THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Global geometric graph kernels and applications

FREDRIK D. JOHANSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG

Gothenburg, Sweden 2014

Global geometric graph kernels and applications
FREDRIK D. JOHANSSON

Thesis for the degree of Licentiate of Engineering
ISSN 1652-876X
Technical Report No. 124L
Department of Computer Science and Engineering

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Gothenburg, Sweden 2014

To mum and dad

Abstract

This thesis explores the topics of graph kernels and classification of graphs. Graph kernels have received considerable attention in the last decade, in part because of their value in many practical applications, such as chemoinformatics and molecular biology, in which classification using graph kernels have become the standard model for several problems. Perhaps even more important is the inclusion of graph kernels in the rich field of kernel methods, making a large family of machine learning algorithms, including support vector machines, applicable to data naturally represented as graphs.

Graph kernels are similarity functions defined on pairs of graphs. Traditionally, graph kernels compare graphs in terms of features of subgraphs such as walks, paths or tree patterns. For the kernels to remain computationally efficient, these subgraphs are often chosen to be small. Because of this fact, most graph kernels adopt an inherently local perspective on the graph and may fail to discern global properties, such as the girth or the chromatic number, that are not captured in local structure. Furthermore, existing work on graph kernels lack results justifying a particular choice of kernel for a given application.

In this thesis we propose two new graph kernels, designed to capture global properties of graphs, as described above. At the core of these kernels is Lovász number, an important concept in graph theory with strong connections to graph properties like the chromatic number and the size of the largest clique. We give efficient sampling approximations to both kernels, allowing them to scale to large graphs. We also show that we can characterize the separation margin induced by these kernels in certain classification tasks. This serves as initial progress towards making theory aid kernel choice. We make an extensive empirical evaluation of both kernels on synthetic data with known global properties, and on real graphs frequently used to benchmark graph kernels.

Finally, we present a new application of graph kernels in the field of data mining by redefining an important subproblem of entity disambiguation as a graph classification problem. We show empirically that our proposed method improves on the state-of-the-art.

Acknowledgements

Working on this thesis has been a pleasure, largely because of the people who have supported me throughout the process. It has been an immensly rewarding experience and would not have been possible without several people. First of all I would like to thank my advisor, Devdatt Dubhashi, for his constant support and overwhelming interest in all things research. He has taught me the value of always keeping an open mind.

I am immensly grateful to Vinay Jethava who has been my big brother in research since the beginning. He has taught me many things, and we've had a lot of fun together. (Yes, even when applying the toothbrush approach to meeting deadlines!)

Thanks also to Olof and Mikael, who have been great office mates, and to Joel, Nina, Willard, Azam, Leonid, Christos, Chien-Chung, Jacob and many others for interesting discussions and lots of fun. Thanks to Jonna Amgard, Peter Dybjer and Eva Axelsson for helping me with all kinds of things.

I would also like to thank Chiru Bhattacharyya, David Sands, Gerardo Schneider and Henk Wymeersch for their guidance and Svetoslav Marinov and Staffan Truvé for their interesting collaborations.

Thanks also to my family and other friends who have supported me in this process.

List of publications

This thesis is based on the following manuscripts.

Paper I F. D. Johansson, V. Jethava, D. Dubhashi, and C. Bhattacharyya (2014). "Global graph kernels using geometric embeddings". *Proceedings of the 31st International Conference on Machine Learning (ICML)*. ed. by T. Jebara and E. P. Xing. JMLR Workshop and Conference Proceedings, pp. 694–702

Paper II L. Hermansson, T. Kerola, F. Johansson, V. Jethava, and D. Dubhashi (2013). "Entity disambiguation in anonymized graphs using graph kernels". *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*. ACM, pp. 1037–1046

The following manuscripts have been published, but are not included in this work.

Paper III F. Johansson, V. Jethava, and D. Dubhashi (2013). "DLOREAN: Dynamic LOcation-aware REconstruction of multiwAy Networks". *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. IEEE, pp. 1012–1019

Paper IV F. Johansson, T. Färdig, et al. (2012). "Intent-aware temporal query modeling for keyword suggestion." *PIKM*. ed. by A. S. Varde and F. M. Suchanek. ACM, pp. 83–86

Contents

Part I
Extended summary

Chapter 1

Introduction

In recent years, we have seen a dramatic increase in the application of machine learning methods to graphs. Many labor-intensive tasks such as labelling or categorizing graph data can now be alleviated, if not solved, using machine learning methods. Problems of this kind arise in diverse fields, ranging from chemoinformatics and bioinformatics to social sciences, where graphs are well suited to represent the data. For example, in drug development, some candidate compounds will be harmful to humans while some will not, even if the compounds belong to the same group (Debnath et al. 1991). Predicting which of the compounds are harmless based on molecular structure, rather than through empirical studies, can represent large savings (Debnath et al. 1991). Another area rich in graph data is social network analysis (Wasserman and Faust 1994). In social networks, every node represents a person and every edge a relationship or an interaction. Recently, problems related to analysis of such networks have received considerable attention. For example, determining key influencers in a social network, can be highly profitable for marketing firms who can target influential individuals to maximize the spread of a campaign.

In order to harness the power of machine learning methods in the settings above, graphs must be represented in an efficient way. Most algorithms for important learning problems such as classification, clustering and dimensionality reduction, are designed for data represented by vectors of real values. An important exception is kernel methods that allow learning algorithms to interact with data via particular similarity functions, known as kernels. Defining expressive similarity functions for graphs is not trivial however. For example, there is no polynomial time algorithm for determining whether two graphs are isomorphic. To this end, graph kernels were introduced, representing an attractive middle-ground between precision and efficiency (Gärtner, Flach, and Wrobel 2003).

Graph kernels have been sucessfully used, chiefly in classification tasks, on diverse types of data (Shervashidze, Schweitzer, et al. 2011). Many existing graph kernels compare graphs based on a specific type of subgraphs, be it *walks*, *paths*, *subtrees* or *graphlets*. To remain efficient, most of these kernels consider only small subgraphs. While fast to compute, this fact may cause them to fail in capturing important global properties of graphs, such as the *clique number*, *girth*[1] or *chromatic number*. This is one of the core

[1]The girth of a graph is the length of the smallest cycle present in the graph.

issues addressed in this thesis.

Main contributions. This thesis is an extended summary of two papers. We make the following contributions in Paper I. We define two novel graph kernels motivated from graph theory, based on Lovász number, and designed to capture global properties of graphs. We show that on certain classification tasks, we can characterize the separation margin between classes of graphs. Further, we show empirically that our kernels are competitive with state-of-the-art graph kernels in terms of accuracy in classification of unlabeled benchmark graphs.

In Paper II we define an important subproblem within entity disambiguation as a graph classification problem. We make several extensions to existing graph kernels, designed for the entity disambiguation problem. We show empirically that these contributions leads to improved results in detecting ambiguous entities.

Thesis outline. In the remainder of Chapter 1, we give a background to and present theory relevant for the following chapters, as well as the appended papers. In Chapter 2, we summarize our work on global graph kernels, and in Chapter 3, our work on entity disambiguation.

