

$$\mathbf{L}_{\mathrm{RL}} \approx \mathbf{I} + \beta(-\mathbf{L}) = \mathbf{I} - \beta\mathbf{D} + \beta\mathbf{A} \qquad (\beta \ll 1). \tag{2.12}$$

The diagonal matrices $\mathbf{I}$ and $\mathbf{D}$ do not affect the off-diagonal elements of the similarity matrix and hence do not affect k-nearest neighbor lists either. Therefore, when parameter β is close enough to 0, k-nearest neighbor list based on the regularized Laplacian matrix $\mathbf{L}_{\mathrm{RL}}$ becomes that of an original adjacency matrix $\mathbf{A}$.

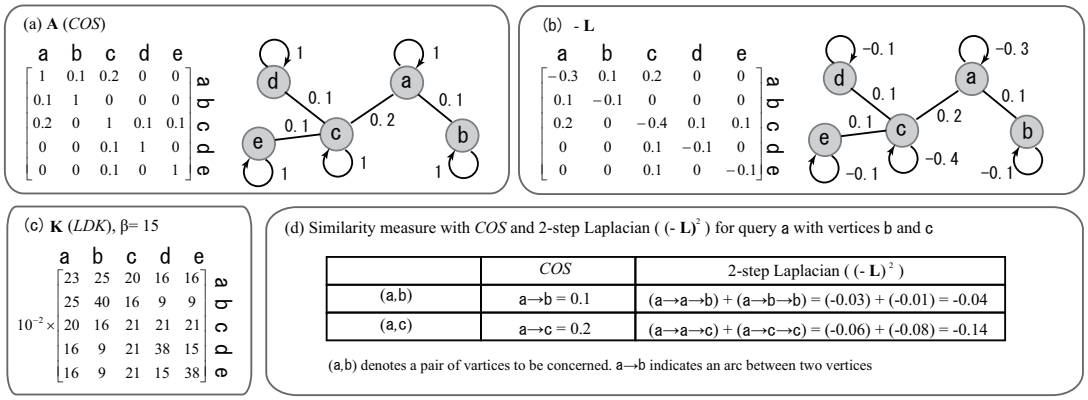


Figure 2.1: A small example of *COS* and *LDK* calculation. Given a graph with an adjacency matrix $\mathbf{A}$ in panel (a), $-\mathbf{L}$ and *LDK* are calculated as shown in panel (b) and (c), respectively. Panel (d) shows self-loop effects on traversing the graph Laplacian with 2 steps.

2.4 Method

In this section, we describe two ways to measure similarity between biomedical terms, namely (1) cosine similarity (*COS*) and (2) the Laplacian diffusion kernel based similarity (*LDK*). We exemplify the difference using a small example.

Suppose that there are N types of biomedical terms occurring in a given corpus. For each $term_n$ ($n = 1, \ldots, N$), we develop a feature vector $\mathbf{v}_n$ whose elements are values of feature functions determined upon contexts surrounding the term in the corpus. We then compute the normalized feature vector $\mathbf{z}_n$ by $\mathbf{v}_n / \|\mathbf{v}_n\|$. The cosine similarity between $term_i$ and $term_j$ is equal to the inner product of the normalized feature vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, that is, $\mathbf{z}_i \cdot \mathbf{z}_j$.

We then construct a graph, each of whose vertices corresponds to a biomedical term, and the weight on a link connecting vertices i and j is equal to $\mathbf{z}_i \cdot \mathbf{z}_j$. The matrix $\mathbf{A}$ whose (i, j)-element takes the value of *COS* for $term_i$ and $term_j$, $\mathbf{z}_i \cdot \mathbf{z}_j$, is called the adjacency matrix of the graph. We illustrate an example of a graph and its adjacency matrix in Figure 2.1(a).

We then compare the two similarity measures, *COS* and *LDK*, respectively given by the elements of matrices $\mathbf{A}$ and $\mathbf{L}_{\mathrm{DF}}$, by examining a small example displayed in Figure 2.1. In Figure 2.1(a), we observe that the *COS* score for a pair of vertices (a, c) is larger than that for vertices (a, b). On the other hand, Figure 2.1(c) shows that the

LDK score for (a, c) becomes smaller than the score for (a, b). Likewise, while the *COS* score for (c, a) is larger than that for (c, d), the *LDK* score for (c, a) turns out to be smaller than that for (c, d).

In order to examine what causes the difference between *COS* and *LDK*, notice that $\mathbf{L}_{\mathrm{DF}}$ consists of $(-\mathbf{L})^k$ whose (i, j)-element can be considered as a path score traversing from the i to the j vertex with k-steps. Accordingly, *LDK* especially discounts the path scores via pivotal vertices. As an example in Figure 2.1(d), we focus on "2-step Laplacian" $(-\mathbf{L})^2$ that is one of the components of $\mathbf{L}_{\mathrm{DF}}$. There, we notice that the $(-\mathbf{L})^2$ scores for (a, b) and (a, c) are all negative. Also the absolute value of the score for (a, c) is larger than that for (a, b). This is because the weight on the self-loop of vertex c has a larger negative weight than that on vertex b.

Recall that $\mathbf{L}$ is defined as $\mathbf{D} - \mathbf{A}$. The weight of the self-loop in $\mathbf{L}$ is calculated as the sum of the weights on all links except the self-loop in $\mathbf{A}$. As a result, a pivotal vertex that shares many links with other vertices, such as the vertex c in Figure 2.1(b), tends to have a larger negative weight on the self-loop in $-\mathbf{L}$. This implies that *LDK* tends to depreciate pivotal vertices. In the following sections, we examine the implications of this tendency using real biomedical data.

2.5 Experiment

The experiment presented in this section has two objectives. First, to confirm whether the Laplacian diffusion kernel depreciates pivotal vertices on a graph constructed from a biomedical corpus, and second, to simulate the performance of the Laplacian diffusion kernel and cosine similarity in thesaurus expansion.

2.5.1 Setup

We used the GENIA biomedical corpus[4] consisting of 1,999 MEDLINE abstracts, along with the MeSH thesaurus. The thesaurus comprised 62,932 tokens and 2,701 types of MeSH terms we identified in GENIA. Of the 2,701 term types, 500 were selected as query terms (i.e., pseudo target terms to add to MeSH) upon the conditions that: (1) a query term resided as a unique node in the MeSH thesaurus tree structure (i.e., did not have multiple meanings); and (2) the unique node was a leaf node. We set

[4]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi

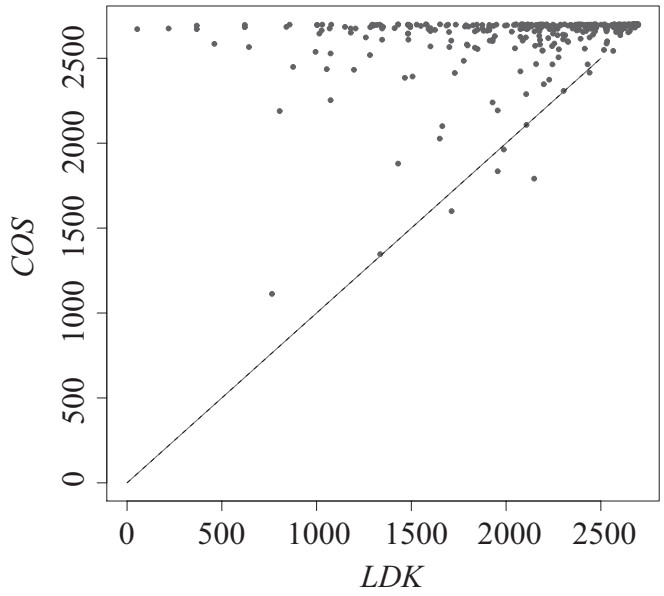


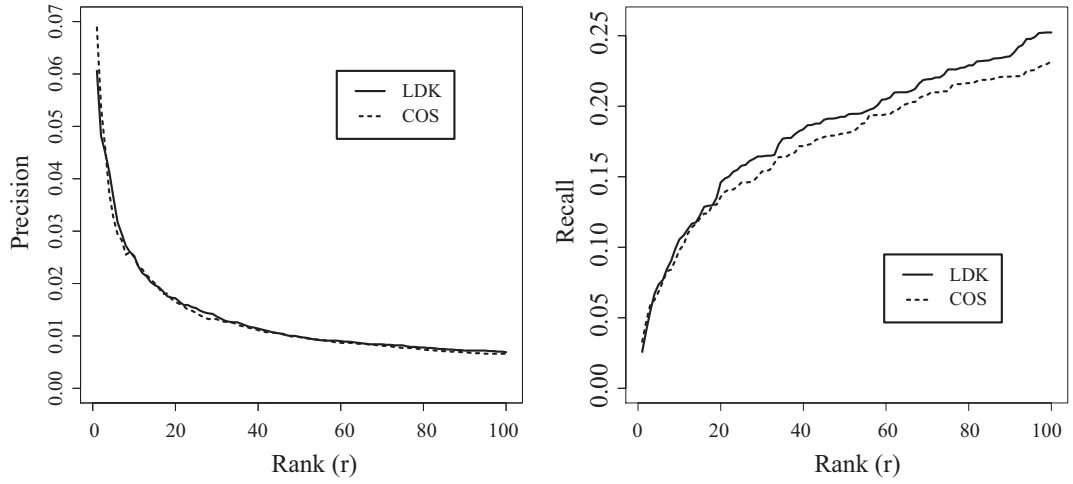
Figure 2.2: Comparison between *COS* and *LDK*; the number of links attached to the top ranked vertices.

uniqueness as a condition because word sense disambiguation is not a task dealt with in this work, and we limited to leaf nodes because most of the newly added terms for the 2009 version of MeSH were attached as leaf nodes in the thesaurus tree structure.

For each of 62,932 tokens, we extracted content words appearing in a sentence around the token. Content words were defined as all non function words (i.e. nouns, verbs, adjectives and adverbs). The content words for all tokens of a single type were collapsed into a single pool and considered as term features. After applying tf-idf weighting, we obtained a feature vector for each of the 2,701 MeSH terms appearing in GENIA.

We then constructed a graph with 2,701 vertices, and calculated both cosine similarity (*COS*) and the Laplacian diffusion kernel matrix based similarity (*LDK*)[5] as described in the previous section. Finally, for each of the 500 query terms, 2,700 terms (excluding the query term itself) were ranked according to their similarity scores with the original term.

[5]We set $\beta = 0.01$.



(a) Precision (b) Recall

Figure 2.3: Comparison between *COS* and *LDK*; Precision and recall curves in simulating thesaurus expansion.

2.5.2 Results

In order to confirm that *LDK* does not prefer pivotal vertices in comparison with *COS*, we compare the number of links to their top-ranked vertices, as shown in Figure 2.2. A point in the figure corresponds to a query term, and the vertical and horizontal axes denote the number of links attached to vertices selected by *COS* and *LDK* respectively. If a point lies above the diagonal (dotted line), it means that *LDK* selects a vertex having fewer links compared to *COS*. We omit points lying on the diagonal for brevity and highlight the fact that very few points in the figure lie below the diagonal. It follows that *LDK* has a tendency to depreciate pivotal vertices, which is consistent with our finding in the previous section.

Next we evaluate the performance of *COS* and *LDK* in simulating thesaurus expansion. Recall that the pseudo query terms we are adding to MeSH are already MeSH terms with known sets of neighboring terms. We therefore are interested in how many of those neighboring terms are successfully included in the set of top r ranked terms: the more, the better. Neighbor terms include the parent and siblings of a query term in the thesaurus tree structure. We use averaged precision, recall and F1-score over query terms as evaluation measures. For each query we calculate precision as the fraction of top r ranked terms that are neighbors, and recall as the fraction of neighbor terms

Table 2.1: Comparison between *COS* and *LDK*; Precision (P), recall (R), and F1-score (F). Numerals in brackets denote top r ranked terms concerned.

	(r)	P	R	F
COS	(1)	0.069	0.033	0.041
LDK	(1)	0.061	0.023	0.033
COS	(10)	0.025	0.098	0.036
LDK	(10)	0.025	0.106	0.038
COS	(20)	0.017	0.136	0.027
LDK	(20)	0.017	0.146	0.029
COS	(50)	0.010	0.181	0.018
LDK	(50)	0.010	0.193	0.018

that are in top r ranks. Figure 2.3 and Table 2.1 show the results. While *COS* achieves higher performance around the top few ranks, *LDK* outperforms *COS* in recall (i.e., *LDK* successfully picks up more relevant terms than *COS*) when we consider ranks greater than around 20, indicating that the two approaches are complementary.

2.6 Conclusion

This chapter investigated the effect of employing a Laplacian diffusion kernel matrix for the task of determining the correct neighboring terms of a new term being added to a thesaurus. We confirmed that the method depreciates pivotal vertices on a graph, and showed that it does pick up more relevant terms, exhibited as higher recall values, than cosine similarity in a simulation study.

LDK does not significantly improve upon *COS* in this work. One possible reason is that the GENIA corpus were collected using only three MeSH terms,"human", "blood cells", and "transcription factors" from MEDLINE abstracts, and therefore the terms in the corpus are highly connected with one another in the graph. Future work includes examining *LDK* performance with the whole MEDLINE articles with which the induced term graph becomes more sparse. Another concern is that neither approach reliably predicts the correct neighboring MeSH terms. To this end, feature weight tuning or feature engineering, such as limiting context words (features) to nominals or using a stopword list, might be an effective way to improve the performances.

2.7 Summary

The addition of new terms to biomedical thesauri is important for keeping pace with new research. In the context of a thesaurus expansion task, we investigate the property of Laplacian diffusion kernel matrices that depreciate pivotal vertices having many links to surrounding vertices. We confirm that this property can be seen on the Laplacian matrix of a graph that we construct from the GENIA corpus (a subset of MEDLINE abstracts) and simulate thesaurus expansion by employing either the Laplacian diffusion kernel matrix, or the adjacency matrix (i.e., cosine similarity), to determine the correct position for new biomedical terms being added to the MeSH thesaurus. Whilst results do not show the desired precision, our approach is shown to be complementary to calculation of cosine similarity between thesaurus terms and we recognize directions for future work.

Chapter 3

Effectiveness of Laplacian-based Kernels in Hub Reduction

3.1 Introduction

In recent studies, Radovanović et al. investigated *hubs* that emerge in high dimensional space [37, 38]. A hub is an object similar (or close) to many other objects in a dataset. Radovanović et al. observed that hub objects emerge as dimension increases, for a number of common similarity or distance measures. They also made a notable finding that the objects closer (more similar) to the data mean, or *centroid*, tend to become hubs.

Hub objects emerge even in space of moderately high dimension (e.g., 50-dimensions), whereas systems for real data analysis, such as those for natural language processing, often deal with more than one million features (dimensions).

As Radovanović et al. have pointed out, hubs impair the accuracy of k-nearest neighbor (k-nn) classification. In k-nn classification, the label of a test object is predicted by the (weighted) majority voting of the k-nn objects whose labels are known. If test objects follow the same distribution as that of the objects in the dataset, hubs in the dataset should frequently appear in the k-nn list for test objects as well. As a result, hub objects pose strong bias on the predicted labels, causing the classification results to be inaccurate. As we will discuss in a later section, hubs also impair information retrieval, and the label propagation methods for semi-supervised classification.

In this thesis, we examine if Laplacian-based kernels, such as the commute-time kernels [40] and the regularized Laplacian [10, 42], are effective for reducing hubs. We explore Laplacian-based kernels because in the implicit feature space induced by these

kernels, the inner product with the centroid is uniform for every object in the dataset; thus, no objects are closer to the centroid. According to Radovanović et al., objects close to the centroid become hubs, and we expect these kernels are more robust to the hubness phenomenon. We empirically examine if Laplacian-based kernels reduce hubs and consequently improve the performance of information retrieval as well as multi-class and multi-label k-nearest neighbor classification.

3.2 Hubs in High Dimensional Space

High dimensionality causes various problems that go under the name of *curse of dimensionality*. The most well-known "curse" includes overfitting [27, 5] and distance concentration [3, 17].

The "emergence of hubs" is a new type of the curse which has been discovered only recently [37]. This phenomenon particularly affects methods based on nearest neighbor search, i.e., those which list objects similar (or near) to a query object according to a certain similarity (or distance) measure. In a high dimensional space, some objects become *hubs*, which are the objects that occur in the nearest neighbor list of many objects. Since hubs nearly always included in the search result irrespective of query objects, such objects render search results less meaningful.

We can check whether or not hubs exist in a dataset by counting the number of times that each object $\mathbf{x}$ appears in the k-nearest neighbor list of other objects. Let this number be $N_k(\mathbf{x})$. If hubs exist in the dataset, the distribution of N_k should skew to the right (provided that $k \ll n$, where n is the number of the objects).

Now we illustrate the emergence of hubs using synthetic data. Following [37], we generate a dataset of 500 objects, each of which is a d-dimensional binary vector. For each dimension $i = 1, \ldots, d$, we first sample a real number from the log-normal distribution with mean 5 and variance 1, and compute its rounded integer n_i. Then we choose n_i objects (vectors) out of 500 uniformly at random, and assign 1 to their ith component. After 500 d-dimensional binary vectors are generated in this way, we measure their pairwise similarity by the cosine of the angle between them.

The histograms of N_{10} frequency for two datasets with different dimensions d ($d = 10, 50$) are shown in the top panels of Figure 3.1. We can see objects with extremely large N_{10} values (e.g., the point at $N_{10} = 60$) in the top right panel (50-dimensional data), while no such points can be seen for 10-dimensional data.