1.1 Graphs, classification & kernels

The concept most central to this thesis is the *graph*. Graphs are denoted $G = (V, E)$ and comprise a set V of *nodes* or *vertices* and a set E of ordered pairs of nodes, or *edges*. If it holds that $(i, j) \in E \Rightarrow (j, i) \in E$, for any edge $e = (i, j)$, we call the graph *undirected*. If this does not hold, the graph is *directed*. Unless otherwise stated, we let $n = |V|$ and $m = |E|$. For our purposes, a graph may also be associated with a labelling function $L : V \to \mathcal{L}$, assigning a label to each node from a set of labels $\mathcal{L}$. Furthermore, the graph may have a weighting function $W : E \to \mathbb{R}$ assigning a real valued weight to every edge. If all nodes share the same label, i.e. L is a constant function, we call the graph *unlabeled*. If all edge weights are equal, we call the graph *unweighted*.

The canonical problem, on which we apply the methods developed in this thesis, is *graph classification*. Given a *training set* comprising pairs $\{(G^{(i)}, y_i)\}_{i=1}^{N}$ of graphs $G^{(i)}$ and class labels $y_i \in \mathcal{Y}$, our task is to automatically assign labels to a new, previously unseen *test set* of graphs $\{G^{(N+j)}\}_{j=1}^{N_{test}}$. In most cases, classification is binary, that is $\mathcal{Y} = \{-1, +1\}$. Classification is a general problem associated with rich theory. The concepts most relevant to this thesis are *support vector machines (SVM)* (Vapnik 1995) and *kernel methods* (Schölkopf and Smola 2001).

Support vector machines (Vapnik 1995) are supervised machine learning models commonly used for classification of many different types of data. While originally developed as linear, binary classifiers of real valued vector data, the *kernel trick* (Schölkopf and Smola 2001) enables implicit learning of nonlinear functions. The kernel trick exploits the observation that several learning algorithms, such as dual solvers for SVMs, interact with data only through inner products of pairs of data points. Inner products in the input space $\mathcal{X}$, may then be replaced by a kernel function expressed as an inner product

in another vector space $\mathcal{V}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a valid kernel only if its Gram matrix $K = (k(\mathbf{x}_i, \mathbf{x}_j)))_{ij\in[n]}$ is positive semi-definite for any choice of $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Valid kernels are often also called *positive definite kernels*.

The kernel trick removes the need for specifying an explicit mapping of the input data to another space, and lets us instead consider only similarities between data points. Not only does this allow for learning nonlinear functions, it also opens up kernel methods to data that is not naturally represented by real-valued vectors, but have some natural notion of similarity. The example of such data most relevant to this thesis is, of course, the graph.

1.2 Graph kernels

Graph kernels (Gärtner, Flach, and Wrobel 2003; S. Vishwanathan et al. 2010) are similarity measures on graphs defined on graphs. As such they enjoy all the theoretical benefits associated with kernel methods. They have also gained popularity in practical applications, and have been used in diverse fields including computational biology (Schölkopf, Tsuda, and Vert 2004), chemistry (Mahé and Vert 2009) and information retrival (See Paper II).

Graph kernels have primarily been motivated from the desire to capture similar structural properties in graphs (K. M. Borgwardt and Kriegel 2005). Searching for structural similarities in a pair of graphs is often computationally expensive, most notably perhaps in the case of subgraph isomorphism, widely known to be NP-hard. While not as precise, graph kernels represent an attractive trade-off between expressivity and computational efficiency(Ramon and Gärtner 2003).

Existing kernels predominantly compare graphs through counts or distributions of subgraph features. For example, random walk kernels (Gärtner, Flach, and Wrobel 2003; Kashima, Tsuda, and Inokuchi 2003) compare weighted counts of random walks of every length. The shortest-path kernel (K. M. Borgwardt and Kriegel 2005) compare features of the shortest paths between every pair of nodes in each graph, and subtree kernels (Ramon and Gärtner 2003; Mahé and Vert 2009) compare tree patterns. Recently, there has been a lot of research on how to handle node and edge attributes efficiently. An important family of kernels in that line of work are the Weisfeiler-Lehman kernels (Shervashidze, Schweitzer, et al. 2011), based on the Weisfeiler-Lehman isomorphism test.

Many graph kernels are *R-convolution* kernels (Shervashidze, Schweitzer, et al. 2011; S. Vishwanathan et al. 2010). We define the R-convolution kernel below.

Definition 1.2.1 (Haussler 1999)**.** *Let χ and χ' be spaces and $k : \chi' \times \chi' \to \mathbb{R}$ a positive semi-definite kernel. The R-convolution kernel for points $x, y \in \chi$, associated with finite subsets $\chi'_x \subseteq \chi'$ and $\chi'_y \subseteq \chi'$ is defined by*

$$K(x, y) = \sum_{(x',y')\in\chi'_x\times\chi'_y} k(x', y') \ . \tag{1.2.1}$$

R-convolution kernels are positive definite (Haussler 1999). Conceptually, graph kernels based on the R-convolution kernel compare features of small subgraphs or walks extracted from the original graphs. This leads to an inherently local perspective, which may fail

to capture global properties of graphs. Further, as Shervashidze, S. Vishwanathan, et al. 2009 identified, *"There is no theoretical justification on why certain types of subgraphs are better than others"*.

For example, the graphlet kernel (Shervashidze, S. Vishwanathan, et al. 2009) counts instances of subgraph patterns of at most 5 nodes. The random walk kernel (Gärtner, Flach, and Wrobel 2003) counts walks of any length, but the counts are often weighted with a factor decreasing exponentially with the length of the walk (S. V. N. Vishwanathan, K. M. Borgwardt, and Schraudolph 2007). Subtree kernels (Ramon and Gärtner 2003; Shervashidze and K. Borgwardt 2009), consider tree patterns of a limited size.

It is known, however, that there are graph properties which cannot be captured by studying only local structures, such as small subgraphs. Perhaps the most celebrated result on this topic is Erdős' seminal proof of existence of graphs with high girth and high chromatic number (Alon and Spencer 1992, p. 41-42), graphs for which all small-sized subgraphs will be trees. Because of this problem, we seek representations of graphs that capture precisely such properties. We give an introduction to one type of such representation in the following section.

1.3 Geometric representations of graphs

This section introduce geometric representations of graphs as well as the celebrated Lovász number, on which we build our new graph kernels.

A geometric representation of a graph $G = (V, E)$ is an embedding U_G of each node $v \in V$ into a geometric space $\mathbb{R}^p$,

$$U_G := \{\mathbf{u}_i \in \mathbb{R}^p\}_{i \in V} \ . \tag{1.3.1}$$

We say that a representation U_G is *orthogonal* if

$$(i, j) \notin E \Rightarrow \mathbf{u}_i^\top \mathbf{u}_j = 0 \ , \tag{1.3.2}$$

and *orthonormal* if also $\|\mathbf{u}_i\| = 1$ for all $i \in V$. We note that this definition on its own does not provide fruitful ways of representing graphs. For example, letting each $\mathbf{u}_i$ be a different basis vector, results in a valid representation, but does preserve the graph structure at all. In the sequel, we focus on a particular representation, associated with the celebrated Lovász number.

1.3.1 Lovász number and global properties of graphs

Lovász number (Lovász 1979), usually denoted $\vartheta(G)$, was introduced as a polynomial-time computable upper bound on the Shannon capacity of G, an important concept in information theory for which a polynomial time algorithm is not known. It was also shown to have the following attractive relationship with the clique number $\omega(G)$ and the chromatic number $\chi(G)$, both of which are NP-hard to compute.

$$\omega(G) \leq \vartheta(\bar{G}) \leq \chi(G) \tag{1.3.3}$$

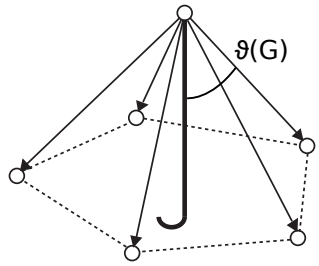


Figure 1.3.1: *Lovász number $\vartheta(G)$ and the orthogonal representation for the pentagon.*

Here $\bar{G}$ is the graph complement to G. The result above is sometimes refered to as Lovász *sandwhich theorem*. Because of the polynomial complexity of computing $\vartheta(G)$ and its relation to several important quantities, known to be NP-hard to compute, Lovász number has received considerable attention since its introduction.

Formally, $\vartheta(G)$ is defined as the smallest angle[2] of a cone, enclosing any orthonormal representation U_G,

$$\vartheta(G) = \min_{\mathbf{c}, U_G} \max_{i \in V} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \ , \tag{1.3.4}$$

where the minimization is taken over all orthonormal representations U_G and all unit vectors $\mathbf{c}$. An illustration of $\vartheta(G)$ can be seen in Figure 1.3.1.

Since its introduction, it has had large impact on combinatorial optimization, graph theory and approximation algorithms (Goemans 1997). $\vartheta(G)$ and the associated minimizing orthogonal representation, has been used to derive state-of-the-art approximation algorithms for max k-cut (Frieze and Jerrum 1997), graph coloring (Karger, Motwani, and Sudan 1998; Dukanovic and Rendl 2008) and planted clique problems (Feige and Krauthgamer 2000). These results provide ample motivation for us to design a graph kernel around $\vartheta(G)$ and aimed towards capturing global properties of graphs.

It is well-known that $\vartheta(G)$ can be computed to arbitrary precision in polynomial time, by means of solving a semi-definite program (Lovász 1979). While polynomial, state-of-the-art algorithms for computing Lovász number are often prohibitively slow for real-worl applications with time complexities $O(n^5 \log n \cdot \epsilon^{-2})$ (Chan, Chang, and Raman 2009) and $O(n^2 m \log n \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$ (Iyengar, Phillips, and Stein 2011), where n and m are the number of nodes and edges respectively and ϵ the error. In the next section, we introduce a recent approximation to $\vartheta(G)$ (Jethava et al. 2014).

1.3.2 The SVM-ϑ approximation

In this section, we introduce SVM-ϑ, a recent approximation to $\vartheta(G)$, with considerably lower computational complexity. Jethava et al. 2014 defined SVM-ϑ as an alternate characterization of $\vartheta(G)$, that involves solving a kernel one-class support vector machine (Schölkopf, Platt, et al. 2001). They observed that a one-class SVM, like $\vartheta(G)$,

[2]We note that $\vartheta(G)$ is really the inverse squared cosine of the half-angle of the cone, but as they grow and decrease together, we refer to $\vartheta(G)$ as angle henceforth.

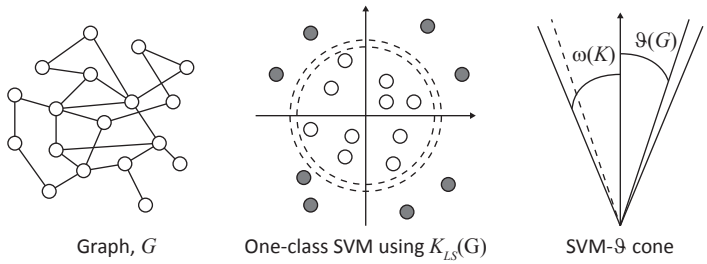


Figure 1.3.2: *The components of* SVM-ϑ *and an illustration of its relationship to* $\vartheta(G)$.

searches for the minimum cone enclosing a set of vectors, and that for a particular choice of kernel, the SVM and ϑ cones become equivalent.

Formally, for any graph $G = (V, E)$, such that $n = |V|$, it holds that

$$\vartheta(G) = \min_{\kappa \in L} \omega(\kappa) \tag{1.3.5}$$

where $\omega(\kappa)$ is the solution to a kernel one-class SVM,

$$\omega(\kappa) = \max_{\substack{\alpha_i > 0 \\ i=1,\ldots,n}} 2\sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa_{ij} \ . \tag{1.3.6}$$

L is the set of kernel matrices that respect the same orthogonality constraints as the orthonormal representations defined in Section 1.3,

$$L := \{\kappa \in S_n^+ | \kappa_{ii} = 1, \forall i, \kappa_{ij} = 0, (i,j) \notin E\} \ , \tag{1.3.7}$$

and S_n^+ is the set of $n \times n$ positive semi-definite matrices. With slight abuse of notation, from now on, we let α_i denote the *maximizers* of (1.3.6).

So far, we have not gained anything in terms of complexity. The optimization over L involves solving a semi-definite program, and the SVM simultaneously, which in general is no faster than computing $\vartheta(G)$. Instead, our hope is that a particular choice of kernel κ gives a good approximation to the minimum of (1.3.5), without doing the optimization. As it happens, there is a choice of κ, that while not optimal, gives good theoretical guarantees of this nature. We define this choice of κ below.

Definition 1.3.1 (Luz and Schrijver 2005)**.** *Let A be the adjacency matrix of G, $\rho \geq -\lambda_n(A)$, with $\lambda_n(A)$ the minimum eigenvalue of A, and set*

$$\kappa_{LS}(G) = \frac{A}{\rho} + I \succeq 0 \tag{1.3.8}$$

$\kappa_{LS}(G)$ can be thought of as a projection from the adjacency matrix A to the set of positive semi-definite matries. Jethava et al. 2014 showed that,

$$\omega(\kappa_{LS}(G)) = \sum_{i=1}^{n} \alpha_i \tag{1.3.9}$$

where α_i are the maximizers of (1.3.6). Henceforth, when referring to SVM-ϑ, we refer to the optimizers of (1.3.6) with $\kappa = \kappa_{LS}$. SVM-ϑ is illustrated in Figure 1.3.2, for this particular choice of κ. Jethava et al. 2014 proved that on families of graphs, referred to by them as SVM-ϑ graphs, $\omega(\kappa_{LS})$ is w.h.p. a constant factor approximation to $\vartheta(G)$,

$$\vartheta(G) \leq \omega(\kappa_{LS}) \leq \gamma\vartheta(G) \ . \tag{1.3.10}$$

Important graph families such as Erdős-Rényi random graphs and planted clique graphs have this property. SVM-ϑ is for a given kernel computable in $O(n^2)$ due to the one-class SVM (Hush et al. 2007), but κ_{LS} requires $O(n^3)$ time due to the computation of the minimum eigenvalue of A.