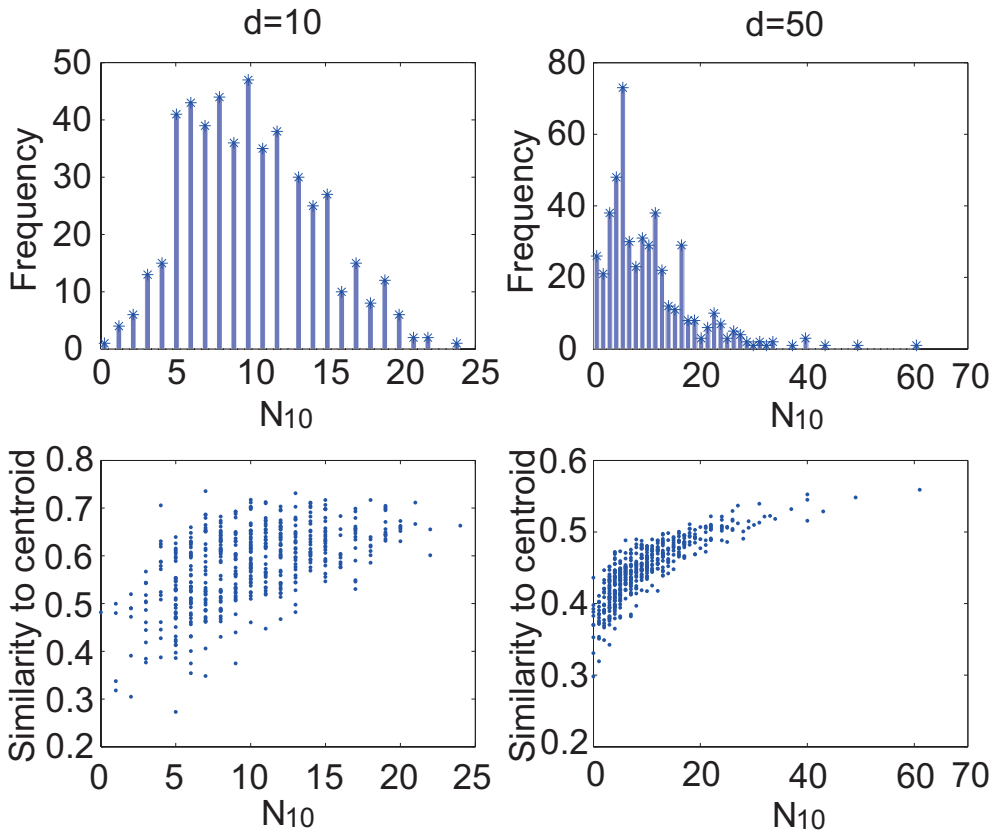


Figure 3.1: Top panels: Histograms of N_{10} frequency for two synthetic datasets in low ($d = 10$) and high ($d = 50$) dimensional feature spaces. Bottom panels: Scatter plots of the N_{10} value of an object against its similarity to the centroid. Each dot corresponds to a data object.

Another important finding by Radovanović et al. is that in high dimensional spaces, objects similar (or close) to the data mean (centroid) tend to become hubs. We can verify this with the dataset of 50-dimensional vectors above. The bottom panels of Figure 3.1 are the scatter plots of N_{10} values of the data objects against their cosine similarity to the centroid. For $d = 50$ (high-dimensional data; bottom-right), N_{10} values show a strong correlation with the similarity to the centroid, whereas for $d = 10$ (low-dimensional data; bottom-left), the correlation is much weaker.

3.3 The Hubness Phenomenon and Laplacian-based Kernels

If objects close to the data centroid tend to become hubs, a possible direction to their reduction should be to seek a similarity (or distance) measure which evaluates all objects equally similar to (or distant from) the centroid. We show that the Laplacian-based kernels indeed give measures which meet this requirement.

3.3.1 Centroid in the kernel-induced feature space

Suppose we have n data objects, $\mathscr{X} = \{\mathbf{x}_i\}$ $(i = 1, \ldots, n)$ in a vector space $\mathbb{D}$. We are also given a kernel $\mathbf{K}$, which, for now, is not necessarily the Laplacian-based kernels introduced above.

Let $\mathbb{F}$ be the implicit feature space induced by kernel $\mathbf{K}$, and $\phi(\cdot)$ be its associated feature mapping; i.e., a mapping of an object in $\mathbb{D}$ to its image in $\mathbb{F}$. Abusing notation, we also denote by $\mathbf{K}$ its $n \times n$ Gram matrix computed for the dataset. Thus, component $[\mathbf{K}]_{ij}$ of matrix $\mathbf{K}$ is the inner product of $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ in $\mathbb{F}$, or,

$$[\mathbf{K}]_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle.$$

And the data centroid in the feature space $\mathbb{F}$, which we denote by $\overline{\phi}$, is given by

$$\overline{\phi} = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i).$$

Note that $\overline{\phi}$ differs from the data centroid in the original vector space $\mathbb{D}$, and, in general, from its image in $\mathbb{F}$, because $\phi(\cdot)$ can be non-linear.

Now the inner product between $\phi(\mathbf{x}_i)$ and the data centroid $\overline{\phi}$ in $\mathbb{F}$ is

$$\langle \phi(\mathbf{x}_i), \overline{\phi} \rangle = \langle \phi(\mathbf{x}_i), \frac{1}{n} \sum_{j=1}^{n} \phi(\mathbf{x}_j) \rangle = \frac{1}{n} \sum_{j=1}^{n} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$
$$= \frac{1}{n} \sum_{j=1}^{n} [\mathbf{K}]_{ij} = \frac{1}{n} [\mathbf{K}\mathbf{1}]_i. \tag{3.1}$$

Thus, it is the mean of the inner products between the ith object and all objects in the dataset, taken in the feature space induced by $\mathbf{K}$. The last two equalities show that this quantity can be calculated simply by taking the mean of the ith row of the Gram matrix $\mathbf{K}$.

3.3.2 Laplacian-based kernels and similarity to the centroid

We now restrict $\mathbf{K}$ to Laplacian-based kernels, i.e., those which can be expressed as in Eq. (2.5). We show that Laplacian-based kernels define similarity measures which make the data centroid equally similar to all objects in the dataset.

Because Laplacian-based kernels assume that the data is represented as a graph, we treat the vector dataset $\mathscr{X}$ as a fully-connected graph. In this graph, data objects $\mathbf{x}_i$ corresponds to vertices, and edge weights are given by the pairwise similarity of objects measured in the original vector space $\mathbb{D}$.[1] In other words, the weighted adjacency matrix $\mathbf{A}$ of this fully-connected graph is given by the all-pairs similarity matrix for the dataset computed in $\mathbb{D}$. There may be many ways to measure similarity, but we only require that the similarity score be non-negative and symmetric; hence $[\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} \geq 0$ for all i, j. Given such an $\mathbf{A}$, we compute the graph Laplacian and then a Laplacian-based kernel $\mathbf{K}$, e.g., using one of Eqs. (2.2)– (2.4).

Now, recall that the Laplacian-based kernels share the same eigenvectors as the Laplacian $\mathbf{L}$ from which they are computed, but the eigenvalues are transformed by $r(\cdot)$; see Eq. (2.5). In particular, for the smallest eigenvalue λ_1 of $\mathbf{L}$ and its corresponding eigenvector $\mathbf{u}_1$, it holds that $\mathbf{K}\mathbf{u}_1 = r(\lambda_1)\mathbf{u}_1$. And since $\mathbf{u}_1 = \mathbf{1}$ and $\lambda_1 = 0$, we have

$$\mathbf{K}\mathbf{1} = r(0)\mathbf{1}. \tag{3.2}$$

[1] If a distance measure is given instead of similarity, we assume it is converted to a similarity in a standard way, e.g., by taking its reciprocal, or by using a Gaussian kernel.

By Eq. (3.1), the left-hand side of this equation becomes

$$\mathbf{K1} = n \begin{bmatrix} \langle \phi(\mathbf{x}_1), \overline{\phi} \rangle \\ \vdots \\ \langle \phi(\mathbf{x}_n), \overline{\phi} \rangle \end{bmatrix}. \tag{3.3}$$

On the other hand, the right-hand side of Eq. (3.2) is a constant vector whose components are all equal. It follows that all the components in Eq. (3.3) are equal. In other words,

$$\langle \phi(\mathbf{x}_1), \overline{\phi} \rangle = \langle \phi(\mathbf{x}_2), \overline{\phi} \rangle = \cdots = \langle \phi(\mathbf{x}_n), \overline{\phi} \rangle$$

Thus, in the feature space induced by $\mathbf{K}$, the inner products between the centroid and all object in the dataset are equal.

Remark The above property holds only if the components (inner products in the feature space) of Laplacian-based kernels $\mathbf{K}$ are used as they are as similarity scores. That is, the similarity to the centroid may not be uniform if the closeness of objects is measured by distance in $\mathbb{F}$, i.e., via

$$d(\mathbf{x}_i, \mathbf{x}_j)_{\mathbb{F}} = ([\mathbf{K}]_{ii} + [\mathbf{K}]_{jj} - 2[\mathbf{K}]_{ij})^{1/2}. \tag{3.4}$$

We will show in later experiments that using distance in the feature space of Laplacian-based kernels in fact promotes hubs, and is always a bad idea.

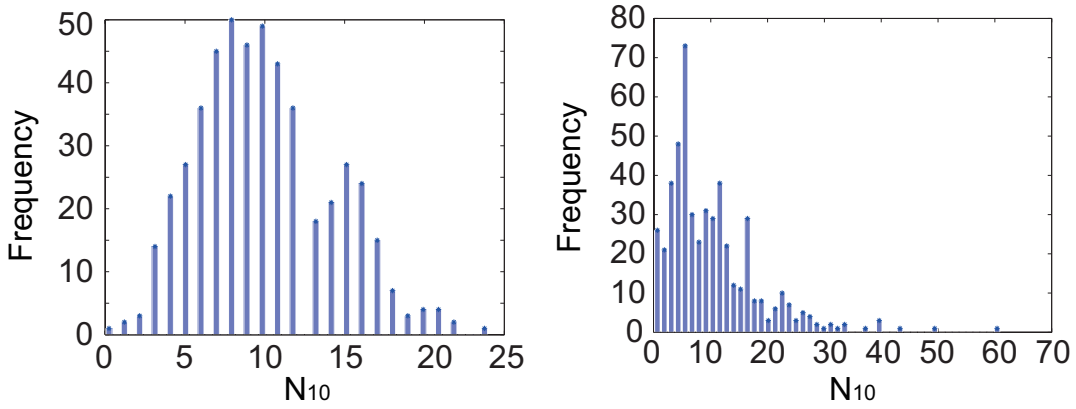
According to Radovanović et al., objects close (or similar) to the centroid become hubs. As shown above, Laplacian-based kernels provide a similarity measure which makes data objects equally similar to the centroid. For this reason, we can expect them to suppress emergence of hubs.

3.4 Experiments

We apply Laplacian-based kernels to real and synthetic datasets to see whether hubs are reduced by these kernels.

3.4.1 Synthetic data

First, as an illustration, we apply the commute-time kernels on the same 50-dimensional dataset we used previously to plot Figure 3.1. Figure 3.2(a) shows the histograms of



(a) Commute-time kernels ($\mathbf{L}_{\mathrm{CT}}$) (b) Cosine similarity (Cos)

Figure 3.2: Histograms of N_{10} frequency for the synthetic 50-dimensional dataset of Figure 3.1: (a) commute-time kernels and (b) cosine similarity.

N_{10} frequency for the commute-time kernels. For ease of comparison, (b) duplicates the top-right panel of Figure 3.1, which plots the histogram for cosine similarity. We see that with the commute-time kernels, no objects exhibit extremely large N_{10} values. Hence, the kernel has worked as expected for this dataset, and mitigated the hubness phenomenon.

3.4.2 Real data

For real data, we examine not only whether hubs are reduced by Laplacian-based kernels, but also whether they contribute to improve accuracy in tasks that uses these datasets. We consider three tasks: (1) ranking (information retrieval), (2) multi-class classification, and (3) multi-label classification. These tasks are chosen because they require fine-grained similarity measures to distinguish individual data objects, which are not necessary for a simple task such as binary classification.

Ranking task

We rank biomedical terms in the MeSH thesaurus[2], to simulate mapping a new term onto the thesaurus. For each term in MeSH, we rank other terms by the similarity

[2]http://www.nlm.nih.gov/mesh/2009/introduction/introduction.html

of contexts in which they appear, collected from abstracts in MEDLINE 2009[3]. The baseline measure evaluates the context similarity of terms by the cosine between "bag-of-words" feature vectors, which consist of the frequency of words occurring in the neighborhood of the terms in the corpus. We then compare this cosine similarity with the regularized Laplacian and commute-time kernels computed from the cosine similarity matrix.

In this task, a similarity measure is deemed better if it ranks terms located near the query term in the MeSH thesaurus tree higher in the ranking for the query term. Because different query terms have different nearby terms in the MeSH tree, the similarity measure is required to return distinct rankings for each query term. If hub objects (terms) exist that tend to take higher positions in many rankings, they are considered to be harmful.

In this experiment, we make four datasets, each of which corresponds to the set of terms under the top categories A, B, C, and D of the MeSH tree.

Multi-class classification

For multi-class classification, we use two document classification datasets: Reuters-52[4], and TDT2-30[5]. A document in these datasets is classified into one of 52 and 30 categories, respectively. For Reuters-52, we used the default training-test data split accompanying the dataset. For TDT2-30, we randomly split the data into halves. For these tasks, we classify test documents by k-nearest neighbor ($k$nn) classification. The similarity measures used with $k$nn are the cosine between bag-of-words feature vectors of documents, and the regularized Laplacian and commute-time kernels from the cosine similarity. Parameter k is chosen by cross validation using training data.

We also employ Naive-Bayes classifier (NB) for multi-class classification, as another baseline.

It is worth noting the large number of categories in the datasets (52 and 30). This makes difficult the application of support vector machines and other high-performance classifiers for binary classification.

[3] We limited the abstracts to those published in year 2000 or later.

[4] http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html

[5] http://www.zjucadcg.cn/dengcai/Data/TextData.html

Multi-label classification

In multi-label classification tasks, a document may be associated with one or more categories. For these tasks, we use the Enron and the Bibtex datasets[6]. The classification procedure follows that of multi-class classification, with one exception that we use the ML-*k*nn algorithm [51] in place of *k*nn classification.

The number of unique assignment of category combination to an object is 753 in Enron, and 2856 in Bibtex, which are again extremely large.

Evaluation metrics

For all tasks, we compare cosine similarity (Cos) with the regularized Laplacian ($\mathbf{L}_{\mathrm{RL}}$) and commute-time kernels ($\mathbf{L}_{\mathrm{CT}}$), in terms of the degree of hub emergence and the task performance.

Following Radovanović et al. [37], we evaluate the degree of hubness by the skewness of the N_{10} distribution, which is defined as

$$S_{N_{10}} = \frac{\mathbb{E}[N_{10} - \mu_{N_{10}}]^3}{\sigma_{N_{10}}^3},$$

where $\mathbb{E}[\,\cdot\,]$ is the expectation operator, and $\mu_{N_{10}}$ and $\sigma_{N_{10}}^2$ are the mean and the variance of the N_{10} distribution, respectively. Larger skewness indicates a stronger emergence of hubs in the data.

We evaluate the performance of the ranking tasks by the highest rank of terms that are the "family members" of a query term. Here, the family members of a term is its parent, children or siblings in the MeSH tree. Because these are the terms that are close to the query in terms of meaning, a sensible similarity measure should rank them higher than other terms in the ranking list for the query term. Hence, for the ranking tasks, smaller this metric, the better. The results are averaged over all terms (queries) in the MeSH tree.

For the multi-class classification we calculate the accuracy predicting the correct category that a test document belongs to (larger the better), and for the multi-label classification we count the number of disagreement between the correct categories and the predicted ones for each test document (smaller the better), and then averaged over test documents as well.

[6] http://mulan.sourceforge.net/index.html

Results

Experimental results are shown in Table 3.1. As we show in section 2.3.3 that the off-diagonal elements of the regularized Laplacian ($\mathbf{L}_{\mathrm{RL}}$) matrix become proportional to those of cosine similarity (Cos) matrix as β approaches to 0, and to those of the commute-time kernels ($\mathbf{L}_{\mathrm{CT}}$) as β tends to infinity. For this reason, we place the results for $\mathbf{L}_{\mathrm{RL}}$ between those of Cos and $\mathbf{L}_{\mathrm{CT}}$ in Table 3.1.

In the ranking task with MeSH categories A–D, $\mathbf{L}_{\mathrm{RL}}$ and $\mathbf{L}_{\mathrm{CT}}$ lower skewness compared to Cos, and simultaneously improve the performance (the 'Rank' row showing the averaged highest rank of family terms). Note that a smaller rank shows a better performance. This trend is observed through all categories A–D.

In multi-class classification tasks with Reuters-52 and TDT2-30 datasets, both datasets show high skewness with cosine similarity (Cos), with an especially high skewness value in Reuters-52.

In contrast, skewness in Reuters-52 decreases as the parameter β of $\mathbf{L}_{\mathrm{RL}}$ is increased, indicating the reduction of hubness. Performance (accuracy) is also improved with the increase of β. For this dataset, the commute-time kernels also outperforms the naive Bayes (NB) classifier. With TDT2-30, however, skewness drops but then goes up as parameter β is increased. The accuracy remains nearly constant, which tells us that Laplacian-based kernels are not effective for this dataset.

In multi-label classification tasks, the Enron dataset shows high skewness with cosine similarity (Cos). The skewness decreases as parameter β increases with the Laplacian-based kernels and for the commute-time kernels. That is, hubs are reduced with the Laplacian-based kernels. As the skewness decreases, the performance (disagreement) improves. For the Bibtex dataset, however, skewness of the Laplacian-based kernels is higher than that of cosine similarity. The performance (disagreement) remains more or less identical to that of Cos, only slightly worse.

In Table 3.1, we also show the results of using commute-time distance in column '$\mathbf{L}_{\mathrm{CT}}$ dist', which is the distance in the feature space of the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$, computed with Eq. (3.4). For all datasets, the extremely high skewness and poor performance of '$\mathbf{L}_{\mathrm{CT}}$ dist' suggest that commute-time *distance* in fact promotes hubs, and is not a good idea to use with *k*nn classification.

Table 3.1: Experimental Results. The rows for MeSH A – D show the results of ranking tasks, Reuters-52 and TDT2-30 show the multi-class classification tasks and Enron and Bibtex correspond to the multi-label classification tasks. Cos, $\mathbf{L}_{\mathrm{RL}}$, $\mathbf{L}_{\mathrm{CT}}$, and NB respectively stands for cosine, regularized Laplacian, the commute-time kernels, and Naive Bayes classifiers. $\mathbf{L}_{\mathrm{CT}}$ dist is the commute-time distance obtained with the application of Eq. (3.4) to the commute-time kernels matrix $\mathbf{L}_{\mathrm{CT}}$. λ_n is the spectral radius of Laplacian $\mathbf{L}$. The number of objects and features for each dataset are shown in the last colomn.