Chapter 2

Global graph kernels

This chapter introduces two novel graph kernels designed to capture important global properties of graphs, such as the girth or the clique number. In contrast to earlier graph kernels, focusing on local structure to maintain efficiency, our kernels are designed to capture global properties of graphs. To remain efficient, the kernels still decompose into features of substructures, but in a manner that retains desired properties.

We begin by defining the Lovász ϑ kernel based on Lovász number and the associated orthogonal representation. We then define a kernel on the SVM-ϑ approximation, enabling faster computation while retaining good accuracy. We show that for certain classification tasks, we can bound the separation margin induced by our kernels, providing theoretical justification for the choice of graph kernels in some applications .

Further, in Paper I, we show empirically that our kernel is competitive with state-of-the-art graph kernels on established benchmark datasets.

2.1 The Lovász ϑ kernel

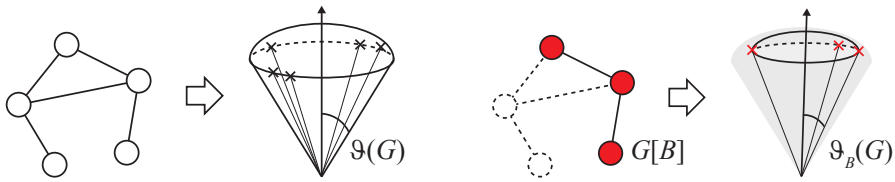
Motivated by the strong connection between $\vartheta(G)$ and global graph properties such as max-cut and graph coloring, as described in Section 1.3.1, we proceed to define the *Lovász ϑ kernel* using $\vartheta(G)$. Henceforth, when referring to an orthonormal representation $U_G = \{\mathbf{u}_1, \dots \mathbf{u}_n\}$, we always refer to the maximizer of (1.3.4).

We begin by defining the useful notion of the *Lovász value* of a subset of nodes $B \subseteq V$, which represents the angle of the smallest cone enclosing a subset of vectors $U_{G|B} \subseteq U_G$, as defined below.

Definition 2.1.1. *Let $G[B]$ be the subgraph of $G = (V, E)$ induced by $B \subseteq V$. Then, the* Lovász value *of $G[B]$ is defined by,*

$$\vartheta_B(G) = \min_{\mathbf{c}} \max_{\mathbf{u}_i \in U_{G|B}} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \ , \tag{2.1.1}$$

where $U_{G|B} := \{\mathbf{u}_i \in U_G \,|\, i \in B\}$ and U_G is the maximizer of (1.3.4).



(a) (b)

Figure 2.1.1: *An illustration of the difference between $\vartheta(G)$ (a) and $\vartheta_B(G)$ (b).*

Note that in general $\vartheta_B(G) \neq \vartheta(G[B])$. The difference between these quantities is what we'll exploit in building our kernel, and is illustrated in Figure 2.1.1. More specifically, $\vartheta_B(G)$ adheres to the global set of orthogonality constraints, defined by all of G. In contrast $\vartheta(G[B])$ uses only the information present in $G[B]$ and is therefore a completely local feature.

We now present the formal definition of Lovász ϑ kernel in terms of the Lovász values of subgraphs.

Definition 2.1.2 (Lovász ϑ kernel)**.** *The Lovász ϑ kernel on two graphs, G, G', with a positive semi-definite kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, is defined by*

$$K(G, G') = \sum_{B \subseteq V} \sum_{\substack{C \subseteq V' \\ |C|=|B|}} \frac{1}{Z_{B,C}} \cdot k(\vartheta_B, \vartheta'_C) \ , \tag{2.1.2}$$

with $\vartheta_B = \vartheta_B(G)$, $\vartheta'_C = \vartheta_C(G')$, and $Z_{B,C} = \binom{n}{|B|}\binom{n'}{|B|}$.

The Lovász ϑ kernel compares Lovász values for all pairs of subsets in two graphs. In effect, this corresponds to comparing the independence structure of subgraphs, as ϑ_B depends on how many vectors in $U_{G|B}$ are orthogonal, which in turn depends on how many nodes in B are independent. We can also prove the following result, important for any graph kernel.

Lemma 2.1.1 (Paper I, §3.1)**.** *The Lovász ϑ kernel, as defined in* (2.1.2)*, is a positive semi-definite kernel.*

Proof sketch. The proof involves showing that the kernel is an R-convolution kernel (Haussler 1999). For a complete proof, see Paper I.

As Lovász number is prohibitively expensive to compute for most graphs (see Section 1.3.1), we define a faster, approximate version of the Lovász ϑ kernel in the next section.

2.2 The SVM-ϑ kernel

We proceed to define a new graph kernel called the SVM-ϑ kernel. To create a faster kernel than the Lovász ϑ kernel, but with similar properties, we seek an SVM-ϑ analogue of the

Lovász value, ϑ_B to use as a feature of subgraphs. We note that α_i adhere to the global optimality conditions of (1.3.6) defined by the edge set, and thus capture some global properties of graphs. Based on this observation, and the connection between $\omega(\kappa)$ and $\vartheta(G)$ as described in Section 2.2, we let $\sum_{i\in B}\alpha_i$ serve as an analogue for ϑ_B in (2.1.2), when defining our new kernel.

Definition 2.2.1. *The* SVM-ϑ *kernel is defined, on two graphs* G, G'*, with corresponding* $\boldsymbol{\alpha}$*,* $\boldsymbol{\alpha}'$ *maximizers of* (1.3.6) *for* $\kappa = \kappa_{LS}(G)$*, with a positive semi-definite kernel* $k : \mathbb{R}\times\mathbb{R}\to\mathbb{R}$*, as*

$$K(G,G') = \sum_{B\subseteq V}\sum_{\substack{C\subseteq V'\\ |C|=|B|}} \frac{1}{Z_{B,C}} k(\mathbf{1}^\top\boldsymbol{\alpha}_B, \mathbf{1}^\top\boldsymbol{\alpha}'_C) \tag{2.2.1}$$

where $\boldsymbol{\alpha}_B = [\alpha_{B(1)},\dots,\alpha_{B(d)}]^\top$ *with* $d = |B|$*,* $Z_{B,C} = \binom{n}{|B|}\binom{n'}{|C|}$ *and* $\mathbf{1}$ *the all one vector of appropriate size.*

We make a note that while $\sum_{i=1}^n \alpha_i$ is an upper bound on $\vartheta(G)$, and an constant-factor approximation for classes of graphs, the same cannot be said for the entire SVM-ϑ kernel in relation to the Lovász ϑ kernel. This is due to the fact that $\sum_{i\in B}$ is not a tight bound on $\vartheta_B(G)$ for all $B\subset V$, even for SVM-ϑ graphs. Nevertheless, the SVM-ϑ kernel is still a valid kernel capable of capturing global properties of graphs, such as the clique number, as we will see in the next section. We can also show the following result.

Lemma 2.2.1 (Paper I, §4)**.** *The* SVM-ϑ *kernel, as defined in* (2.2.1)*, is a positive semi-definite kernel.*

Proof sketch. The proof is involves showing that the kernel is an R-convolution kernel (Haussler 1999). For a complete proof, see Paper I.

2.3 Classifying signal subgraphs

In this section we address classification of a particular class of graph, containing what is known as *signal subgraphs*. Signal subgraphs are subgraphs that have common properties within a class of graphs, but differ between classes. Imagine for example several measurements of brain activity. In some of them, certain areas express more activity than in others, possibly indicating an neruological disease. These areas can be modeled as signal subgraphs. Motivated by problems such as these, Vogelstein et al. 2013 introduced a framework for graph classification based on signal subgraphs. Devroye et al. 2011 had earlier considered a hypothesis testing problem arising in applications such as remote sensing and argued that it could be modelled as a *planted clique* problem in a random geometric graph.

Identifying a planted clique is a classical problem in the theory of random graphs and algorithms (Feige and Krauthgamer 2000; Alon, Krivelevich, and Sudakov 1998) with many applications such as cryptography (Juels and Peinado 2004). It also has connections to data mining problems such as epilepsy prediction (Iasemidis et al. 2001). In the classical planted clique problem, a hidden clique of $\Theta(\sqrt{n})$ vertices is planted

into a random graph and the goal is for an algorithm to identify it. In a more general version, the planted subgraph could have significantly higher or lower density compared with the underlying random graph. Such planted models are natural special cases of the general framework of Vogelstein et al. 2013 . In their brain networks setting, a denser subgraph could correspond to a subset of neurons that have significantly higher (or lower) connectivity compared to the rest of the network.

As an application of the kernels developed in this thesis, we consider binary classification of graphs, where one class comprises graphs containing a signal subgraph, and the other class does not. Specifically, we adress the case of Erdős-Rényi random graphs. We let $G(n,p)$ denote the random graph of n nodes, where every edge is present, randomly and independently, with probability p. Further, we let $G(n,p,k)$ denote the graph formed by sampling a random $G(n,p)$ graph and planting a clique of size k within.

We consider now on using the Lovász ϑ kernel for classification of $G(n,p)$ and $G(n,p,k)$ as two different classes. We give a result showing that the two classes of graphs are linearly separable with reasonably large margin in the feature space of the linear Lovász ϑ kernel.

Lemma 2.3.1. *There exist, with high probability, $Pr \geq 1 - O(1/n)$, a linear separator in linear Lovász ϑ kernel space, separating $G(n,p)$ and $G(n,1-p,k)$ graphs, $k = 2t\sqrt{\frac{n(1-p)}{p}}$, where $p(1-p) = \Omega(n^{-1}\log^4 n)$, with margin*

$$\gamma \geq (t-c)\sqrt{\frac{n(1-p)}{p}} - o(\sqrt{n}) \ ,$$

for some constant c, and large enough $t \geq 1$.

Proof. The proof is left to the supplementary material of Paper I.

The result above can be extended to hold for the SVM-ϑ kernel as well, as can be seen in Paper I. Results such as these represent intial steps towards theory aiding the choice of kernel for a particular classification task. When considering large graphs, this becomes increasingly important, as trial-and-error approaches to kernel choice become more expensive as the graphs grow. On this note, in the next section, we consider the problem of computing our proposed kernels efficiently.

2.4 Efficient computation

Direct evaluation of the Lovász ϑ and SVM-ϑ kernels is computationally very expensive. This is easily realized by considering the sums over subsets in the definition of either kernel, for which the number of terms grows exponentially with the size of the graph. To use these kernels with large graphs, we need to rely on approximate computation. In this section we derive such an approximation scheme, based on sampling.

We begin by noting that both the Lovász ϑ kernel and SVM-ϑ kernel can be written on the following form.

$$K(G,G') = \sum_{B\subseteq V} \sum_{\substack{C\subseteq V' \\ |C|=|B|}} \frac{1}{Z_{B,C}} k(f_B(G), f_C(G')) \tag{2.4.1}$$

with $f_B(G) = \vartheta_B(G)$ for the Lovász ϑ kernel and $f_B(G) = \sum_{j\in B} \alpha_j(G)$ for the SVM-ϑ kernel, and $Z_{B,C} = \binom{n}{|B|}\binom{n'}{|C|}$ for both. In this section, we derive an approximate computation scheme for the general form in (2.4.1) applicable to both kernels.

We note that (2.4.1) is easily decomposed into pairs of subsets of nodes. Now, instead of considering all pairs, we sample a small (polynomial) number of subsets for each graph, resulting in overall polynomial complexity. Formally, let S_d and S'_d be multisets of t uniformly sampled subsets of V and V' respectively, such that $|S_d| = |S'_d| = t$. If $d > n$, let $S_d = \emptyset$ and analoguously for S'_d and n'. Then, define,

$$\hat{K}(G,G') = \sum_{d=1}^{n_{max}} \sum_{B\in S_d} \sum_{C\in S'_d} \frac{1}{|S_d||S'_d|} k(f_B(G), f_C(G')) \ . \tag{2.4.2}$$

It is easy to see that (2.4.2) converges to (2.4.1), when the number of samples, k goes to infinity. In practice, it is of course not feasible to use an infinite number of samples. Instead, we are interested to know how many samples are sufficient to get a good approximation. Without specifying the base kernel k, it is difficult to produce a bound of this kind. However, letting k be the linear kernel $k(x,y) = xy$, we can derive precisely such a result.

For the linear base kernel, the general form in (2.4.1) is separable in the following fashion.

$$\begin{aligned} K(G,G') &= \sum_{d=1}^{n_{max}} \sum_{\substack{B\subseteq V \\ |B|=d}} \sum_{\substack{C\subseteq V' \\ |C|=d}} \frac{1}{Z_{B,C}} f_B(G) f_C(G') && (2.4.3) \\ &= \sum_{d=1}^{n_{max}} \left(\sum_{\substack{B\subseteq V \\ |B|=d}} \frac{1}{Z_B} f_B(G) \right) \left(\sum_{\substack{C\subseteq V' \\ |C|=d}} \frac{1}{Z_C} f_C(G') \right) && (2.4.4) \end{aligned}$$

with $Z_{B,C} = \binom{n}{|B|}\binom{n'}{|C|}$, $Z_B = \binom{n}{|B|}$. By defining

$$\varphi(d) = \sum_{\substack{B\subseteq V \\ |B|=d}} \frac{1}{Z_B} f_B(G) \ , \tag{2.4.5}$$

we see that $K(G,G') = \boldsymbol{\varphi}^\top \boldsymbol{\varphi}'$. A similar result can be obtained for the approximate

kernel in (2.4.2), with $\hat{\varphi}(d) = \sum_{B \in S_d} \frac{1}{|S_d|} f_B(G)$,

$$\hat{K}(G, G') = \sum_{d=1}^{n_{max}} \sum_{B \in S_d} \sum_{C \in S'_d} \frac{1}{Z_{B,C}} f_B(G) f_C(G') \tag{2.4.6}$$

$$= \hat{\boldsymbol{\varphi}}^\top \hat{\boldsymbol{\varphi}}' \,. \tag{2.4.7}$$

Now, for the case of the linear kernel, we may derive sample bounds for the feature vector representation in (2.4.7), instead of the kernel itself. By bounding the range of $f_B(G)$ and using standard Chernoff bounds, we can prove the following result for the Lovász ϑ kernel.