Dataset		Cos	$\mathbf{L}_{\mathrm{RL}}$ ($\beta\lambda_n$)							$\mathbf{L}_{\mathrm{CT}}$	$\mathbf{L}_{\mathrm{CT}}$ dist	NB	# objects (# features)
			(0.01)	(0.1)	(0.5)	(1)	(10)	(100)	(1000)				
MeSH A	Skewness	6.6203	6.5802	6.1549	4.4874	3.3225	1.1931	0.9554	0.9294	**0.9188**	8.9454	-	833
	Rank (smaller=better)	14.7	14.7	14.5	14.2	13.9	**13.4**	13.6	13.7	13.7	172.9	-	(274064)
MeSH B	Skewness	9.9111	9.8021	8.6242	5.7524	3.835	2.1594	1.7736	1.7664	**1.7457**	14.37	-	2098
	Rank (smaller=better)	42.6	42.6	42.4	42.0	41.6	39.4	38.6	38.5	38.5	382.1	-	(228522)
MeSH C	Skewness	7.3111	7.2594	6.5986	4.6353	3.3799	**0.9770**	1.0942	1.2097	1.2154	11.466	-	1347
	Rank (smaller=better)	42.0	42.0	41.8	41.2	40.6	38.7	37.7	**37.4**	**37.4**	284.2	-	(200339)
MeSH D	Skewness	9.0052	8.9104	8.3939	6.3507	4.8183	**1.4867**	1.5200	1.5781	1.5753	13.886	-	1961
	Rank (smaller=better)	119.0	118.9	118.7	117.5	116.4	110.4	106.6	105.9	**105.7**	438.1	-	(212614)
Reuters-52	Skewness	14.815	14.721	14.318	12.722	11.044	**6.1597**	6.7267	6.9076	6.9341	30.1115	-	9100
	Accuracy	0.847	0.847	0.852	0.869	0.872	0.893	0.898	0.893	**0.900**	0.4217	0.865	(19241)
TDT2-30	Skewness	3.6291	3.6145	3.4309	2.8966	**2.5600**	3.3985	4.1023	4.2001	4.2027	30.5958	-	9394
	Accuracy	**0.964**	**0.964**	**0.964**	0.963	**0.964**	0.961	0.958	0.958	0.963	0.0469	0.961	(36771)
Enron	Skewness	6.4651	6.3987	5.7334	3.6943	2.7401	**2.5742**	2.9286	3.0307	3.0375	12.8882	-	1694
	Disagreement (smaller=better)	2.80	2.79	2.70	2.65	2.69	**2.61**	2.68	2.64	2.67	3.30	-	(1001)
Bibtex	Skewness	2.4726	2.4620	**2.4306**	2.6106	2.9568	4.7303	5.6225	5.7226	5.7345	27.1330	-	7395
	Disagreement (smaller=better)	**1.93**	**1.93**	**1.93**	1.94	1.95	1.95	1.97	1.97	1.97	2.37	-	(1836)

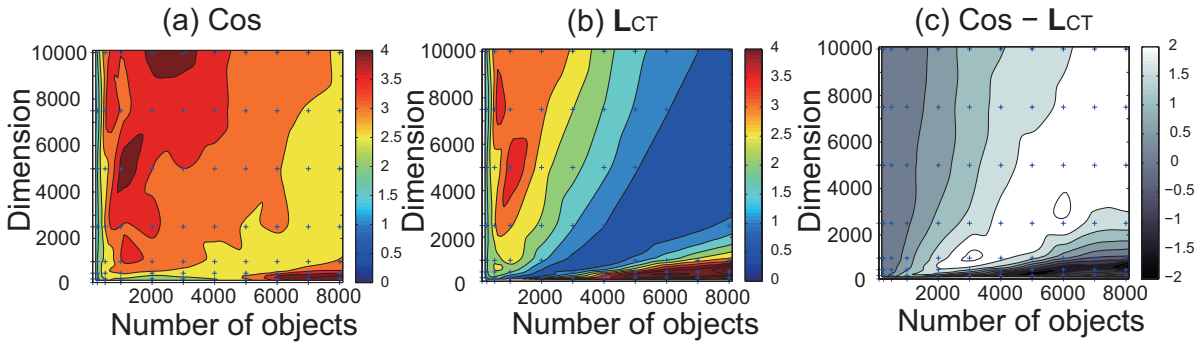


Figure 3.3: Skewness of N_{10} distributions are represented in contour plots, using synthetic sparse vector datasets generated with various number of objects and feature dimensions (+ mark corresponds to a dataset). The panel (a) and (b) show skewness of cosine similarity and the commute-time kernels, respectively, and the panel (c) shows the difference of skewness between cosine similarity and the commute-time kernels.

3.5 Discussion

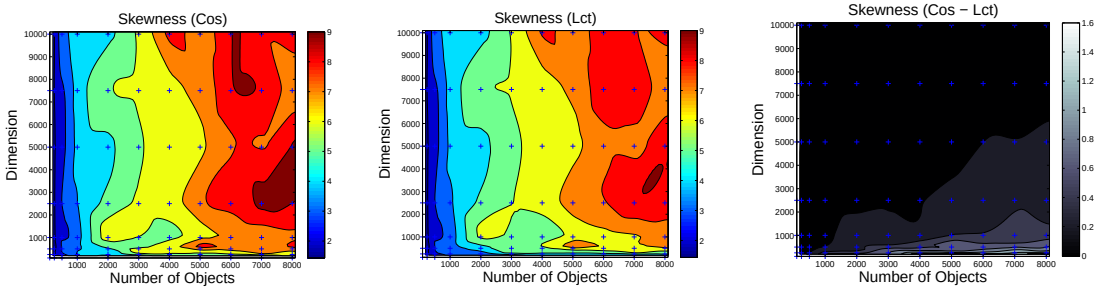
3.5.1 Using skewness for parameter tuning

The experimental results in the previous section showed that, contrary to our expectation, Laplacian-based kernels do not always reduce hubs. Skewness, which is an indicator of hubness, decreased in the MeSH datasets in ranking tasks, Reuters-52 dataset in multi-class classification and Enron dataset in multi-label classification. It did not decrease in Bibtex and TDT2-30 datasets.

However, when the skewness was indeed decreased with Laplacian-based kernels, the task performance was also improved. Moreover, the kernel that gives the smallest skewness value attained the best task performance, or was very close to the best. This result suggests a way to choose kernels, and to automatically tune parameters of the regularized Laplacian, by using the skewness as an indicator of kernel performance.

3.5.2 Hubness phenomenon and dataset size on synthetic data

Let us now discuss hubness from the viewpoint of dataset size (the number of objects), the point not investigated by Radovanović et al., as well as dimensions (the number of features), in simulation studies using synthetic datasets. Here, we generate



(a) Skewness of Cosine Similarity (Cos)

(b) Skewness of the commute-time kernels ($\mathbf{L}_{\mathrm{CT}}$)

(c) Difference of skewness between Cos and $\mathbf{L}_{\mathrm{CT}}$

Figure 3.4: Skewness of N_{10} distributions are represented in contour plots, using synthetic dense vector datasets generated with various number of objects and feature dimensions (+ mark corresponds to a dataset). The panel (a) and (b) show skewness of cosine similarity and the commute-time kernels, respectively, and the panel (c) shows the difference of skewness between cosine similarity and the commute-time kernels.

two types of synthetic data: sparse vector data and dense vector data. We will see that the Laplacian-based kernels reduce hubness in some combinations of dataset size and dimension for sparse data, but do not suppress hubness for dense data at all.

Sparse vector data

We create synthetic datasets of sparse vectors in the same process as we generated Figures 3.1 and 3.2. In these figures, the number of objects n was 500, and the dimension d was 10 or 50. In this section, we vary the number of objects n, between 100 through 8000, and the number of features d between 100 to 10000. Then, $n \times n$ of cosine similarity matrix is calculated with the n objects of d dimensional vectors. We then compute (as the representative of the Laplacian-based kernels) the commute-time kernels matrix, just as we did previously.

Using cosine similarity and the commute-time kernels as similarity measures, we obtain skewness of N_{10} distribution. We use averaged skewness over 10 times repetition of each combination of n and d.

Figure 3.3 shows contour plots of skewness: (a) cosine similarity, (b) the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$ and (c) difference between cosine similarity and the commute-time kernels: (Cos - $\mathbf{L}_{\mathrm{CT}}$). Vertical axis shows the number of objects n and horizontal axis

shows the number of features (dimension) d.

From the Figure 3.3, we observe the following. First, in the case of cosine similarity shown in panel (a) of Figure 3.3, emergence of hubs depends not only feature dimension (that Radovanović reported) but also number of objects. And comparing panel (a) and (b), the commute-time kernels (panel (b)) shows smaller value of skewness in more area (for various number of object and feature dimensions) than that of cosine similarity (panel (a)). Second, from panel (c), by converting cosine similarity matrix into the commute-time kernels, skewness is reduced in various number of objects and dimensions, however, when datasets consist of large number of objects in lower dimensions, such as $n > 5000, d < 1000$, skewness increases and hubs emerge more than when using cosine similarity. We assume this may be related to the increase of skewness with Bibtex dataset in which the number of objects is 7395 and features is 1836.

Dense vector data

We create synthetic datasets consisting of n objects as d dimensional dense vectors, the components of which are generated from the uniform distribution between 0 and 1. Then, in the same way for the experiments using sparse vectors, $n \times n$ cosine similarity matrix is constructed by calculating pairwise cosine similarity between n vectors, and the commute-time kernel is computed from the cosine similarity matrix by using it as an adjacency matrix. For each similarity measure (i. e., the cosine similarity matrix and the commute-time kernel) skewness of N_{10} distribution is calculated. We vary the number of objects n within a range from 100 to 8000, and the number of features d from 100 to 10000. For each combination of n and d, we generate 10 datasets independently, and the averaged skewness over them is plotted in the counter maps as Figure 3.4.

We see from the contour maps skewness stays unchanged between the cosine similarity and the commute-time kernel, in contrast to the experiments using sparse vectors.

Moreover, as for the skewness of cosine similarity, there is a difference seen between the two counter maps for sparse and dense datasets; For dense data, skewness becomes larger as the number of objects increases almost independently of dimensions, while, for sparse data, skewness becomes larger in combinations of a higher dimension and a moderate number of objects.

We would like to make clear the mechanisms behind these findings but we leave them for future work.

Table 3.2: A list of real dataset used for investigating hubness.

Dataset	Task	#classes	#samples	#features
MeSH A	Thesaurus mapping	-	833	274,064
MeSH B	Thesaurus mapping	-	2,098	228,522
MeSH C	Thesaurus mapping	-	1,347	200,339
MeSH D	Thesaurus mapping	-	1,961	212,614
line	WSD (word sense disambiguation)	6	4,146	8,009
interest	WSD (word sense disambiguation)	6	2,368	3,689
Reuters-transcribed	Document classification	10	201	3,863
MovieLens	Collaborative filtering	-	943	1,682
20newsgroup(train)	Document classification	20	11,293	54,580
Reuters(all train sample)	Document classification	52	6,532	16,145
Reuters(earn and acq)	Document classification	2	4,436	11,947
Reuters(earn)	-	1	2,840	7,722

3.5.3 Hubness phenomenon and dataset size on real data

In section 3.5.2, we discussed hubness from the viewpoint of dataset size (the number of objects), with synthetic datasets. In this section, we continue the discussion with real datasets.

We examined skewness on various real datasets. The information of datasets is listed in Table 3.2, and other miscellaneous information is as follows.

- **MeSH** : MeSH datasets consist of biomedical terms stored in the MeSH thesaurus, which we used for thesaurus mapping tasks in section 2.5.
- **WSD** : The two datasets "line" and "interest" are the sets of ambiguous words in different contexts. These are used in the past Senseval workshop for word sense disambiguation, and are obtained from Ted Pedersen's web page (http://www.d.umn.edu/ tpederse/data.html).
- **Reuters-transcribed** : Reuters-transcribed is a document dataset which is annotated with predefined class labels, and is used for a classification task. The dataset is obtained from UCI machine learning repository http://archive.ics.uci.edu/ml/datasets/Reuters+Transcribed+Subset.

- **MovieLens** : MovieLens is a collection of movie ratings. It is a famous benchmark dataset for collaborative filtering, which recommend movies based on the ratings. We use the dataset following Fouss et al. [16].

- **20newsgroup** : 20newsgroup is a collection of newspaper articles annotated with 20 class labels, and it is a widely used benchmark dataset for classification tasks. We use the preprocessed one obtained from http://web.ist.utl.pt/ acardoso/datasets/.

- **Reuters** : Reuters is also a collection of newspaper articles annotated with class labels. We used the preprocessed dataset obtained from http://web.ist.utl.pt/ acardoso/datasets/.

For each dataset, skewness is measured by cosine similarity (Cos), the regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$ and the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$. The results are shown in Table 3.3, Figure 3.5 (MeSH datasets) Figure 3.7 (Reuters) and Figure 3.6 (20newsgroup).

The results show that for most of the datasets hubs are reduced by the Laplacian-based kernels. However, there are cases that hubs are amplified conversely. For example, in case using all samples (11,293) of 20newsgroup dataset, skewness of N_{10} distribution becomes larger as β of the regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$ grows, and becomes the largest when using the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$ (see Table 3.3).

To examine the effect of the number of samples on hubs (skewness of N_k distribution), we randomly select (200, 500, 1000 and 5000) samples from the 20newsgroup dataset and measure skewness using each similarity measures (cosine similarity, the Regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$ and the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$). The skewness is averaged over 10 times repetition for each sample size. The results are shown in Table 3.3 and Figure 3.6. From the results, we see that the number of samples affect the skewness of N_{10} distribution when using the regularized Laplacian with large β and the commute-time kernels. In short, when the number of samples is larger, the skewness of N_{10} distribution becomes larger using the Laplacian-based kernels.

Also, we evaluated the effect of hubs on task performances. The performances are measured for MeSH, 20newsgroup and Reuters datasets. As for the evaluation metrics, we follow the one described in section 3.4.2. The results are shown in Table 3.4.

Remarks on evaluation method

As we described in previous section 2.3.3, as the β of the regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$ approaches to infinity, the off-diagonal elements of $\mathbf{L}_{\mathrm{RL}}$ becomes proportional to those of the commute-time kernels matrix $\mathbf{L}_{\mathrm{CT}}$. Hence the skewness and the performance of the regularized Laplacian $\mathbf{L}_{\mathrm{RL}}$ tend to be those of the commute-time kernels matrix $\mathbf{L}_{\mathrm{CT}}$. However, as shown in Figure 3.6 (a) - (c), the accuracy of the regularized Laplacian with large β drops suddenly without matching that of the commute-time kernels $\mathbf{L}_{\mathrm{CT}}$. This tendency is obvious when k of $k$nn is large (that is $k = 20$ in Figure 3.6 (a) - (c)). This is due to the following reason. Recall that Laplacian-based kernels are written in the form of Eq. (2.5), and the eigenvalue is defined in Eq. (2.7). Combining those equations,

$$\begin{aligned}
\mathbf{L}_{\mathrm{RL}} &= \sum_{i=1}^{n} \frac{1}{1+\lambda_i\beta}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}} \qquad (3.5)\\
&= \frac{1}{1+\lambda_1\beta}\mathbf{u}_1\mathbf{u}_1^{\mathrm{T}} + \frac{1}{1+\lambda_2\beta}\mathbf{u}_2\mathbf{u}_2^{\mathrm{T}} + \cdots + \frac{1}{1+\lambda_n\beta}\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}\\
&= \mathbf{u}_1\mathbf{u}_1^{\mathrm{T}} + \frac{1}{1+\lambda_2\beta}\mathbf{u}_2\mathbf{u}_2^{\mathrm{T}} + \cdots + \frac{1}{1+\lambda_n\beta}\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}\\
&\propto \mathbf{u}_1\mathbf{u}_1^{\mathrm{T}} + \frac{1}{\lambda_2}\mathbf{u}_2\mathbf{u}_2^{\mathrm{T}} + \cdots + \frac{1}{\lambda_n}\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}} \quad (\beta \to \infty).
\end{aligned}$$

When the parameter β approaches to infinity, the eigenvalue $\frac{1}{1+\lambda\beta}$ tends to be $\frac{1}{\lambda}$ which is the eigenvalue of the commute-time kernels described in Eq. (2.6). This also proves the statement "as the β of $\mathbf{L}_{\mathrm{RL}}$ approaches to infinity, the off-diagonal elements of $\mathbf{L}_{\mathrm{RL}}$ becomes proportional to those of $\mathbf{L}_{\mathrm{CT}}$".

Meanwhile, combining Eq. (2.5) and Eq. (2.6), the commute-time kernels is written as

$$\begin{aligned}
\mathbf{L}_{\mathrm{CT}} &= \sum_{i=1}^{n} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}} \qquad (3.6)\\
&= \frac{1}{\lambda_1}\mathbf{u}_1\mathbf{u}_1^{\mathrm{T}} + \frac{1}{\lambda_2}\mathbf{u}_2\mathbf{u}_2^{\mathrm{T}} + \cdots + \frac{1}{\lambda_n}\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}\\
&= 0\cdot\mathbf{u}_1\mathbf{u}_1^{\mathrm{T}} + \frac{1}{\lambda_2}\mathbf{u}_2\mathbf{u}_2^{\mathrm{T}} + \cdots + \frac{1}{\lambda_n}\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}.
\end{aligned}$$

By definition, $\frac{1}{\lambda_1} = 0$ so that the first term $\frac{1}{\lambda_1}\mathbf{u}_1\mathbf{u}_1^{\mathrm{T}}$ becomes 0. As the β of the regularized Laplacian approaches to infinity, the eigenvalues become approximately those

of the commute-time kernels. However the first term in Eq. (3.5) does not disappeared even when β approaches to infinity.

This first term which differentiates Eq. (2.7) from (2.6) makes the inconsistency on the performance (i.e., the accuracy of the regularized Laplacian with large β drops suddenly without matching that of the commute-time kernels) shown in Figure 3.6 (a) – (c) especially when k is large.

We evaluated the task performance of 20newsgroup using the $k$nn classification, and the label of test sample is decided by voting the similarity score of $k$nn training samples to corresponding classes. The class obtained the highest score assigned as estimated class for the test sample.

With the regularized Laplacian, each term in Eq. (2.7) becomes proportional to the terms of the commute-time kernel in Eq. (2.6). However the regularized Laplacian has more value than the commute-time kernels for the first term, because the first term of the commute-time kernel is 0.

When voting the similarity scores calculated by the regularized Laplacian and the commute-time kernels, the class voted by more samples tends to obtain more scores with the Regularized Laplacian than the commute-time kernels.