Theorem 2.4.1. *For graphs of n nodes, each coordinate $\varphi(d)$ of the feature vector of the linear Lovász ϑ kernel can be estimated by $\hat{\varphi}(d)$ such that*

$$Pr\left[\hat{\varphi}(d) \geq (1+\epsilon)\varphi(d)\right] \leq O(1/n) \tag{2.4.8}$$

$$Pr\left[\hat{\varphi}(d) \leq (1-\epsilon)\varphi(d)\right] \leq O(1/n) \tag{2.4.9}$$

using $s_d = O(n \log n/\varepsilon^2)$ samples.

Proof sketch. We apply a multiplicative Chernoff bound on ϑ_{V_r} of sampled subsets V_r. For a full proof, see the supplementary material to Paper I.

The result can be extended to other functions $f_B(G)$ by bounding the range of the corresponding function. In Paper I, we do precisely this for the SVM-ϑ kernel.

2.5 Empirical evaluation

For a comprehensive empirical evaluation of both the Lovász ϑ kernel and the SVM-ϑ kernel, on synthetic graphs with known global properties as well as real-world graphs used as benchmarks for graph kernels, see Paper I, Section 5.

Chapter 3

Applications of graph kernels – Entity disambiguation

Modern data mining applications increasingly deal with vast amounts of text data, often with references to entities such as people and companies. To enable efficient use of such data, it needs to be structured in a way that that is accessible to both humans and algorithms. Annotating documents with names of people mentioned in the text is an example of information extraction of that kind. For instance, a user might be interested to know which cities TED talks curator Chris Anderson is visiting this year. An automated reply to such a query requires extraction of names and places from news texts etc. This task is made difficult by the existence of Chris's namesake, former Wired Magazine editor-in-chief Chris Anderson. A naïve system, considering only the names in isolation, would answer that both Chris's are the same person.

Resolving ambiguities such as the one above is called *entity disambiguation* or *entity resolution* and is a problem which appears in many contexts. In its most general form, this problem is one of finding a mapping between a set of *identifiers* and a set of *entities*. In the example above, names are the identifiers and people are then entities. Related problems include record linkage (Fellegi and Sunter 1969), deduplication (Culotta and McCallum 2005), object distinction (Yin, Han, and Yu 2007) and co-reference resolution (Haghighi and Klein 2007). An subclass of entitity disambiguation problems is *relational entity disambiguation* which makes use of graph structure between entities (Bhattacharya and Getoor 2006b; Bhattacharya and Getoor 2004; Bekkerman and McCallum 2005; Malin 2005). Such information is available in many different contexts. In the example of text documents, entities are related through documents in which they are mentioned together.

In this thesis, we explore an important subproblem of relational entity disambiguation, namely that of determining which identifiers that are ambiguous, i.e. that are used to refer to several different entities. In this chapter, we proceed to define this problem formally and give our approach to solving it using graph kernels. We also propose extensions to existing graph kernels, tailored for the entity disambiguation task. In Paper II we present an empirical evaluation of the proposed method showing that the proposed kernel extensions result in improved classificuation accuracy.

3.1 Entity disambiguation

Resolution of ambiguities in data is a well-studied problem and methods for entity resolution (Bhattacharya and Getoor 2006a; Bhattacharya and Getoor 2006b; Elsayed, Oard, and Namata 2008; Sen 2012), entity matching (Böhm et al. 2012; Rastogi, Dalvi, and Garofalakis 2011) and entity disambiguation (Bunescu and Pasca 2006; Cucerzan 2007; Diehl, Getoor, and Namata 2006; Malin 2005) are all aimed towards associating references in text sources with the correct underlying entity. These methods typically make use of similarities in names (Bhattacharya and Getoor 2006a; Bhattacharya and Getoor 2006b), meta-data (Böhm et al. 2012) or source information (Malin 2005), to decide which entities underly which references.

In this thesis, we let the term *entity* refer to a person or a company etc. while an *identifier* is a name or a label. If several entities have the same identifier, we say that the identifier is *ambiguous*. While a single entity may have several identifiers, we do not address this here; we focus only on ambiguities. In the relational entity disambiguation setting, entities are assumed to be related according to some unknown graph structure, or *entity graph*. We assume that this graph can be partially observed through a graph of identifier relations, or *identifier graph*. Make sure to note the difference between these two graphs, as the identifier graph contains ambiguities we wish to resolve, while the entity graph does not. The difference is illustrated in Figure 3.1.1.

Our running example is the setting in which identifiers are used in a corpus of documents. We let the identifier graph be the graph with one node for each identifier and an edge between every pair of identifiers co-occuring in at least one document. Edges are weighted by the significance of the relationships, such as number of co-occurrences. To provide mild anonymization of the data, we assume that the identifiers have been assigned in a pseudo-random way. In effect, the only information available to our method about the entities and their identifiers is the identifier graph.

We proceed to define anonymized relational entity disambiguation as the following classification problem.

Definition 3.1.1 (Anonymized relational entity disambiguation)**.** *Given an undirected identifier graph* $G = (V, E)$ *with edge weights* $w_{ij} \in \mathbb{R}^+$ *and training data* $S = \{(v_i, y_i) : 1 \leq i \leq m, v_i \in V, y_i \in \{\pm 1\}\}$ *that labels certain nodes as ambiguous* $(+)$ *or unambiguous* $(-)$*,* anonymized relational entity disambiguation *is the task of classifying new nodes as* $+1$ *or* -1*. Each node of* G *may refer to a single entity or several underlying entities. The weight of an edge signifies the importance of the connection between two nodes.*

Note that this definition does not include the actual separation of the entities that share identifiers. However, this problem is of great importance, as pointing out which identifiers are ambiguous can represent very large computational savings for the more expensive task of resolving the ambiguities.

3.1.1 Our approach

Consider Figure 3.1.1 again as it aims to illustrate some of the intuition behind our assumptions. To the left is an identifier graph and to the right the corresponding entity

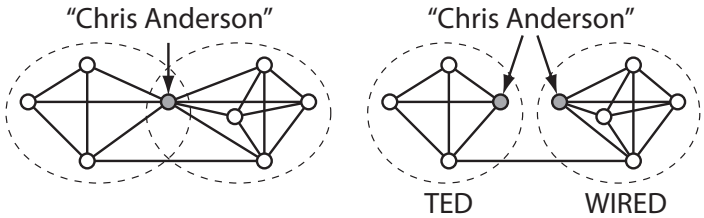


Figure 3.1.1: *A toy example of an identifier graph (left) and the corresponding entity graph (right). In this example, "Chris Anderson" is an ambiguous identifier.*

graph (assuming "Chris Anderson" is the only ambiguous identifier). In the figure, *two* individuals called Chris Anderson have been assigned only *one*, common identifier and thus *one* common node in the graph. This example shows how two otherwise only loosely connected communities (TED and Wired) can become strongly connected in the identifier graph through a single ambiguous node. In other words, it highlights our intuition that the graph structure surrounding "Chris Anderson" is indicative of whether the identifier is ambiguous or not. Although this example involves only people, we stress that nodes can represent any type of entity; an equally troublesome example would be that of the two *cities* Paris, France and Paris, Texas.

We proceed to describe our approach to the entity disambigatuon problem as defined in Definition 3.1.1. Motivated by our intuition that graph structure is indicative of ambiguity, we will formulate our solution to be the result of graph classification. More precisely, we let each node $v_i \in V$ be represented by its κ-neighborhood, $\mathcal{N}_\kappa^{(i)}$ as defined below.

Definition 3.1.2 (κ-neighborhood)**.** *Let $G = (V, E)$ be a graph. Then for any $v_i \in V$, the κ-neighborhood, $\mathcal{N}_\kappa^{(i)}$ is defined by*

$$\begin{aligned} V_\kappa^{(i)} &= \{v_i\} \cup \{v_j \in V : s(v_i, v_j) \leq \kappa\} \\ E_\kappa^{(i)} &= \{(v_p, v_q) : (v_p, v_q) \in E \wedge v_p, v_q \in V_\kappa^{(i)}\} \\ \mathcal{N}_\kappa^{(i)} &= (V_\kappa^{(i)}, E_\kappa^{(i)}) \ . \end{aligned} \tag{3.1.1}$$

In the example of Figure 3.1.1, the 1-neighborhood of the "Chris Anderson" node in the identifier graph is the entire graph.

As each identifier is now represented by a graph of its own, we are ready to define our approach in terms of binary graph classification. We let ambiguous identifiers be represented by a label (+1) and unambiguous by (-1). The setup above allows us to define our approach as shown in Algorithm 1 which we apply to the identifier graph G.

3.1.2 Extensions of graph kernels

In Paper II, in addition to the definition of relational entity disambiguation as a graph classification problem, we make several extensions to existing graph kernels. For a complete description of these extensions, see Paper II, Section 4.

The choice of which kernel to use in Algorithm 1 can have a large impact on the outcome, as can be seen in Paper II, Section 5. To increase the accuracy of the classification,

Algorithm 1 DETECTAMBIGUOUSNODES$(G = (V, E), \kappa, Y, S, T)$

Input: $G = (V, E)$
Input: $Y = \{y_i : i \in S \subset V, y_i \in \{\pm 1\}\}$ - Training labels
Input: $T \subset V$ - Test set
Input: κ - Neighborhood size.
for $v_i \in V$ **do**
  Set $G^{(i)} = \mathcal{N}_\kappa^{(i)}$ according to (3.1.1)
end for
Compute graph kernel matrix $K_{ij} = k(\mathcal{N}_\kappa^{(i)}, \mathcal{N}_\kappa^{(j)}),\ \forall i, j \in S$
Train an SVM with K and labels Y
Output: SVM classification of test nodes T.

we devise several extensions to existing graph kernels. The first is an extension of the random walk kernel (Gärtner, Flach, and Wrobel 2003) based on the observation that the graphs we are classifying are *pointed*, they have a distinguished node representing the identifier of interest. To that end, we consider a kernel variant counting only random walks originating from the distinguished node, not all possible walks.

The second extension we make is to the delta shortest-path kernel (K. M. Borgwardt and Kriegel 2005). The delta variant simply counts the number of paths of equal length in pairs of graphs, and can thus be computed in $O(n^2)$ time, modulo shortest-path computation, as compared to $O(n^4)$ of the general kernel. In our extension, we consider distances of shortest paths, as defined by edge weights, not just the number of steps. Because distances are often real valued, it is not fruitful to test them for equality. Instead, we consider a variant where distances are discretized into a finite number of bins, allowing us to use maintain the efficiency of the delta kernel. We consider different kinds of binnings to compensate for the often power-law like distribution of edge weights. Last, we consider normalizing the resulting feature vectors for all kernels, effectively forming the cosine similarity between feature vectors.

3.2 Empirical evaluation

For an empirical evaluation of our proposed method in the task of disambiguating identifiers in real-world datasets, see Paper II, Section 5. We show that we are better than or competitive to state-of-the-art methods and that all of our kernel extensions increase classification accuracy.

Chapter 4

Concluding remarks

In this thesis we have defined two novel graph kernels based on the concepts of geometric representations of graphs, the Lovász ϑ and SVM-ϑ kernels, designed to capture global properties of graphs. We have shown that we can characterize the separation margin of our kernel for classes of graphs. Further, we have shown efficient ways of computing the kernels approximately using sampling.

The area of graph kernels has been active for over a decade, exploring different features of graphs and new applications. An important trend in recent years is a focus on handling attributed graphs of varying kind. In many applications, leveraging labels on nodes gives a significant increase in classification accuracy. As of yet, neither of our kernels have a natural way of handling such data. Therefore, one of the most promising directions for future work is to explore kernels based on geometric representations of labeled graphs.

Another avenue of research remains to be the lowering of the cost of computing the kernels defined in this paper. While SVM-ϑ offers a tremendous speed-up compared to Lovász ϑ, computing even the sampled versions of our kernels still require considerable time. We note that the Lovász ϑ kernel involves comparing point sets, a problem that we've chosen to solve by comparing angles of subsets of points. However, comparing sets of points is a general problem, and future research should evaluate other approaches to it, in defining new graph kernels. Another way of speeding up computation of SVM-ϑ is to consider other kernels, κ, for the one-class SVM than that described in Section 2.2, which requires $O(n^3)$ time for computing the minimum eigenvalue.

Another line of work open for exploration is that of other global properties potentially captured by our kernel. This notion can be expanded to the entire field of graph kernels, in which very little research has been done on which kernels capture which properties.

Finally, inspired by our application of graph kernels to pseudo-anonymized graphs in entity disambiguation, we have work in progress to consider the use of graph kernels under the much stronger notion of differential privacy.

References

Alon, N. and J. Spencer (1992). *The Probabilistic Method*. Chichester: Wiley.

Alon, N., M. Krivelevich, and B. Sudakov (1998). Finding a large hidden clique in a random graph. *Random Struct. Algorithms* **13**.3-4, 457–466.

Bekkerman, R. and A. McCallum (2005). "Disambiguating Web appearances of people in a social network". *Proc. of WWW*.

Bhattacharya, I. and L. Getoor (2004). "Iterative record linkage for cleaning and integration". *Proc. of DMKD*.

— (2006a). "A Latent Dirichlet Model for Unsupervised Entity Resolution". *Proc. of SDM*.

— (2006b). "Entity Resolutions in Graphs". *Mining Graph Data*. Wiley.

Böhm, C. et al. (2012). "LINDA: distributed web-of-data-scale entity matching". *Proc. of CIKM*.

Borgwardt, K. M. and H.-P. Kriegel (2005). "Shortest-path kernels on graphs". *Proceedings of ICDM*, pp. 74–81.