Table 3.3: N_{10} skewness

Dataset		# sample	Cos	$\mathbf{L}_{\mathrm{CT}}$	$\mathbf{L}_{\mathrm{RL}}$							
	$\lambda\beta$				0.01	0.1	0.5	0.999	10	100	1000	
MeSH	β				0.0001151149	0.001151149	0.005755744	0.01149998	0.1151149	1.151149	11.51149	
A	skewness	833	6.6203	0.9188	6.5802	6.1549	4.4874	3.3214	1.1931	0.9554	0.9293	
	β				0.0001020438	0.001020438	0.005102189	0.01019417	0.1020438	1.020438	10.20438	
B	skewness	2098	9.9111	1.7457	9.8021	8.6242	5.7524	3.8368	2.1594	1.7736	1.7664	
	β				8.566497e-05	0.0008566497	0.004283248	0.00855793	0.08566497	0.8566497	8.566497	
C	skewness	1347	7.3111	1.2154	7.2594	6.5986	4.6353	3.3799	0.97699	1.0942	1.2097	
	β				9.982964e-05	0.0009982964	0.004991482	0.009972982	0.09982964	0.9982964	9.982964	
D	skewness	1961	9.0052	1.5753	8.9104	8.3939	6.3507	4.8206	1.4867	1.52	1.5781	
WSD	β				0.0000	0.0002	0.0008	0.0016	0.0159	0.1594	1.5941	
line	skewness	4146	5.4695	2.3689	5.4312	5.087	4.1873	3.6236	2.1746	2.2929	2.3669	
	β				0.0000	0.0002	0.0012	0.0024	0.0241	0.2415	2.4148	
interest	skewness	2368	2.8673	3.7687	2.8202	2.6958	2.2775	1.9605	3.2033	3.7058	3.7591	
Reuters-transcribed	β				0.0006161754	0.006161754	0.03080877	0.06155592	0.6161754	6.161754	61.61754	
	skewness	201	1.29	0.64197	1.2569	1.2374	1.0888	0.9963	0.74976	0.68579	0.66622	
MovieLens	β				3.6292e-05	3.6292e-04	3.6256e-03	1.8146e-03	0.036292	0.36292	3.6292	100
	skewness	943	2.8083	2.3614	2.8002	2.6490	2.0205	2.2254	2.1334	2.3282	2.3516	2.3620
20newsgroup	β				0.0014	0.0144	0.0718	0.1434	1.4357	14.3568	143.5679	
Random sampling	skewness	200	2.2134	0.2957	2.2103	2.0850	1.6396	1.1870	0.3855	0.2838	0.2862	
	β				0.0006	0.0058	0.0291	0.0581	0.5811	5.8115	58.1147	
Random sampling	skewness	500	2.1451	0.7638	2.1190	1.9564	1.4738	1.1580	0.7612	0.7562	0.7619	
	β				0.0003	0.0028	0.0142	0.0284	0.2848	2.8477	28.4769	
Random sampling	skewness	1000	2.3097	1.1913	2.2943	2.0793	1.5785	1.3259	1.1452	1.1733	1.1912	
	β				0.0001	0.0005	0.0027	0.0053	0.0532	0.5323	5.3232	
Random sampling	skewness	5000	2.9767	2.9189	2.9522	2.7164	2.2319	2.0496	2.2244	2.7649	2.9003	
	β				2.3341e-05	0.00023341	0.001167	0.0023318	0.023341	0.23341	2.3341	
All samples	skewness	11293	3.1062	4.3747	3.0894	2.9227	2.5572	2.4333	3.0604	4.1075	4.3663	
Reuters	β				3.0044e-05	0.00030044	0.0015022	0.0030014	0.030044	0.30044	3.0044	
Train 1 class	skewness	2840	6.3557	3.9722	6.2610	6.1266	5.7104	5.3549	3.8527	3.9037	3.9577	
	β				2.0659e-05	0.00020659	0.0010329	0.0020638	0.020659	0.020659	2.0659	
Train 2 classes	skewness	4436	8.1976	4.9978	8.2183	7.9907	7.3379	6.7446	4.5978	4.5978	4.9456	
	β				1.5888e-05	0.00015888	0.00079442	0.0015873	0.015888	0.15888	1.5888	
Train all	skewness	6532	8.6526	6.1776	8.6596	8.4385	7.8058	7.1696	5.4011	5.9100	6.1663	

3.5.4 Can we make skewness smaller by removing hub objects?

We have discussed in this chapter that the Laplacian-based kernels have potentialities to decrease skewness of N_{10} distribution, or to suppress hubness. However, considering that it takes time to compute the kernels when increasing the number of objects, it seems a good idea simply removing hub objects (i.e., the object whose N_{10} is large) from dataset. Here, we examine whether or not the simple method has an effect on reducing hubness using both synthetic and real datasets.

As for a synthetic dataset, following the way described in section 3.2, we generate 2000 objects as binary sparse vectors in 500 dimensions, and for a real dataset, we use the MeSH C dataset again (1347 objects as sparse feature vectors in 200339 dimensions) that we used in section 3.4.2. We calculate similarities between objects by the cosine of their vectors, and use them to compute skewness of N_{10} distribution.

Table 3.5 shows changes of skewness parallel to the number of objects removed from dataset in descending order of N_{10}, and skewness using a Laplacian-bases kernel (commute-time kernel $\mathbf{L}_{\mathrm{CT}}$) as a reference. We see from the result that the simple method that removes hub objects from dataset does not decrease skewness comparable to the Laplacian-based kernels, even after many objects are removed. In conclusion, removing hub objects cannot suppress hubness more than employing the Laplacian-based kernels.

3.5.5 Commute-time distance

Laplacian-based kernels are sometimes used to compute distance, through the translation of Eq. (3.4). In particular, the distance computed from the commute-time kernels (*commute-time distance*) has a nice interpretation that it is proportional to the expected number of steps a random walk has to take to go from one vertex to another vertex for the first time and then coming back. However, Fouss et al. [16] report that the commute-time distance was less effective in a collaborative filtering task than the inner products given by the commute-time kernels.

Our experimental results agree with their report; using commute-time distance deteriorated the performance in all the experiments we conducted.

Regarding commute-time distance, von Luxburg et al. [47] reported that as the number of objects in the database increases, k-nearest neighbor lists become analogous for all objects. That is, regardless of the query, the same objects appear in k-nearest

neighbor list. This phenomenon is observed in all datasets used in our experiment; i.e., we see strong correlation between the number of objects and the skewness.

3.5.6 Other Similarity Measure Making the Centroid Equally Similar to All Samples

In Section 3.3, we argued that the Laplacian-based kernels evaluates all objects equally similar to the centroid. This is because the Laplacian-based kernels have an eigenvector of all 1's (section 3.3). When a similarity matrix has an eigenvector of all 1's, the similarity matrix provides a similarity measure which makes data objects equally similar to the centroid.

Besides Laplacian-based kernels, there is other way to make a similarity matrix with an eigenvector of all 1's. In this section, we use such a similarity measure which makes data objects equally similar to the centroid, centered cosine similarity.

Centered cosine similarity The cosine similarity between the i-th object and j-th object is the inner product of their feature vectors whose length is normalized to 1. Centered feature vector is obtained by taking the difference of the normalized feature vector and the centroid vector. The centered cosine similarity between i and j-th object is the inner product of their centered feature vectors. We denote cosine similarity matrix as $\mathbf{A}$ and the centered cosine similarity matrix as $\mathbf{A}_{\text{cent}}$.

When the normalized feature vector $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is given, then the centroid vector $\overline{\mathbf{x}}$ is

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_i \mathbf{x}_i. \tag{3.7}$$

The similarity given by the centering cosine similarity between i and j-th object is

$$[\mathbf{A}_{\text{cent}}]_{ij} = \langle \mathbf{x}_i - \overline{\mathbf{x}}, \mathbf{x}_j - \overline{\mathbf{x}} \rangle. \tag{3.8}$$

Table 3.6 shows the results of hubness and classification accuracy with Reuters-52 dataset. The experiment follows the same procedure with the one described in section 3.4.2. The hubness is measured by skewness of N_{10}. The result shows that the centering cosine similarity reduces the skewness and improves the classification accuracy. This result implies that not only the Laplacian-based kernels but also the similarity measures which make data objects equally similar to the centroid have a potential to reduce hubs.

3.6 Conclusion

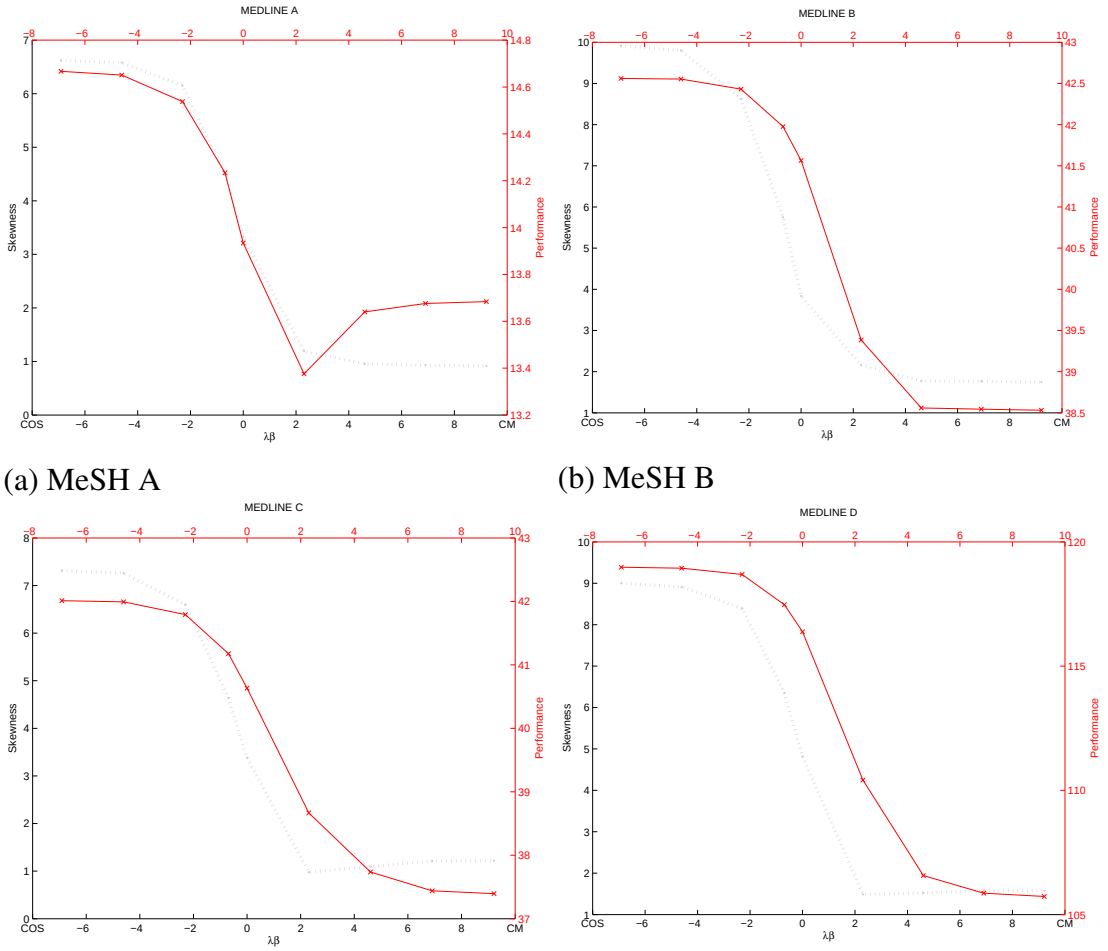
In this chapter, we have investigated whether Laplacian-based kernels such as the commute-time kernels and regularized Laplacian effectively reduce hubs in high-dimensional data. They worked well in ranking and classification tasks (multi-class and multi-label classification), but in some tasks and datasets, they did not lead to performance improvement.

However, whenever these kernels indeed reduced skewness, the kernel that achieves the smallest skewness performed best or close to the best among all the kernels tested. This result suggests that skewness could be used as a yardstick of kernel performance. Note that because skewness can be computed without any label information, its evaluation can be done in an unsupervised manner.

We also found that when Laplacian-based kernels are used, it is almost always better to use the Gram matrix as it is as the similarity matrix, than to translate them into the distance in the feature space, both in terms of skewness of N_{10} distribution as well as the resulting accuracy. A similar experimental result has been reported that commute-time distance were less effective than simply using the components of the commute-time kernels as similarity scores [16]. We suspect that this is also due to the hubness phenomenon.

Table 3.4: Performance

	# samples	kNN	Cos	$\mathbf{L}_{\mathrm{CT}}$	$\mathbf{L}_{\mathrm{RL}}$						
Dataset/$\lambda\,\beta$					0.01	0.1	0.5	0.999	10	100	1000
MeSH A	819		14.667	13.684	14.651	14.538	14.234	13.934	13.376	13.640	13.676
MeSH B	1995		42.561	38.531	42.553	42.431	41.976	41.564	39.384	38.559	38.544
MeSH C	1259		42.011	37.396	41.993	41.792	41.176	40.633	38.667	37.736	37.439
MeSH D	1678		118.98	105.73	118.94	118.69	117.47	116.38	110.41	106.57	105.86
20newsgroup	200	1	0.4965	0.4865	0.4965	0.4985	0.497	0.4935	0.4955	0.4885	0.4865
		2	0.4965	0.4865	0.4965	0.4985	0.4965	0.4935	0.4955	0.4885	0.4865
		3	0.517	0.5065	0.5175	0.519	0.5185	0.5225	0.524	0.5095	0.508
		4	0.535	0.5315	0.5355	0.538	0.5415	0.5455	0.5455	0.546	0.5455
		5	0.5415	0.54	0.5425	0.544	0.547	0.551	0.5495	0.549	0.55
		10	0.5655	0.559	0.5655	0.567	0.574	0.5775	0.573	0.5555	0.5525
		20	0.5705	0.563	0.57	0.5695	0.5675	0.569	0.557	0.532	0.5315
20newsgroup	500	1	0.589	0.5468	0.5892	0.5896	0.5874	0.5882	0.5666	0.5502	0.5478
		2	0.589	0.5468	0.5892	0.5896	0.5874	0.5882	0.5666	0.5502	0.548
		3	0.6136	0.576	0.6134	0.6148	0.6134	0.6158	0.6004	0.584	0.5786
		4	0.637	0.6024	0.6374	0.6384	0.638	0.639	0.6266	0.6094	0.6062
		5	0.645	0.621	0.6452	0.6466	0.6472	0.652	0.6428	0.63	0.6276
		10	0.6724	0.6614	0.6722	0.6732	0.6774	0.6744	0.671	0.661	0.6578
		20	0.683	0.6804	0.6838	0.6846	0.6854	0.6888	0.6928	0.6802	0.6756
20newsgroup	1000	1	0.6586	0.6003	0.6584	0.6587	0.6575	0.6563	0.6362	0.6095	0.6014
		2	0.6586	0.6003	0.6584	0.6587	0.6575	0.6563	0.6361	0.6095	0.6015
		3	0.6816	0.6331	0.6815	0.6819	0.6828	0.6827	0.6656	0.644	0.6373
		4	0.6978	0.6603	0.6981	0.6988	0.7006	0.7012	0.6868	0.6684	0.6626
		5	0.7056	0.6759	0.7056	0.7067	0.7085	0.7093	0.6991	0.6838	0.6787
		10	0.7349	0.7141	0.7352	0.7351	0.7385	0.7402	0.7306	0.7196	0.713
		20	0.7487	0.7397	0.7481	0.7494	0.7532	0.7548	0.7545	0.7397	0.7294
20newsgroup	5000	1	0.81178	0.72144	0.81144	0.81054	0.80878	0.80662	0.77412	0.7349	0.72316
		2	0.8118	0.72156	0.81146	0.81054	0.80874	0.80666	0.7741	0.73486	0.7231
		3	0.8181	0.75632	0.81772	0.8171	0.81682	0.81512	0.79304	0.7655	0.7582
		4	0.82374	0.77348	0.82336	0.82356	0.82274	0.82174	0.8033	0.78058	0.77472
		5	0.82616	0.78246	0.82582	0.82608	0.82544	0.82392	0.809	0.78794	0.7805
		10	0.83274	0.80772	0.83234	0.83264	0.83178	0.83164	0.82348	0.80686	0.79696
		20	0.83538	0.82224	0.83516	0.83562	0.836	0.83554	0.8309	0.8159	0.80328
20newsgroup	11293	1	0.87568	0.77127	0.87506	0.87506	0.87426	0.87258	0.84043	0.7881	0.77313
		2	0.87568	0.77110	0.87497	0.87497	0.87452	0.87249	0.84017	0.78960	0.77313
		3	0.87718	0.80652	0.87656	0.87585	0.87506	0.87532	0.85354	0.82131	0.81094
		4	0.87931	0.82210	0.87931	0.87966	0.87895	0.87789	0.85770	0.83220	0.82316
		5	0.87665	0.83060	0.87647	0.87638	0.87674	0.87630	0.86071	0.83786	0.82742
		10	0.8724	0.84477	0.87178	0.87116	0.87160	0.87098	0.86168	0.84486	0.83388
		20	0.8694	0.85602	0.86895	0.86912	0.87019	0.87036	0.86275	0.84743	0.83220
Reuters 2classes	4436	1	0.86000	0.78787	0.86046	0.86069	0.86159	0.86069	0.82484	0.82484	0.78787
		2	0.86001	0.78787	0.86046	0.86069	0.86159	0.86069	0.82484	0.82484	0.78810
		3	0.87399	0.83070	0.87421	0.87759	0.88458	0.88593	0.87038	0.87038	0.84806
		4	0.88233	0.84152	0.88300	0.88503	0.88841	0.89202	0.88075	0.88075	0.85122
		5	0.89112	0.85685	0.89112	0.89427	0.90126	0.90712	0.89811	0.89811	0.88052
		10	0.90735	0.89923	0.90825	0.91141	0.92493	0.93034	0.92944	0.92944	0.91208
		20	0.91546	0.92628	0.91569	0.91952	0.93372	0.94116	0.94815	0.94815	0.93733
Reuters	6532	1	0.83772	0.74587	0.83665	0.8368	0.83389	0.83022	0.79409	0.75566	0.74709
		2	0.83788	0.74648	0.83680	0.83680	0.83405	0.83068	0.79378	0.75566	0.74724
		3	0.85041	0.79424	0.85012	0.85028	0.85135	0.85257	0.83374	0.80925	0.80588
		4	0.85824	0.81292	0.85808	0.85884	0.86283	0.86314	0.84752	0.82425	0.81782
		5	0.86712	0.82655	0.8665	0.86911	0.87002	0.87232	0.86084	0.84216	0.83742
		10	0.87952	0.86359	0.87998	0.88166	0.8861	0.8884	0.88962	0.87247	0.86926
		20	0.88273	0.88380	0.88319	0.88579	0.89329	0.8974	0.89651	0.88472	0.88181



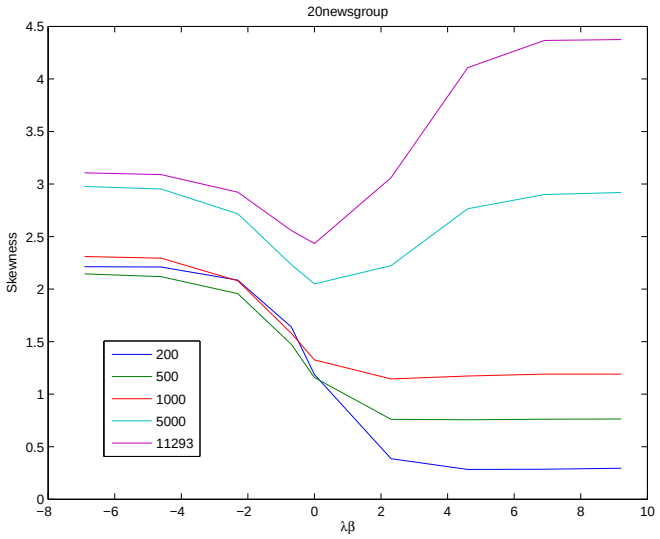
(a) MeSH A

(b) MeSH B

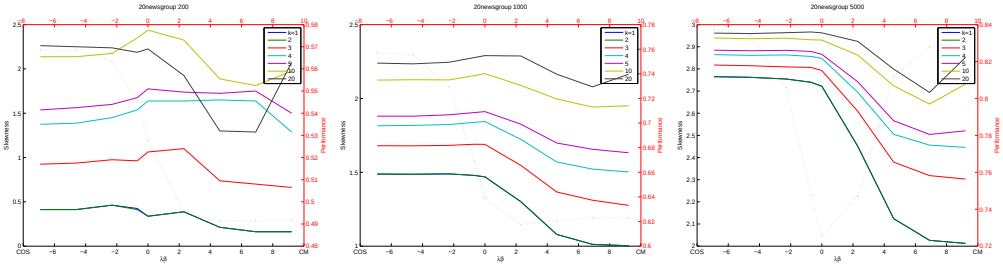
(c) MeSH C

(d) MeSH D

Figure 3.5: Skewness and performance (highest averaged rank (smaller is better)) of MeSH datasets. The black axis corresponds to skewness and the red to performance. The horizontal axis shows the type of similarity (from left to right: cosine similarity, the regularized Laplacian with $\lambda\beta = 10^{-6}$ through 10^{8}, and the commute-time kernels).

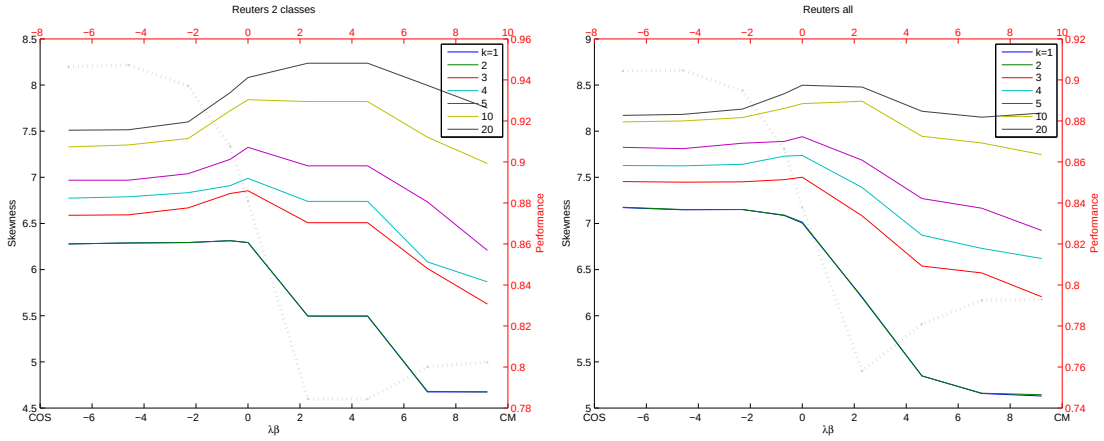


Skewness



(a) 200 samples (b) 1000 samples (c) 5000 samples

Figure 3.6: Skewness of the 20newsgroup dataset with varying sample size (200, 500, 1000, 5000 ranomly chosen samples, and all 11293 samples). Panels (a) - (c) respectively show the performance (classification accuracy) of *k*nn classifiers with various k for 200, 1000, and 5000 sample datasets. We omit 500 and 11293 samples because they are similar to 1000 and 5000 samples. The black axis corresponds to skewness and the red to performance. The horizontal axis shows the type of similarity (from left to right: cosine similarity, the regularized Laplacian with $\lambda\beta = 10^{-6}$ through 10^{8}, and the commute-time kernels).



(a) Reuters 2 classes (b) Reuters all(6532)

Figure 3.7: Skewness and performance results for Reuters datasets. In the panel (a), samples belonging to the most frequent two classes in training data are used. In the panel (b), all training samples are used. The black axis corresponds to skewness and the red to performance. The horizontal axis shows the type of similarity (from left to right: cosine similarity, the regularized Laplacian with $\lambda\beta = 10^{-6}$ through 10^{8}, and the commute-time kernels).

Table 3.5: Changes of skewness parallel to the number of objects removed from dataset in descending order of N_{10}. The last column shows skewness using a Laplacian-bases kernel (commute-time kernel $\mathbf{L}_{\mathrm{CT}}$).

# Removed objects	0	10	20	30	40	50	$\mathbf{L}_{\mathrm{CT}}$
Synthetic data	4.4695	2.1983	1.9667	1.8049	1.6633	1.5587	1.3172
Real data (MeSH C)	7.3111	3.9659	3.4792	3.4117	2.6508	2.6798	1.2154

Table 3.6: The results of hubness and classification accuracy with Reuters-52 dataset by using centering cosine similarity (Cent-Cos). As reference, we also show the results of cosine similarity (Cos) and the commute-time kernels ($\mathbf{L}_{\mathrm{CT}}$).

Similarity measure	Cos	Cent-Cos	$\mathbf{L}_{\mathrm{CT}}$
Skewness	14.815	11.0373	6.9341
Accuracy	84.7%	88.9%	90.0%

Chapter 4

A Robust Model Selection for Classification of Microarrays

4.1 Introduction

Microarray technology [9] has been applied to compare gene expression profiles in cancer tissue samples with different prognoses, and its power to predict cancer prognosis has been demonstrated for several types of cancers [1, 29, 36]. There are, however, two problems in expanding the use of the microarray-based prediction systems in real clinical scenes, namely, observation cost and reliability [14].

In order to reduce the observation cost in clinical scenes, mini-chip microarrays, including hundreds of spots, were developed [36, 30]. Namely, after a predictor is constructed based on a supervised analysis with a full dataset taken by a full microarray system using thousands or tens of thousands of spots, the predictor is implemented with a mini-chip microarray. Note that, for designing an effective mini-chip, the number of genes to be spotted in a chip should be as small as possible because the cost per chip is approximately proportional to the number of spots in the chip. The cost per chip includes the cost of manufacturing the chip, the running cost of observation, and the mass of the clinical specimen used in the chip for each patient. The mass of the clinical specimen required for each patient is proportional to the size of the chip, which can be reduced if the number of spots in the chip is small and if the density of spots in the chip is high. In addition, each gene can be multiply spotted to gain reliable measurements by reducing the cost with small number of genes.

To achieve reliability of the predictor, a well known trade-off problem is that we should select as a large number of informative genes and a small number of non-

informative genes as possible. Namely, putting the observation cost aside, we need to reduce the number of genes. On the other hand, too a small number of genes can lead significant loss of prediction accuracy. Thus, in the supervised analysis process, our goal should be stated that as reliable predictor based on as few genes as possible.

In general, supervised analyses include the following three processes:

- a gene selection process,
- a supervised learning process that constructs a predictor based on a labeled set of expression data of the selected genes, and
- an assessment process of the constructed candidate predictors.

There have been many options proposed for the first two processes, and comparisons of their combinations were made from the viewpoint of prediction errors on test data sets, namely generalization performances [15, 34]. In the present study, we use the following two procedures, as proposed by [39]:

- Weighted voting (WV) classifier [20] with gene selection based on absolute t-score (T-WV)
- Linear kernel support vector machine (SVM) [45] with recursive elimination of genes that had the smallest contribution to current classification performance (R-SVM)[52].

These procedures construct multiple candidate predictors corresponding to various numbers of genes included in the predictors. Since their prediction performances for independent test data sets depend on the number of genes, the assessment process is crucial.

In the assessment process, the prediction performance of each candidate predictor is estimated based on the training data, and good estimation is obtained by reducing the estimation bias and the variance. Since the true performance on independent unknown data in the future is unknown, we should select the best predictor with less biased and smaller variance in the estimated performance. In general, the bias-variance trade-off problem is inherent to all statistical models used for prediction, especially in the classification framework [18, 7]. For prognosis prediction by microarray, several past studies focused on reducing the estimation biases of the prediction error rates in determining the best model [35, 46, 49] because inclusion of biases could lead to over-estimation

of the classification performance of the proposed system. The cross-validation (CV) technique is used widely for predicting true classification error rate in unknown samples that are not included in either the training or the test sample sets. Among the CV methods, the leave-one-out technique (LOO) is often used because of its small bias [35]. These studies, however, did not pay much attention to the variances of estimated classification error rates.

The estimated variances in the assessment process are extremely important for practical applications. Even if a classifier has sufficiently low error rate if it exhibits large variance in prediction, the classifier runs a high risk of having a large actual error rate when it's applied to unknown test samples [8]. The LOO criterion sometimes selects a classifier involving a very small number of genes, or even a single gene. Although the single-gene classifier might fit the "as few genes as possible" criterion, classifiers involving more redundant genes tend to exhibit lower noise and provide stronger evidence with respect to prognosis [30]. Recently, several papers proposed methods which considered the estimated error rate variances [8, 19, 50, 31], and also unsupervised methods [13, 25] which tried to minimize the variance of the model by looking at stability of the signatures without seeing directly the class labels. However, there has been no comparison from the viewpoint of mini-chip design, namely, a reliable predictor based on as few genes as possible.

In the present chapter, we consider both the bias and the variance of performance estimation in order to achieve a reliable predictor. We applied a bootstrap sampling method to estimate the distribution of possible error rates, with bias and variance, and proposed a min-max criterion to obtain a stable classifier. We conducted a simulation study that revealed that the min-max criterion tended to select better candidate predictors than the LOO criterion, especially when the number of samples is small. Then, we compared the two typical supervised analysis procedures, T-WV and R-SVM, and found that the T-WV achieves reliable predictors with a small number of genes, which indicated that the T-WV with min-max criterion was more desirable to our purpose, namely, a reliable predictor with as few genes as possible.

4.2 Methods

4.2.1 Notations

Let $x_i = (x_{i1}, \ldots, x_{iM})$ be a vector of the M-dimensional gene expression profile of the i-th sample, and let y_i be a binary class label $y_i \in \{-1, 1\}$ representing the binary status of the i-th sample, for example, tumor or non-tumor. The numbers of samples in the negative ($y_i = -1$) and positive ($y_i = 1$) classes are denoted as n_n and n_p, respectively. Suppose that we have a dataset $D = \{d_i \mid i = 1, \ldots, N\}$, including N samples, where $d_i = (x_i, y_i)$ is a pair of input (expression) and output (class label) of the i-th sample. By applying a supervised machine learning method to the dataset D, we construct a discriminant function $h(x \mid D)$ such that we predict a label $\hat{y}(x')$ for a new input x' by

$$\hat{y}(x') = \begin{cases} 1 & \text{if} \quad h(x' \mid D) \geq 0 \\ -1 & \text{if} \quad h(x' \mid D) < 0. \end{cases} \tag{4.1}$$

4.2.2 T-WV method

The WV method is a typical example of a supervised machine learning method that employs the top k significant genes. Since the significance of the j-th gene is defined according to the following t-score, the entire procedure is referred to as the T-WV method,

$$t_j = \frac{\bar{x}_{pj} - \bar{x}_{nj}}{\sqrt{1/n_p + 1/n_n}\ S_j}, \tag{4.2}$$

where $\bar{x}_{pj}$ and $\bar{x}_{nj}$ are the average expression levels of the j-th gene within training samples labeled 1 and -1, respectively, and S_j^2 is the pooled within-class variance of the j-th gene,

$$S_j^2 = \frac{\sum_{i:y_i=-1}(x_{ij} - \bar{x}_{nj})^2 + \sum_{i:y_i=1}(x_{ij} - \bar{x}_{pj})^2}{n_n + n_p - 2}. \tag{4.3}$$

The genes are ranked according to the absolute value of $|t_j|$, and the top-ranked k genes are selected as significant genes so that the set of these genes is denoted as C_k. The discriminant function obtained by the T-WV method is then constructed as

$$h_k(x \mid D) = \frac{1}{k} \sum_{j \in C_k} t_j (x_j - \bar{x}_j), \tag{4.4}$$

where $\bar{x}_j \equiv \frac{1}{N}\sum_j^N x_{ij}$ is the average expression level of the j-th gene in the training samples.

In the discriminant function h_k, the difference between the j-th gene expression and its average is weighted by its significance, that is, the t-score. Note that the function h_k depends on the number k of significant genes, and thus we need to set k appropriately.

4.2.3 R-SVM method

The R-SVM is another typical example of a supervised machine learning method, which was developed to select important genes for SVM classification [52]. An R code package is publicly available at http://www.hsph.harvard.edu/bioinfocore/R-SVM.html. The discriminant function of a linear SVM is defined as

$$h_k(x' \mid D) = (w \cdot x') + b = \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x') + b, \tag{4.5}$$

where x' is a new input expression vector and x_i is the i-th sample expression vector in the training dataset. α_i and b are parameters to be determined so that training dataset with different labels are classified with the largest margin. $x \cdot x' = \sum_{j=1}^{M} x_j x'_j$ denotes the inner product. Each element of w, w_j, is defined as

$$w_j = \sum_{i=1}^{n} \alpha_i y_i x_{ij}, \tag{4.6}$$

the absolute value $|w_j|$ of which represents the significance weight of the jth gene in the current discriminant function.

As in the T-WV method, the classification performance of the SVM also depends on gene subset selection. The R-SVM applies a recursive feature elimination (RFE) procedure [21]. In the RFE, less significant genes in the current discriminant function are recursively eliminated, and the next discriminant function is constructed based on the new smaller set of genes. Consequently, a sequence of discriminant functions based on decreasing numbers of genes is constructed. Thus, the prediction performance of each discriminant function h_k depends only on the number k of significant genes, which leads to the same problem as in the T-WV, i.e., setting an appropriate number k. In the following section, we describe a common way to set the number of genes in both the T-WV and R-SVM.

4.2.4 LOO model selection

The above mentioned procedures, T-WV and R-SVM, produce many candidate classifiers, from which we should select the best candidate by an assessment process. Although the true performance of a classifier should be measured by classification accuracy on an unknown dataset in the future, we should estimate performance using the dataset obtained in the assessment process. Note that we refer to each candidate in the assessment process as a *model* because we assess a model that includes all procedures used to construct a classifier rather than directly assessing the classifier. In T-WV and R-SVM, a model corresponds to the number of significant genes included in the model.

The LOO procedure has been widely used to estimate, or predict, the true future performance of a classifier. In LOO, a classifier h is built using each leave-one-out dataset $D^{-i}, i = 1, ..., N$, that is, the i-th sample d_i is selected as a validation sample from the dataset D, and its classification performance is assessed using the validation sample. After the assessments for $d_1, \ldots, d_N$, the LOO error rate of the classifier h, $G_{\mathrm{LOO}}(h \mid D)$, is calculated as the averaged error rate

$$G_{\mathrm{LOO}}(h \mid D) = \frac{1}{N} \sum_{i=1}^{N} I(y_i h(x_i \mid D^{-i}) < 0), \tag{4.7}$$

where $I(R)$ denotes the indicator function that takes a value of one if condition R holds, and is otherwise zero. When we select the number k of significant genes by

$$h_k^{\mathrm{LOO}} = \operatorname*{argmin}_k G_{\mathrm{LOO}}(h_k \mid D), \tag{4.8}$$

this model selection procedure is referred to as the LOO criterion.

4.2.5 Resampling bootstrap method

The error rates used to estimate the LOO procedure are known to be nearly unbiased. [35] compared estimated generalization error rates among different resampling methods and showed that LOO had the smallest bias for a simulation dataset and a real microarray dataset. However, LOO has a tendency to include large variance, despite its small biasness [28], because classifiers constructed based on the leave-one-out datasets, D^{-i}, are quite similar to each other. The large variance of the error rate estimation leads to a high risk of selecting a classifier of which the 'true' performance is poor for unknown samples, and the risk becomes higher as the number of candidates

becomes larger. When we assess the performance of many candidate classifiers with large variances, it frequently happens that some of the candidates have remarkably low error rates, even if their true performance is not so good. This is the same problem as that seen in overfitting, which was originally found in parametric learning by applying too many parameters. Therefore, it is important to reduce the estimation variance in order to obtain a robust classifier.

We applied a bootstrap method to obtain a distribution of LOO error rates that simulated the possible variation of the dataset. We generated bootstrap datasets $\{D^{*b} \mid b = 1,\ldots,B\}$, in which each bootstrap dataset is defined as

$$D^{*b} = \{d_r^{*b} = (x_r^{*b}, y_r^{*b}) \mid r = 1,\ldots,N-1\}, \tag{4.9}$$

where d_r^{*b} is randomly sampled with replacements from the LOO dataset D^{-i}. The single validation sample d_i is evaluated by classifiers that were trained by different datasets D^{*b}, which lead to a set of LOO error rates: $G_{\mathrm{LOO}}(h_k^{*1} \mid D^{*1}), G_{\mathrm{LOO}}(h_k^{*2} \mid D^{*2}),\ldots,G_{\mathrm{LOO}}(h_k^{*B} \mid D^{*B})$, where $h_k^{*b}, b = 1,\ldots,B$, is given by Eq. (4.4) after replacing the dataset D with the bootstrap dataset D^{*b}. This set of LOO error rates is considered to be a distribution of G_{LOO} and provides a guideline to determine the number of genes to be used in the T-WV classifier.

4.2.6 Min-max model selection

Using the simulated distribution of LOO error rates, $\{G_{\mathrm{LOO}}(h_k^{*b} \mid D^{*b})\}_{b=1}^{B}$, we defined a risk score, called a min-max criterion,

$$G_{\mathrm{BOOT}}(h_k \mid D) = \mathrm{Per95}\left(\left\{G_{\mathrm{LOO}}\left(h_k^{*b} \mid D^{*b}\right)\right\}_{b=1}^{B}\right), \tag{4.10}$$

where 'Per95' is the 95th percentile of the set of values. Based on this risk score, an appropriate model (i.e., the number of genes, k) is selected as

$$h_k^{\mathrm{BOOT}} = \underset{k}{\mathrm{argmin}}\{G_{\mathrm{BOOT}}(h_k \mid D)\}. \tag{4.11}$$

We considered the 95th percentile with the number of bootstrap $B = 100$ as a representative of the highest error rates that would be possible with each model, i.e., the number of genes. We did not adopt the standard variation of the error rates because the distribution has an asymmetric nature and we were interested in the risk of selecting a worse model.