Bunescu, R. and M. Pasca (2006). "Using encyclopedic knowledge for named entity disambiguation". *Proc. of EACL*.

Chan, T.-H. H., K. L. Chang, and R. Raman (2009). "An SDP Primal-dual Algorithm for Approximating the Lovász-theta Function". *Proceedings of ISIT*. Coex, Seoul, Korea: IEEE Press, pp. 2808–2812.

Cucerzan, S. (2007). "Large-scale named entity disambiguation based on Wikipedia data". *Proc. of EMNLP-CoNLL*.

Culotta, A. and A. McCallum (2005). "Joint deduplication of multiple record types in relational data". *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pp. 257–258.

Debnath, A. K. et al. (1991). Structureactivity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry* **34** (2), 786–797.

Devroye, L. et al. (2011). High-Dimensional Random Geometric Graphs and their Clique Number. *Electron. J. Probab.* **16**, no. 90, 2481–2508.

Diehl, C. P., L. Getoor, and G. Namata (2006). "Name reference resolution in organizational email archives". *Prof. of SDM*.

Dukanovic, I. and F. Rendl (Feb. 22, 2008). A semidefinite programming-based heuristic for graph coloring. *Discrete Applied Mathematics* **156**.2, 180–189.

Elsayed, T., D. W. Oard, and G. Namata (2008). "Resolving personal names in email using context expansion". *Proc. of ACL*.

Feige, U. and R. Krauthgamer (2000). Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms* **16**.2, 195–208.

Fellegi, I. P. and A. B. Sunter (1969). A Theory for Record Linkage. *Journal of the American Statistical Association* **64**, 1183–1210.

Frieze, A. M. and M. Jerrum (1997). Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica* **18**.1, 67–81.

Gärtner, T., P. Flach, and S. Wrobel (2003). On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines*, 129–143.

Goemans, M. X. (1997). Semidefinite programming in combinatorial optimization. *Math. Program.* **79**, 143–161.

Haghighi, A. and D. Klein (June 2007). "Unsupervised Coreference Resolution in a Nonparametric Bayesian Model". *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Haussler, D. (1999). *Convolution kernels on discrete structures*. Tech. rep. University of California at Santa Cruz.

Hermansson, L., T. Kerola, F. Johansson, V. Jethava, and D. Dubhashi (2013). "Entity disambiguation in anonymized graphs using graph kernels". *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*. ACM, pp. 1037–1046.

Hush, D. R. et al. (Feb. 21, 2007). QP Algorithms with Guaranteed Accuracy and Run Time for Support Vector Machines. *Journal of Machine Learning Research* **7**, 733–769.

Iasemidis, L. D. et al. (2001). Quadratic Binary Programming and Dynamical System Approach to Determine the Predictability of Epileptic Seizures. *J. Comb. Optim.* **5**.1, 9–26.

Iyengar, G., D. J. Phillips, and C. Stein (2011). Approximating Semidefinite Packing Programs. *SIAM Journal on Optimization* **21**.1, 231–268.

Jethava, V. et al. (2014). Lovasz theta function, SVMs and Finding Dense Subgraphs. *Journal of Machine Learning Research* **14**, 3495–3536.

Johansson, F. D., V. Jethava, D. Dubhashi, and C. Bhattacharyya (2014). "Global graph kernels using geometric embeddings". *Proceedings of the 31st International Conference on Machine Learning (ICML)*. Ed. by T. Jebara and E. P. Xing. JMLR Workshop and Conference Proceedings, pp. 694–702.

Johansson, F., T. Färdig, et al. (2012). "Intent-aware temporal query modeling for keyword suggestion." *PIKM*. Ed. by A. S. Varde and F. M. Suchanek. ACM, pp. 83–86.

Johansson, F., V. Jethava, and D. Dubhashi (2013). "DLOREAN: Dynamic LOcation-aware REconstruction of multiwAy Networks". *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. IEEE, pp. 1012–1019.

Juels, A. and M. Peinado (Feb. 5, 2004). Hiding Cliques for Cryptographic Security. *Des. Codes Cryptography* **20**.3, 269–280.

Karger, D. R., R. Motwani, and M. Sudan (1998). Approximate Graph Coloring by Semidefinite Programming. *J. ACM* **45**.2. Earlier version in FOCS'94, 246–265.

Kashima, H., K. Tsuda, and A. Inokuchi (2003). "Marginalized kernels between labeled graphs". *Proceedings of the 20th International Conference on Machine Learning*.

Lovász, L. (1979). On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **25**.1, 1–7.

Luz, C. J. and A. Schrijver (2005). A Convex Quadratic Characterization of the Lovász Theta Number. *SIAM J. Discrete Math.* **19**.2, 382–387.

Mahé, P. and J.-P. Vert (Dec. 14, 2009). Graph kernels based on tree patterns for molecules. *Machine Learning* **75**.1, 3–35.

Malin, B. (2005). "Unsupervised name disambiguation via social network similarity". *Workshop on link analysis, counterterrorism, and security*. Vol. 1401, pp. 93–102.

Ramon, J. and T. Gärtner (2003). "Expressivity versus Efficiency of Graph Kernels". *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences at ECML/PKDD*. Ed. by L. D. Raedt and T. Washio, pp. 65–74.

Rastogi, V., N. N. Dalvi, and M. N. Garofalakis (2011). Large-Scale Collective Entity Matching. *PVLDB* **4**.4, 208–218.

Schölkopf, B., J. C. Platt, et al. (July 2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation* **13**.7. Ed. by and.

Schölkopf, B. and A. J. Smola (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., K. Tsuda, and J.-P. Vert (2004). *Kernel methods in computational biology*. The MIT press.

Sen, P. (2012). "Collective context-aware topic models for entity disambiguation". *Proc. of WWW*.

Shervashidze, N. and K. Borgwardt (2009). "Fast Subtree Kernels on Graphs". *Proceedings of NIPS*, pp. 1660–1668.

Shervashidze, N., P. Schweitzer, et al. (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research* **12**, 2539–2561.

Shervashidze, N., S. Vishwanathan, et al. (2009). "Efficient graphlet kernels for large graph comparison". *Proceedings of AISTATS*.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vishwanathan, S. V. N., K. M. Borgwardt, and N. N. Schraudolph (Oct. 25, 2007). "Fast Computation of Graph Kernels." *NIPS*. Ed. by B. Schölkopf, J. Platt, and T. Hoffman. MIT Press, pp. 1449–1456. ISBN: 0-262-19568-2. URL: `http://dblp.uni-trier.de/db/conf/nips/nips2006.html#VishwanathanBS06`.

Vishwanathan, S. et al. (2010). Graph kernels. *Journal of Machine Learning Research* **11**, 1201–1242.

Vogelstein, J. T. et al. (2013). Graph Classification Using Signal-Subgraphs: Applications in Statistical Connectomics. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**.7, 1539–1551.

Wasserman, S. and K. Faust (1994). *Social network analysis: Methods and applications*. Cambridge Univ Pr.

Yin, X., J. Han, and P. Yu (2007). "Object distinction: Distinguishing objects with identical names". *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*.