In the present study, this model is referred to as the "min-max" selection criterion because we minimized the risk of selecting a model for which the expected prediction error rate was the maximum in the distribution of possibilities. This min-max model selection likely refuses classifiers for which the estimated error rates are distributed with a large variance, even if LOO shows the lowest error rate from a single dataset. Therefore, the min-max criterion reduces the instability stemming from the variation of possible datasets that could be obtained by random sampling from a large pool of samples.

In other words, the min-max criterion assumes an underlying game between an analyzer and nature. A dataset is given by nature, and a model is selected by an analyzer. In order for the analyzer to achieve stability, one good idea is to minimize risk (Eq. (4.11)), which stems from the possibility that nature could provide a bad situation (and hence the classifier has been overtrained) (Eq. (4.10)).

The number 95 of the percentile and number of bootstrap $B = 100$ were determined arbitrarily, but there were following reasons. In determining these numbers, there are tradeoffs among computation time, estimation variance of the percentile point and preciseness as a representative of highest error rate:

- The computation time costs proportionally to the number of bootstrap.
- Estimation variance is a monotonic function of both the percentile number and the number of bootstrap. Namely, the variance becomes larger if the percentile number is further from 50 and if the number of bootstrap is smaller.
- The criterion should be a representative of possible highest error rates in an arbitrary asymmetric shape of distribution of bootstrap samples.

We did not select the 50th percentile because of the last reason above, namely, we attempt to obtain a safe classifier rather than those showing good performance on average. Although the 99th percentile is a precise representative of highest error rates, we did not select it because it relied on 1% of bootstrap samples, and will therefore lead to high variance with small B. The estimation variance of the percentiles of bootstrap error rate can be calculated as a standard deviation of order statistics if original distribution is known. In Table 4.1, the standard deviations (SDs) of percentile estimation with assuming a standard normal distribution as an original distribution. Note that these SDs are proportional to the SD of original distribution, and that, although the original distribution should not be normal in reality, the above represents well the scale of the SDs.

Table 4.1: Estimated standard deviations of bootstrap percentiles. Bold type is the setting which we selected.

	$B = 100$	$B = 500$	$B = 1000$
99th	0.315	0.171	0.120
95th	**0.216**	0.095	0.067
90th	0.172	0.077	0.054
50th	0.125	0.056	0.040

4.3 Results

4.3.1 Results for real datasets

We demonstrated our method using four published real gene expression profile datasets:

- Breast cancer
  [44] investigated gene expression microarray data of approximately 5,000 genes for 78+19 breast cancer tissue samples. The samples were classified into favorable and unfavorable samples; namely, recurrence free survival in five years and recurrence in five years were observed, respectively. They trained supervised classifiers by using 78 samples (34 favorable and 44 unfavorable samples), which we call training samples, and tested by using 19 independent samples (7 favorable and 12 unfavorable samples), which we call test samples. The same research group also provided a larger data set which consists of 295 samples [43]. Among the 295 samples, 32 samples was appeared in training dataset [44] and 10 samples were censored in five years, and hence, we used 253 (192 favorable and 61 unfavorable) samples for another test dataset.

- Colon cancer
  The colon cancer dataset [2] contains microarray expression data of 2,000 genes for 62 colon tissues. Among the 62 tissue samples, 40 and 22 samples were labeled as "tumor" and "normal," respectively.

- Neuroblastoma (NBL)
  The NBL dataset [36] consists of the microarray expression data of 5,180 genes

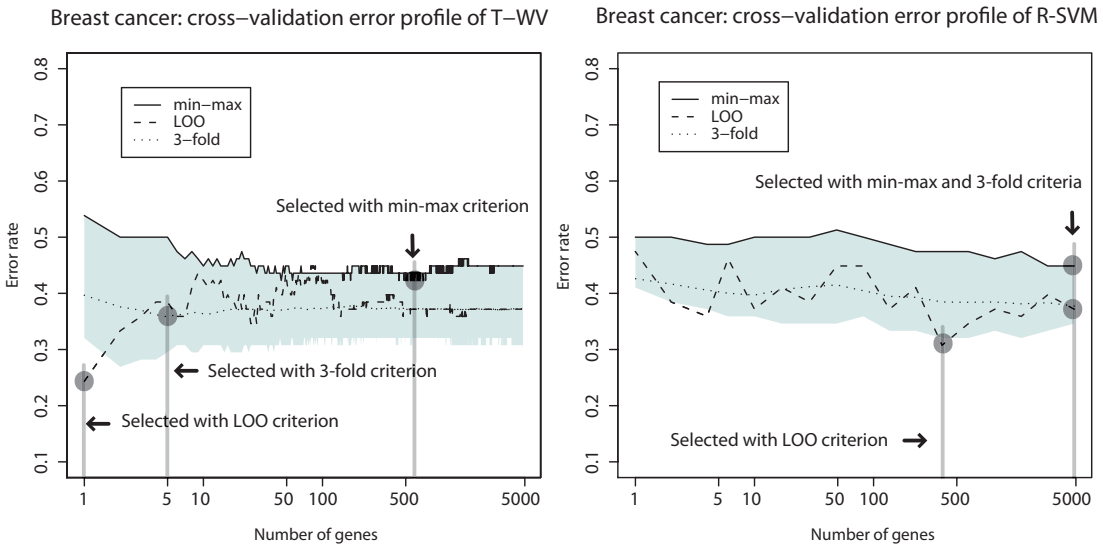


Figure 4.1: Assessed result on breast cancer dataset. The left and right panels show the results based on the T-WV and R-SVM methods, respectively. The estimated 90% interval of classifier performance (gray area), min-max criterion (solid line on the top of gray area), LOO error rate (dashed line), 3-fold-CV error rate (dotted line) are plotted with respect to different numbers of genes. Vertical lines indicate the numbers of genes selected by LOO, the min-max and 3-fold-CV criteria.

for 136 patients (samples). Among the 136 samples, 25 and 102 samples were labeled as "favorable" and "unfavorable" patients, respectively, according to their status at 24 months after diagnosis. The remaining nine samples of unknown status at 24 months after diagnosis were omitted.

- Breast cancer Affymetrix (Affymetrix)
  [48] analysed 286 breast cancer patients with Affymetrix chip harboring 22283 genes. Among the 286 patients, 183 and 93 samples were called favorable and unfavorable, respectively, and 10 samples were censored in five years. Although the Affymetrix dataset is based on breast cancer, we did not consider relationship between the Affymetrix dataset and the breast cancer dataset because they have fairly different natures in distribution, which is not a scope of the chapter.

For each of the above four datasets, we trained T-WV and R-SVM classifiers with various numbers of genes by using training samples, and assessed the classifiers by LOO, 3-, 5-, 10-fold-CV, and min-max criteria. In the case of breast cancer dataset

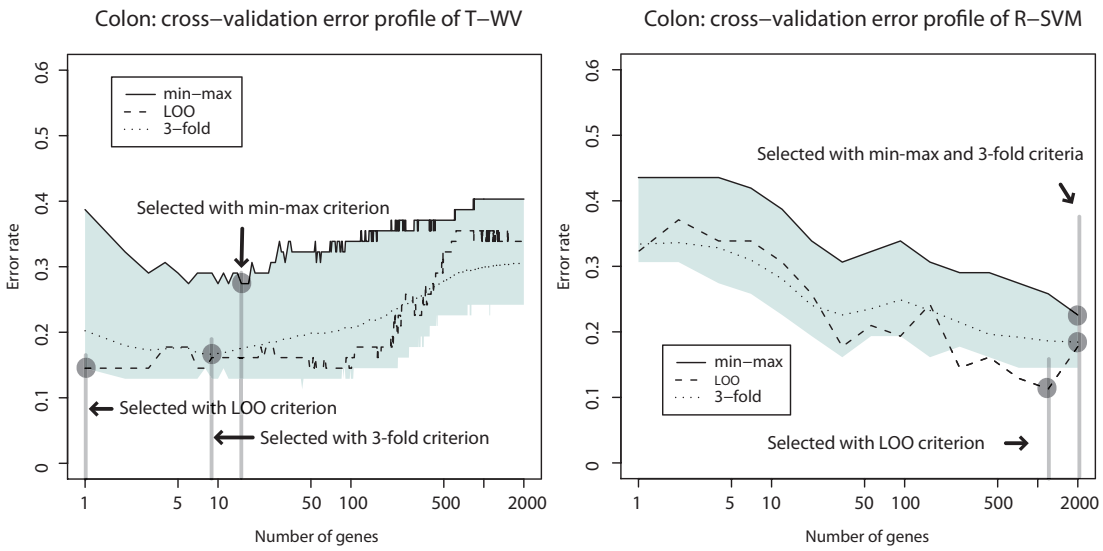


Figure 4.2: Assessed result on colon cancer dataset. See figure 4.1 caption for legend.

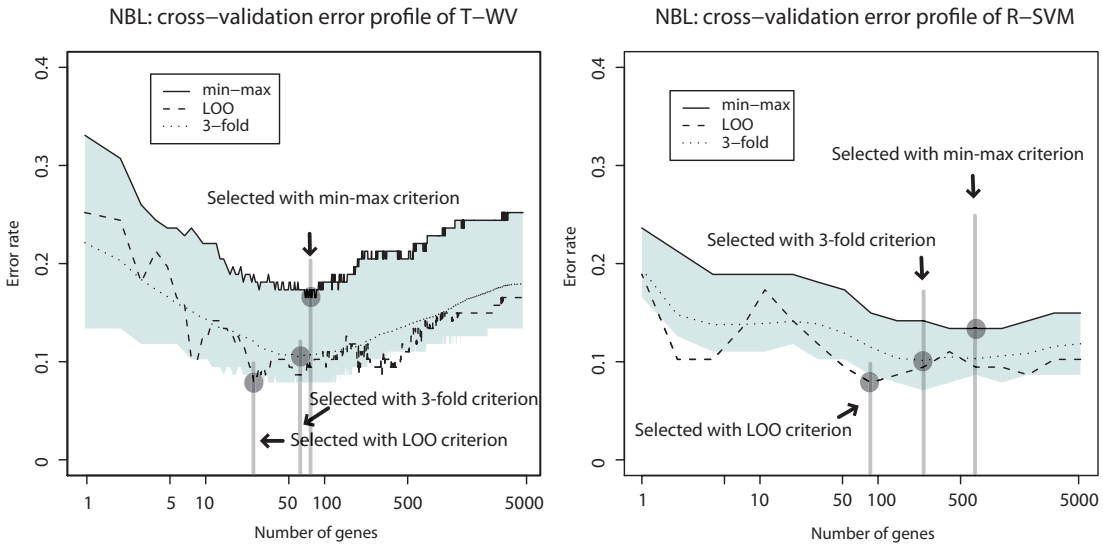


Figure 4.3: Assessed results for NBL dataset. See figure 4.1 caption for legend.

Table 4.2: Test error rate of breast cancer with LOO, min-max and k-fold CV assessed by 19 and 253 test samples.

	T-WV			R-SVM		
	Selected number of genes	Test error rate of 19 samples	Test error rate of 253 samples	Selected number of genes	Test error rate of 19 samples	Test error rate of 253 samples
LOO	1	0.2105	0.4862	376	0.4737	0.4664
min-max	590	0.1578	0.2925	4,833	0.4211	0.3992
3-fold	5	0.3158	0.3992	4,833	0.4211	0.3992
5-fold	2	0.2632	0.4071	626	0.6316	0.5217
10-fold	1	0.2105	0.4862	376	0.4737	0.4664

with large numbers of test samples [44, 43], we also assessed the classifiers in the test samples.

Figure 4.1 shows the result of the breast cancer dataset. In the left panel, the estimated error rates of the T-WV classifier are shown for different numbers of genes, k; where the error rates are estimated by LOO (dash line) and 3-fold-CV (dot line), and the 90% interval of LOO error rate (gray area) with 95th percentile, which represents the risk score G_{BOOT} (solid line at the top of the 90% interval), are estimated by re-sampling bootstrap method. The LOO error rate profile reached the lowest value with a small number of genes, namely $k = 1$, and hence $k = 1$ is selected as the best number of genes with the LOO criterion. On the other hand, the 90% interval of the bootstrap distribution at $k = 1$ exhibits a large variance in the error rate, and the 95th percentile error rate is over the chance level 0.5, which indicates large risk of falling into a poor predictor under the chance level. Also, the LOO error rate at $k = 1$ lies under the 5th percentile and the 3-fold-CV error rate, which reveals that the lowest LOO error rate at $k = 1$ is likely achieved by chance. The min-max criterion, i.e. the 95th percentile, selected a larger number of genes, $k = 590$. Although the LOO error rate and the 90% interval at $k = 590$ showed higher error rate than LOO at $k = 1$, the classifier of $k = 590$ has lower risk to take a poor predictor than that of $k = 1$. 3-fold-CV selects a classifier with $k = 5$ which has less variance than LOO in the 90 % interval.

In the right panel of Figure 4.1, a similar comparison is shown between LOO, 3-fold-CV, and min-max criteria with the R-SVM classifier. The LOO criterion showed an instability which is similar to that of T-WV, and the lowest LOO error rate at $k = 376$ seemed to be achieved by chance. All criteria selected larger numbers of genes than

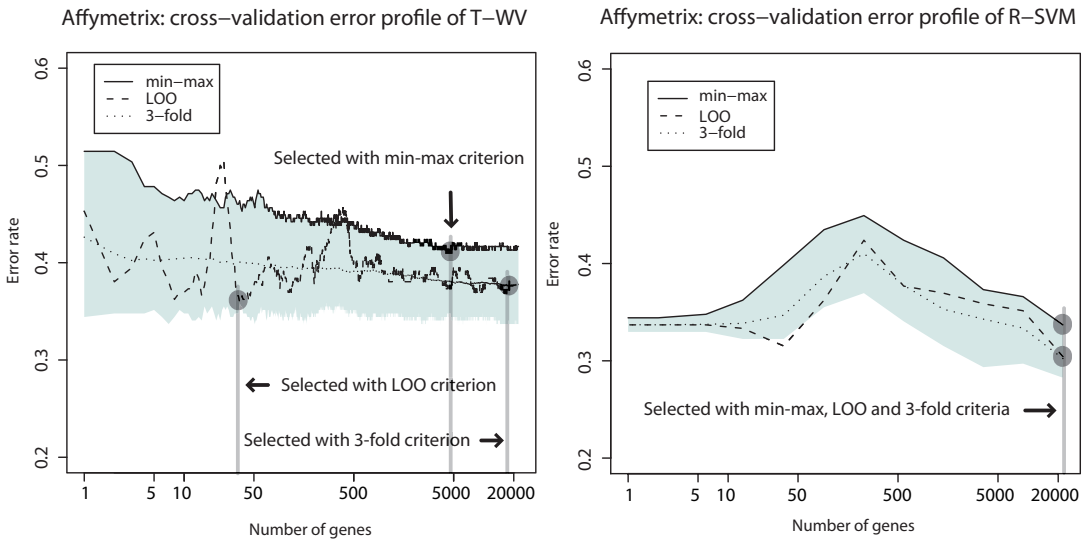


Figure 4.4: Assessed result on breast cancer affymetrix dataset. See figure 4.1 caption for legend.

the T-WV classifier.

In Table 4.2, test error rates of the selected predictors are assessed by using two test datasets with 19 and 253 samples, where five criteria (LOO, min-max, 3-, 5-, and 10-fold-CV) with two classifiers (T-WV and R-SVM) are compared. While min-max outperformed the other criteria, LOO and k-fold-CV, about both 19 and 253 test samples, LOO exhibits poor performance with 19 test samples and worse with 253 test samples whose test error rate is nearly the chance level. Intuitively, this result points out a defect of LOO against the risk of takeing a poor classifier, which is already predicted by the 90% interval in Figure 1. By considering variance, 3-fold-CV achieved better performance in test error rates than LOO, but less than min-max. T-WV tended to exhibit smaller error rate than R-SVM with smaller numbers of genes, although we cannot insist supremacy of the T-WV based on this single example.

Test error rates on 253 samples seemed significantly worse than the error rates on 19 samples and several reasons were considered:

- The 19 samples might include many those samples which were easy to be classified.
- The number of samples 19 was too small to reproduce the error rate in low variance,

- The test data of 253 samples were gathered from different populations from those for the 78 training data and 19 test data.

- The microarray measurement system was somewhat different between those used for observing test data of 253 samples and the 78 and 19 samples.

We should note that the above reasons are the cases which we should consider in order to design mini-chip based on training datasets. The last reason, different microarray system, was not likely the case with this breast cancer dataset, however, it will likely be the case with mini-chip which should be designed based on the other full-size chips.

We compared three criteria, LOO, min-max, and 3-fold-CV, with two classifiers T-WV and R-SVM on the other three datasets: colon cancer, neuroblastoma (NBL), and breast cancer Affymetrix in Figures 4.2, 4.3, and 4.4, respectively. From the comparison between the Figures 4.2, 4.3, and 4.4, we observe the following tendencies:

- Although the error rates estimated by LOO fluctuate as the number of genes increases, they are mostly kept within the 90% interval. This suggests that the LOO estimation for each number of genes originally includes a large variance and the variance is captured by the estimated 90% interval.

- In contrast to the fluctuating LOO error rate profile with respect to the numbers of genes, the profile of the 95th percentile (G_{BOOT}) exhibites a smoother curve. This suggests a more stable nature of the min-max criterion than the LOO criterion. The k-fold-CV shows the characteristics of median value, and hence has smoother effects, however, it is not enough to assess the risk of classifier since the error rate distributions show asymmetric nature.

- The 90% confidence intervals tends to be wider when the numbers of genes are smaller, $k < 5$, which indicates that discriminant models based on too few genes is risky, therefore, when we apply the LOO criterion, we occasionally take high risk to consider a model with a small number of genes. On the other hand, the 95th percentile tends to indicate higher error rate for a smaller numbers of genes $k < 5$ than for a larger numbers of genes $k > 5$. Thus, the min-max criterion based on the 95th percentile avoids risky models with very small numbers of genes, and so min-max is expected to achieve better models with lower error rate.

- For datasets with larger sample size, the 90% interval tends to be narrower. In the neuroblastoma (NBL) dataset, the number of samples is almost twice as large as that in the colon cancer dataset, which leads to a narrower confidence interval.
- In R-SVM as well as T-WV, LOO shows a fluctuating curve and min-max is a smoother curve with minimized risk. Hence, the min-max criterion is expected to be a better model selection than by the LOO criterion for R-SVM.
- The best performances of R-SVM are similar to those of T-WV with larger numbers of genes than the T-WV. Thus, T-WV which involves a smaller number of genes is suitable for practical clinical applications, which is consistent with a previous finding[39].
- The confidence intervals for R-SVM tends to be narrower than those for T-WV, which indicates that the SVM achieved a large margin classifier that is more stable with respect to observation noise within the margin, as compared to that for T-WV.

Even though we are not interested in classifiers with a large number of genes, say $k > 1000$, this finding is important for some applications other than mini-chip construction. Note that R-SVM often predicted the labels with unique answer (e.g. 1 for all samples) in small number of genes which led to narrower confidence interval. In this case, the narrower confidence interval doesn't assure stable predictions with the small number of genes.

4.3.2 Simulation study on synthetic dataset

In the previous section, we tested our procedure on some real datasets, however, true models were unknown and the numbers of samples were limited except for breast cancer dataset, which prevented strong evidence for the predominance of the proposed criterion from being obtained. To assess the performances of the proposed min-max criterion, we conducted a simulation study based on a sufficient number of artificial test samples, which were difficult to achieve in real cases.

We randomly generated expression profiles of 2,000 genes, where 30 out of the 2,000 genes were differentially expressed (DE) between two classes of samples and the others were not differentially expressed (non DE). For non DE genes, expression levels were generated from a normal distribution with mean zero, $N(0,1)$, and for DE

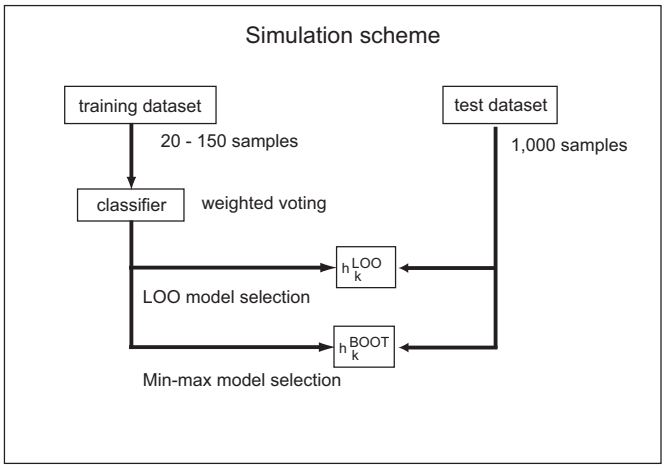


Figure 4.5: Setting of the simulation experiment.

genes, the expression levels of samples with positive and negative class labels were generated from $N(\mu, 1)$ and $N(-\mu, 1)$, respectively, where we set $\mu = 0.5$ for all DE genes. According to the above process, we generated synthetic datasets of 20 to 150 samples for training, and 1,000 samples for testing, where the numbers of samples with the two class labels were set to be equal.

The proposed simulation scheme is illustrated in Figure 4.5. For each training dataset, the candidate classifiers involving various numbers of genes were trained and assessed, and the best numbers of genes were selected by the LOO and the min-max criteria, where the number B of the bootstrap in the min-max procedure was set at 100. The performance of the finally selected classifier was then assessed by a test dataset with 1,000 samples. We repeated this process with a randomly generated training dataset and assessed the corresponding test error rates by using a test dataset of 1,000 samples. The distributions of the test error rates were compared between different conditions.

We designed the above setting in order to clarify how the min-max criterion improves the model selection. The number of test datasets was set sufficiently large and is commonly used for various settings of the other features in order to reduce the variance of error rates that stemmed from random sampling of the test dataset. The number of DE genes, 30, and the strength of differential expression, $\mu = 0.5$, were determined in order to examine typical situations that arose in realistic cases. We omitted other

features of datasets that may arise in realistic cases, for example, variations in the number of DE genes, strength μ, and the proportion of numbers of positive and negative samples, because they had no significant effect in our preliminary experiments. We also omitted correlations of gene expression patterns between DE genes because such correlations would not affect either T-WV or R-SVM.

Table 4.3: Test error rate of simulation dataset

Number of training samples	Selection criterion	Mean	Standard deviation
20	LOO	0.241	0.077
	min-max	0.210	0.064
50	LOO	0.042	0.024
	min-max	0.026	0.012
100	LOO	0.015	0.013
	min-max	0.006	0.003
150	LOO	0.012	0.010
	min-max	0.004	0.002

Figure 4.6 shows the distributions of test error rates of the T-WV classifiers selected by two criteria, LOO and min-max, with 20, 50, 100 and 150 training samples. There are certain levels of variance for both criteria, and the variance is larger for smaller numbers of samples. Frequently, either LOO or the min-max outperforms the other. However, LOO sometimes shows much worse results than min-max, as indicated by the cloud of points in the bottom-right area of each panel in Figure 4.6. Note that the number of test samples, 1,000, is large enough so that there is no significant increase in sampling variance. Table 4.3 shows the means and standard deviations of test error rates of the classifiers selected by LOO and min-max. Through 20 - 150 training samples, min-max outperforms LOO in terms of smaller means and standard deviations of test error rates.

We showed the intersection sets of the genes selected in the model and the real DE genes with respect to the number of genes in the model in Figure 4.8-4.11. In the Figure 4.8 to 4.11, the top panels of each figure show test error rate plotted against the number of genes selected in LOO (left) and min-max (right). The bottom panels shows the number of DE genes included in the selected number of genes with two selection criteria, LOO (left) and min-max (right). It showed that min-max included

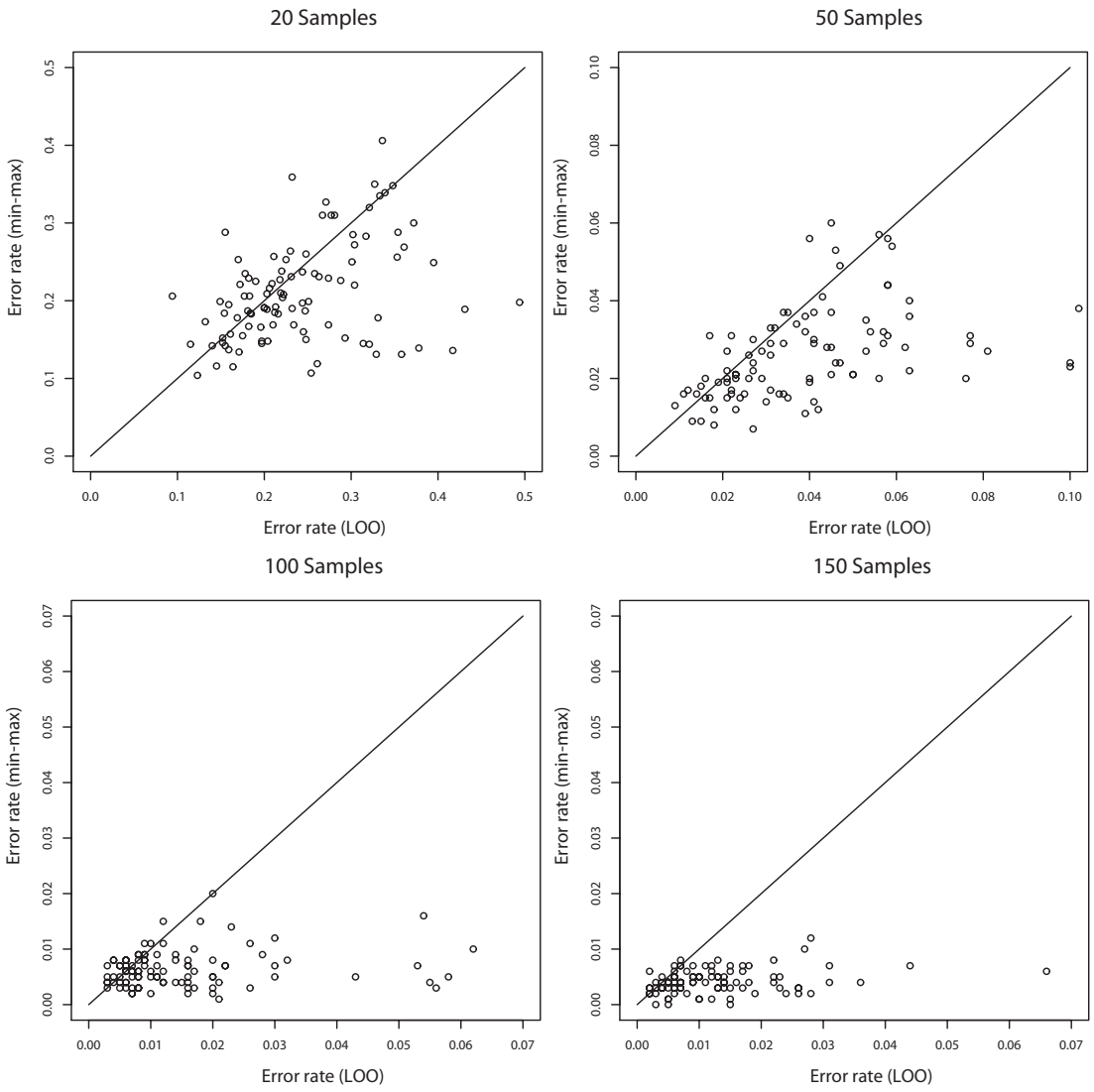


Figure 4.6: Distribution of test error rates of T-WV. The vertical and horizontal axes denote the test error rates of classifiers selected by the min-max and LOO criteria, respectively. The results from 100 trials of random sampling of 20, 50, 100 and 150 samples are shown in the four panels.

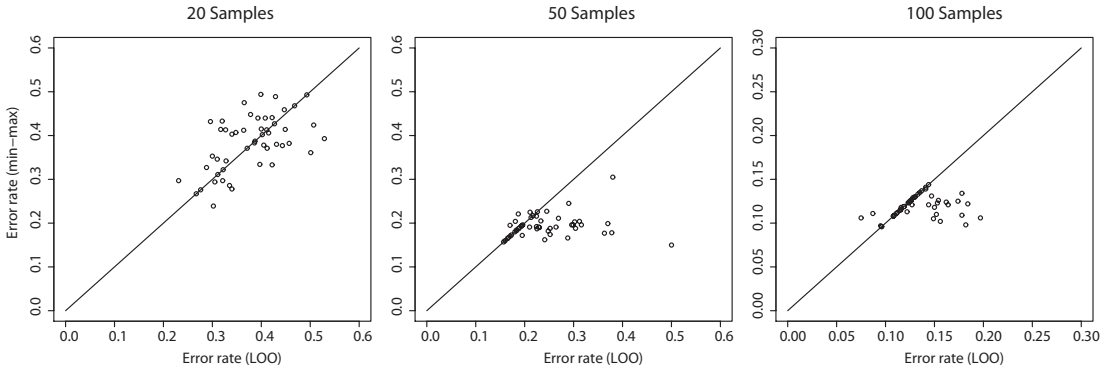


Figure 4.7: Distribution of test error rates of R-SVM. The vertical and horizontal axes denote the test error rates of classifiers selected by the min-max and LOO criteria, respectively. The results from 50 trials of random sampling of 20, 50, and 100 samples are shown in the three panels.

more DE genes with 50 to 150 training samples. While min-max avoided to select the number of genes which showed good error rate by chance, LOO tends to select very few number of genes or large number of genes fortuitously. As the number of training samples increased, the means and the variance of test error rate become smaller and the number of DE genes included in the selected number of genes approach to the true number of DE genes, 30. Even the number of training samples increase, however, test error rate of LOO showed larger variance than min-max caused by fortuitous number of gene selection.

Figure 4.7 showes the distributions of test error rates of R-SVM classifiers selected by LOO and min-max with 20, 50, and 100 training samples. We ploted the results of 50 trials, which is a half of the number of trials, 100, for T-WV, and we did not calculated the case of 150 samples because of computational costs of boostrap simulation for R-SVM. Similar tendency to that of T-WV is observed in the cases of 50 and 100 samples. In the case of 20 samples, label prediction error rates are near the chance level 0.5 and there was no difference shown between the LOO and min-max criteria.

4.4 Concluding remarks

In the present study, we investigated gene subset selection methods in order to design a low cost mini-chip microarray for diagnosis of cancers by gene expression profiles,

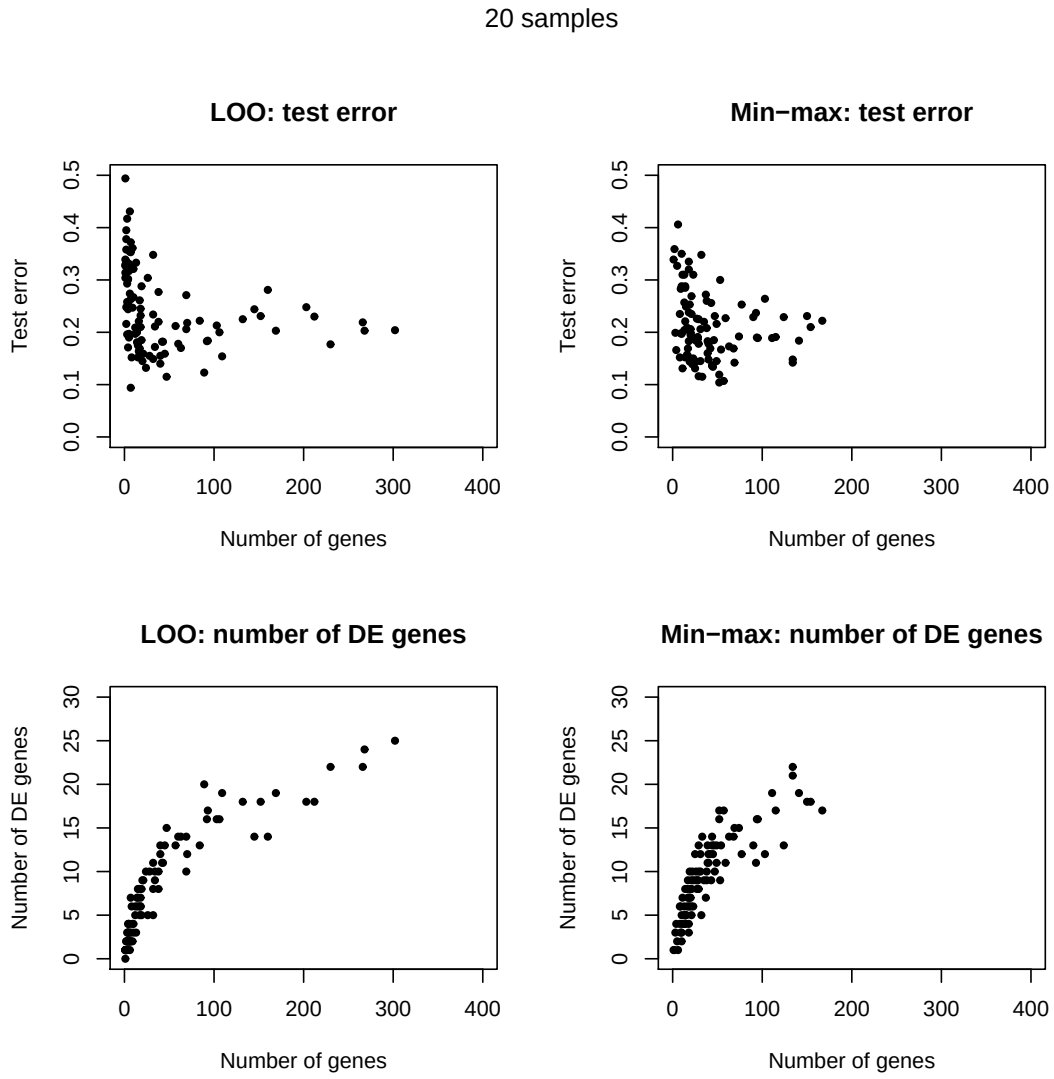


Figure 4.8: 20 samples. The vertical axes for all figures show the number of genes selected in a model. The horizontal axes for the top two figures denote test error of selected model and for the bottom two figures denote the number of DE genes included in a model.

50 samples

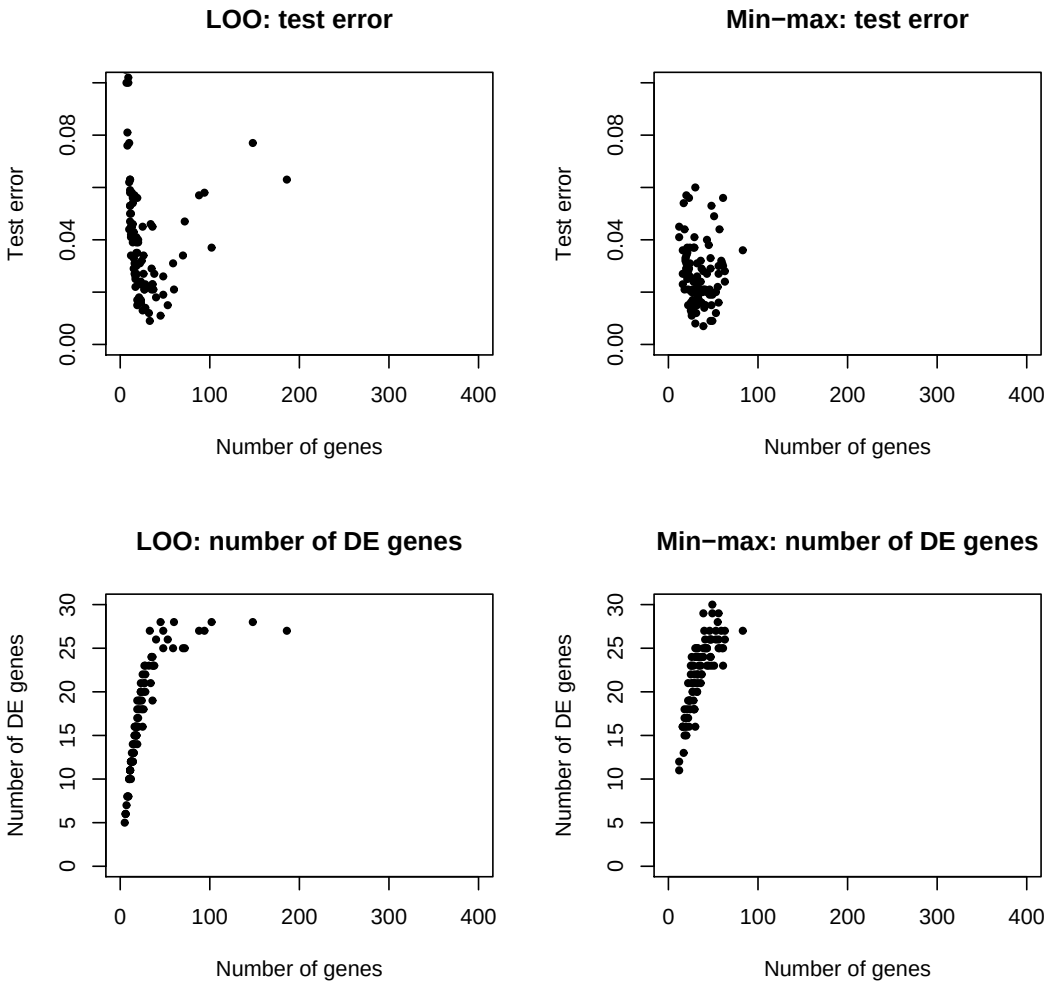


Figure 4.9: 50 samples

100 samples

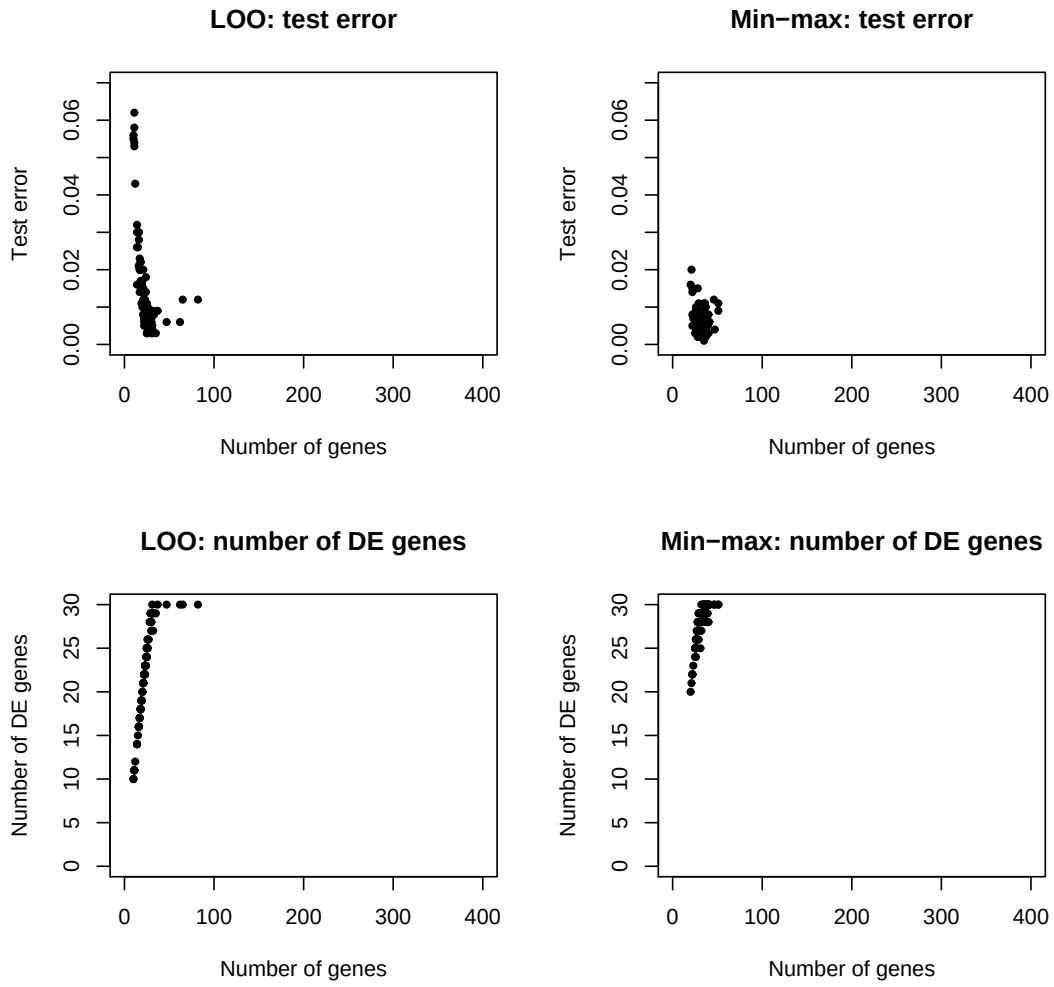


Figure 4.10: 100 samples

150 samples

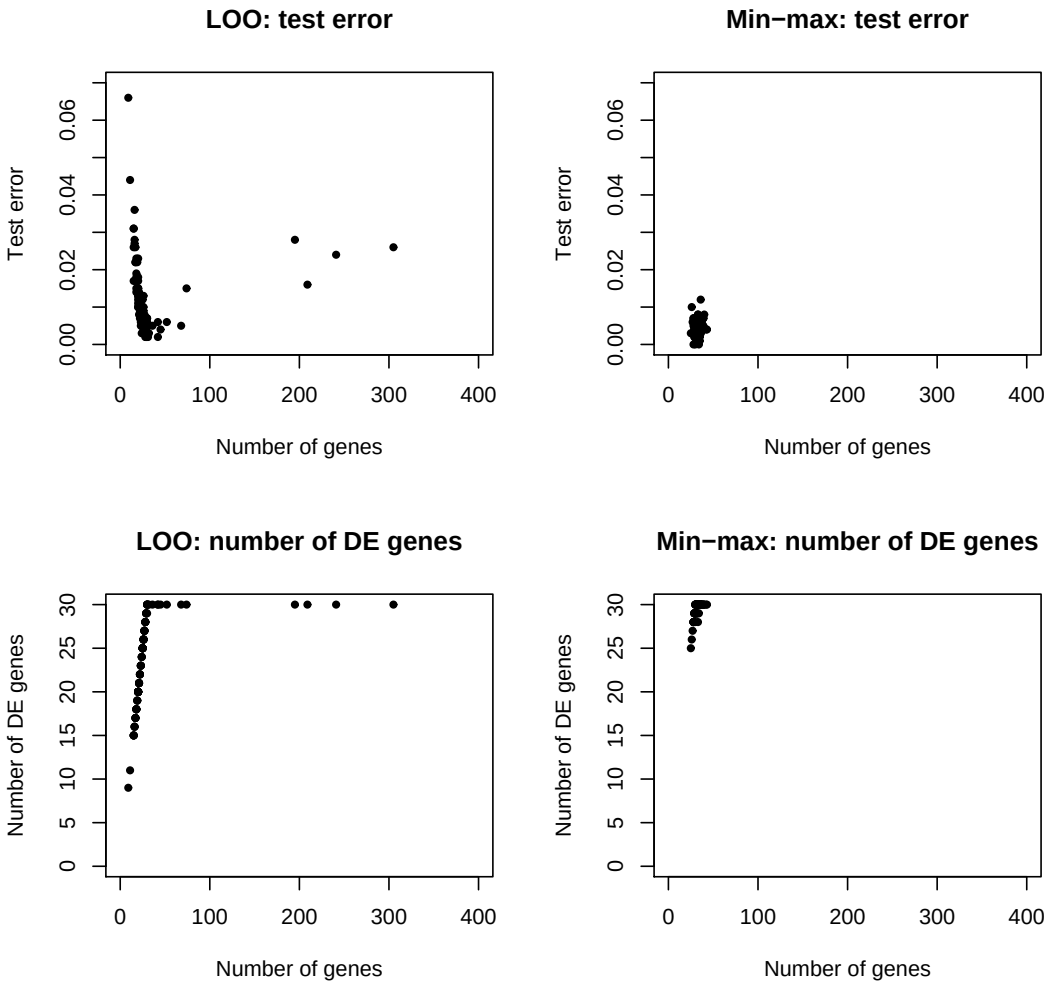


Figure 4.11: 150 samples

which required a reliable predictor with as small number of genes as possible. We investigated the resampling bootstrap distribution of classification error rates for supervised microarray classification problems and proposed a novel min-max criterion that was based on a simulated distribution of classification error rates.

In numerical comparisons on real and simulation datasets, we showed the stable nature of the min-max criterion in comparison to the state-of-the-art criterion. We also showed that the conventional LOO estimation of error rates resulted in large variances, consequently, the LOO criterion of model selection had a risk of choosing inappropriate classifiers that would exhibit extremely poor test error rates.

We compared two different supervised analysis procedures, T-WV and R-SVM, and concluded that the T-WV is suitable for mini-chip design. Although the mean and variance of the best performances, i.e. error rates, were not significantly different between T-WV and R-SVM, the best performances were achieved by smaller number of genes for T-WV than those for R-SVM. Thus, we recommend T-WV with min-max criterion in total.

For future studies, theoretical justification for our recommendation will be important. Although we compared two state-of-the-art methods, T-WV and R-SVM, there is room to develop a novel supervised classification method with smaller number of genes that incorporates the requirements of smaller expected error rate and smaller expected variance of the error rate for designing a mini-chip.

4.5 Summary

In order to design a low-cost minichip microarray for clinical application of a cancer diagnosis system, we need a supervised classifier involving the smallest number of genes, as long as the classifier is sufficiently reliable. To achieve a reliable classifier, we should assess candidates classifiers and select the best candidate. In the selection process of the best classifier, however, the assessment criterion must involve large variance because the number of samples is limited and observation noise is not negligible. Therefore, even if a classifier with a very small number of genes exhibited the smallest leave-one-out (LOO) error, the classifier would not necessarily be reliable because classifiers based on a small number of genes tend to show large variance.

We proposed a model selection criterion, the min-max criterion, based on a resampling bootstrap simulation to assess the variance of estimation of classification errors. We applied two state-of-the-art classifiers for microarray analysis, weighted voting

with t-statistics (T-WV) and a support vector machine with recursive feature elimination (R-SVM), to real and synthetic datasets and found that the conventional LOO criterion for WV classifiers had a non-negligible risk of selecting extremely poor classifiers and that the new min-max criterion could eliminate the risk. We also compared the T-WV and R-SVM, and showed that T-WV with the min-max criterion achieved the smallest error rate, which was equal to the best error rate by the R-SVM, with a smaller number of genes.

Chapter 5

Conclusions

We stated in Chapter 1, that we need to check the following two aspects of data, in order to build accurate predictive models in machine learning: (1) If the data is represented by a vector, is the number of dimensions small enough not to be affected by the so-called "curse of dimensionality"? (2) Is the amount of training data large enough to predict unseen test data?

For the issue addressed in (1), we investigated the case when a data object is represented as a high dimensional feature vector. In Chapter 3, we built a hypothesis that the Laplacian-based kernels reduce hubs following the report of Radovanović et al., which insisted that objects which are closer to the centroid become hubs in high dimensional spaces. We saw in the experiments in section 3.4 the hypothesis is true in some cases, but not always. However, when hubs are reduced by using the Laplacian-based kernels, the performance improves for tasks such as information retrieval, multi-class and multi-label classification.

The issue addressed for (2) in this thesis, we focused on the case where only small number of samples are available, yet, desired to obtain robust classifiers based on supervised learning techniques. The analysis of microarray gene expression data for cancer diagnosis is one of such cases. In Chapter 4, we investigated gene subset selection methods in order to design a low cost mini-chip microarray for diagnosis of cancers by gene expression profiles, which required a reliable predictor with as small number of genes as possible. We investigated the re-sampling bootstrap distribution of classification error rates and proposed a novel min-max criterion that was based on a simulated distribution of classification error rates. In the experiments on real as well as synthetic datasets, we showed that the min-max criterion made more stable results in comparison to the conventional LOO criterion. We also showed that the LOO

method for estimating error rates resulted in large variances, hence, the LOO criterion for model selection had a risk of choosing inappropriate classifiers that would exhibit extremely poor error rates for test data.

Bibliography

[1] A. A. Alizadeh, M. B. Eisen, E. E. Davis, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[2] U. Alon, N. Barkai, D. A. Notterman, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

[3] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory (ICDT '99)*, pages 217–235. Springer, 1999.

[4] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. of SIGIR-98*, pages 104–111, 1998.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[6] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. V. Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.*, 46(4):647–666, 2004.

[7] Y. L. Borgne. Bias variance trade-off characterization in a classification. what differences with regression? Technical report, ULB, 2005.

[8] U. M. Braga-Neto and E. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[9] P. O. Brown and D. Botstein. Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, 21:33–37, 1999.

[10] P. Y. Chebotarev and E. V. Shamis. The matrix-forest theorem and measuring relations in small social groups. *Automation and Remote Control*, 58(9):1505–1514, 1997.

[11] F. R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics 92. American Mathematical Society, 1997.

[12] D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[13] C. A. Davis, F. Gerick, V. Hintermair, et al. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics*, 22(19):2356–2363, 2006.

[14] S. Draghici, P. Khatri, A. C. Eklund, et al. Reliability and reproducibility issues in DNA microarray measurements. *Trends in Genetics*, 22:101–109, 2006.

[15] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[16] F. Fouss, L. Yen, A. Pirotte, and M. Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Proceedings of the Sixth International Conference on Data Mining (ICDM '06)*, pages 863–868, 2006.

[17] D. François, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19:873–886, 2007.

[18] J. H. Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Mining Knowledge Discovery*, 1(1):55–77, 1997.

[19] W. J. Fu, C. R. J, and S. Wang. Estimating misclassification error with small samples via bootstrap cross-validation. *Bioinformatics*, 21(9):1979–1986, 2005.

[20] T. Golub, D. Slonim, P. Tamayo, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[21] I. Guyon, J. Weston, S. Barnhill, et al. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

[22] M. Hagiwara. A supervised learning approach to automatic synonym identification based on distributional features. In *Proc. of the ACL-08: HLT Student Research Workshop*, pages 1–6, 2008.

[23] M. Hagiwara, Y. Ogawa, and K. Toyama. PLSI utilization for automatic thesaurus construction. In *Proc. of IJCNLP-05*, pages 334–345, 2005.

[24] M. Hagiwara, Y. Ogawa, and K. Toyama. Selection of effective contextual information for automatic synonym acquisition. In *Proc. of COLING/ACL-06*, pages 353–360, 2006.

[25] B. Haibe-Kains, C. Desmedt, S. Loi, et al. *Computational Intelligence in Clinical Oncology: Lessons Learned from an Analysis of a Clinical Study*, volume 122. Springer-Verlag Berlin/Heidelberg, 2008.

[26] Z. Harris. Distributional structure. *The Philosophy of Linguistics*, pages 26–47, 1985.

[27] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, USA, 2001.

[28] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[29] N. Iizuka, M. Oka, H. Yamada-Okabe, et al. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. *Lancet*, 361:923–929, 2003.

[30] J. Jaeger and R. Spang. Selecting normalization genes for small diagnostic microarrays. *BMC Bioinformatics*, 7:388, 2006.

[31] W. Jiang, S. Varma, and R. Simon. Calculating confidence intervals for prediction error in microarray classification using resampling. *tatistical Applications in Genetics and Molecular Biology*, 7, 2008.

[32] M. Komachi, T. Kudo, M. Shimbo, and Y. Matsumoto. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of*

the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP '08), pages 1011–1020, 2008.

[33] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning (ICML '02)*, 2002.

[34] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.

[35] A. M. Molinaro, R. Simon, and R. M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[36] M. Ohira, S. Oba, Y. Nakamura, et al. Expression profiling using a tumor-specific cDNA microarray predicts the prognosis of intermediate risk neuroblastomas. *Cancer Cell*, 7:337–350, 2005.

[37] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubs in space: popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

[38] M. Radovanović, A. Nanopoulos, and M. Ivanović. On the existence of obstinate results in vector space models. In *Proceedings of the 33rd Annual International Conference on Research and Development in Information Retrieval (SIGIR '10)*, pages 186–193, 2010.

[39] S. Ramaswamy, P. Tamayo, R. Rifkin, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98(26):15149–15154, 2001.

[40] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of graph, and its relationships to spectral clustering. In *Proceedings of ECML'04*, Lecture Notes in Artificial Intelligence 3201, pages 371–383. Springer, 2004.

[41] N. Shimizu, M. Hagiwara, Y. Ogawa, K. Toyama, and H. Nakagawa. Metric learning for synonym acquisition. In *Proc. of COLING-08*, pages 793–800, 2008.

[42] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and*

7th Kernel Workshop, Proceedings, Lecture Notes in Artificial Intelligence 2777, pages 144–158. Springer, 2003.

[43] van de Vijver M. J., Y. D. He, L. J. van't Veer, et al. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347(25):1999–2009, 2002.

[44] L. van't Veer, H. Dai, van de Vijver MJ, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

[45] V. N. Vapnik. *The nature of statistical learning theory*. Springer, New York, 2000.

[46] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.

[47] U. von Luxburg, A. Radl, and M. Hein. Getting lost in space: large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems 23*, pages 2622–2630, 2010.

[48] Y. Wang, J. Klijn, Y. Zhang, et al. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365:671–679, 2005.

[49] I. A. Wood, P. M. Visscher, and K. L. Mengersen. Classification based upon gene expression data: bias and precision of error rates. *Bioinformatics*, 23(11):1363–1370, 2007.

[50] Q. Xu, J. Hua, U. Braga-Neto, et al. Confidence intervals for the true classification error conditioned on the estimated error. *Technology in Cancer Research and Treatment*, 5:579–589, 2006.

[51] M.-L. Zhang and Z.-H. Zhou. ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, 40:2038–2048, 2007.

[52] X. Zhang, X. Lu, Q. Shi, et al. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7:197, 2006.

List of Publication

Journal Papers

(1) I. Suzuki, T. Takenouchi, M. Ohira , S. Oba and S. Ishii, Robust model selection for classification of microarrays, Cancer Informatics, Vol.7, pp.141-157, June 2009.

International Conference/Workshop Papers

(1) I. Suzuki, S. Oba and S. Ishii, A selection criterion for robust classifiers: cancer prognosis with microarray gene expression, in Genome Informatics Workshop (GIW), December 2005.

(2) I. Suzuki, K. Hara, M. Shimbo and Y. Matsumoto, A Graph-based Approach for Biomedical Thesaurus Expansion, In Proceedings of the ACM Third International Workshop on Data and Text Mining in Bioinformatics (DTMBIO), Short Papers, pp. 79-82. Hong Kong, November 2009.

Other Publication

(1) I. Suzuki, S. Oba and S. Ishii, A Selection Criterion for Robust Classifiers by Considering the Variance of Test Performance, in IEICE technical report. Neurocomputing, Vol.105, No.418, pp.25-30, November 2005. [in Japanese]

(2) I. Suzuki, K. Hara, M. Shimbo and Y. Matsumoto, Synonym Acquisition for Biomedical Thesaurus Expansion : a Graph-based Approach , in Information Processing Society of Japan, Natural Language Processing, Vol. 2009, No. 2, pp.65-70, January 2009. [in Japanese]

(3) I. Suzuki, K. Hara, M. Shimbo and Y. Matsumoto, Effectiveness of Laplacian-based kernels from the hubness point of view, in IEICE technical report. IBISML 2011-80, Vol. 111, No. 275, pp. 257-262, November 2011. [in Japanese]