Los Alamos
National Laboratory

Anomaly Detection on Graphs

Don Hush and James Howse
Modeling, Algorithms and Informatics Group, CCS-3
Los Alamos National Laboratory
{dhush,jhowse}@lanl.gov

LANL Technical Report: LA–UR–05–8440

Report Date: November 3, 2005

Abstract

This paper describes the results of some initial experiments that explore the problem of anomaly detection on graphs.

1 Introduction

Anomaly (novelty) detection attempts to detect anomalous observations from a system. Since the set of rules characterizing anomalous behavior is unknown, it must be learned from sample observations.

Consider the case where the system we are studying is all groups of people who are interacting so they can accomplish some set of tasks. We would like to identify collections of people which are unusual in some way. This is a reasonable first step in the process of trying to identify groups which may be dangerous. It seems unlikely that we can write down a complete set of rules characterizing anomalous behavior of groups of people, but we can obtain sample observations. It is quite natural to represent a group of interacting people by a graph. Therefore the problem of identifying unusual groups of people is an anomaly detection problem whose inputs are graphs.

We have a unique capability for solving this problem because of our recent discovery (Steinwart, Hush, & Scovel, 2005) that allows us to design anomaly detectors having both proven performance and guaranteed computational efficiency. Our work provides, for the first time, a performance measure for anomaly detection that can be estimated from data. Furthermore, we show that anomaly detectors can be designed using standard classification algorithms. In addition, our formalism allows the anomaly detector to have an arbitrary input space (*e.g.*, graphs). In this report we demonstrate the practical utility of our discovery by applying our methods to an anomaly detection problem where the graphs represent the interactions between characters in the famous novels Anna Karenina, David Copperfield, The Iliad, Huckleberry Finn, and Les Misérables.

2 Anomaly Detection Problem Formulation

Anomalies are often described as rare or unusual events. This notion can be represented mathematically by defining anomalies to be points with low probability density value. In particular the set of points with density value below a threshold ρ comprise the *anomalous set*, while the complement of this set is called the *normal set*. Our goal is to design a binary function (an anomaly detector) that assigns the value -1 to points in the anomalous set and $+1$ to points in the normal set.

To formalize these notions we first recall the basic concept of *density*. *Density* is a (local) valuation of the relative concentration of two measures. In particular, for two measures Q and μ on a space X where Q is absolutely continuous with respect to μ (i.e. every μ–negligible set is a Q–negligible set) the density h of Q with respect to μ is the Radon–Nikodym derivative $h = dQ/d\mu$. In the anomaly detection problem Q is an (unknown) probability measure that describes the data and μ is a (known) reference measure. For example when $X \subseteq \mathbb{R}^d$ the reference μ is usually taken to be the Lebesgue measure (*i.e.*, the standard volume). In principle however the reference measure is chosen by the user in a way that establishes a definition of anomalies relevant to the application. For example in this paper X is a graph space, and while a uniform measure on graphs is analogous to the Lebesgue measure on $\mathbb{R}^d$, we will see later that a uniform measure may not be a good choice for the reference.

Given a density level $\rho > 0$, the normal set $\{h > \rho\}$ is called the *ρ-level set*. The goal of the *density level detection* (DLD) problem is to find an estimate of the ρ-level set of h and therefore an estimate of the anomalous set (by taking the complement). To find this estimate we use information given to us by a training set $T = (x_1, \ldots, x_n) \in X^n$ that is i.i.d. drawn from Q. With the help of T a DLD algorithm constructs a function $\hat{f} : X \to \mathbb{R}$ for which the set $\{\hat{f} > 0\}$ is an estimate of the ρ-level set $\{h > \rho\}$. A standard performance measure that quantifies how well $\{\hat{f} > 0\}$ approximates the set $\{h > \rho\}$ is (see e.g. (Ben-David & Lindenbaum, 1997))

$$\mathcal{S}(f) := \mu\Big(\{f > 0\} \vartriangle \{h > \rho\}\Big)\,, \tag{1}$$

where $\vartriangle$ denotes the symmetric difference. The goal of the DLD learning problem is to find $\hat{f}$ such that $\mathcal{S}(\hat{f})$ is close to zero. It is important to note that regardless of how we attempt to achieve this goal we encounter a problem when attempting to validate our result. Indeed, since the density h is unknown we cannot compute the function $\mathcal{S}$ and therefore the performance $\mathcal{S}(\hat{f})$ is generally unknown. Furthermore, there appears to be no way to compute a reliable estimate of $\mathcal{S}$ from sample data. Thus it would seem that the only way to rigorously certify the performance of a solution method is through a deductive analysis that relies on premises that we hope or believe to be true (e.g. by establishing a bound of the form $Pr(\mathcal{S}(\hat{f}) > \epsilon) < \delta$ where ϵ and δ are small and the bound holds for all distributions that we believe will be encountered in practice). However the discovery in Steinwart et al. (2005), which we now describe, provides a resolution to this impasse.

Let μ be a probability measure and define the risk

$$\mathcal{R}(f) := \frac{1}{1+\rho}\, Q(f \le 0) + \frac{\rho}{1+\rho}\, \mu(f > 0). \tag{2}$$

Steinwart et al. (2005) show that any function that minimizes $\mathcal{R}$ also minimizes $\mathcal{S}$. Furthermore they prove a very tight relation between $\mathcal{R}$ and $\mathcal{S}$ for all functions f. This establishes $\mathcal{R}$ as a bona fide risk function for the DLD problem. Therefore $\mathcal{R}$ *is a legitimate performance measure for anomaly detection*. Consequently our goal of choosing $\hat{f}$ to (approximately) minimize $\mathcal{S}$ can be revised to choosing $\hat{f}$ to (approximately) minimize $\mathcal{R}$. This is important because, unlike $\mathcal{S}$, we *can* compute a reliable estimate of $\mathcal{R}$ from sample data. For example if we *collect* n_1 i.i.d. samples $(x_1, \ldots, x_{n_1})$ from Q and we *synthesize* n_{-1} i.i.d. samples $(\bar{x}_1, \ldots, \bar{x}_{n_{-1}})$ from μ then we can compute a reliable estimate of $\mathcal{R}$ using

$$\hat{\mathcal{R}}(f) = \frac{1}{(1+\rho)n_1} \sum_{i=1}^{n_1} I(f(x_i) \le 0) + \frac{\rho}{(1+\rho)n_{-1}} \sum_{i=1}^{n_{-1}} I(f(\bar{x}_i) > 0) \tag{3}$$

where $I(\cdot)$ is the indicator function, i.e. $I(\theta) = 1$ when θ is true and $I(\theta) = 0$ when θ is false.

It turns out that $\mathcal{R}$ is also a performance measure for the following (artificial) supervised classification problem. Let $Y := \{1, -1\}$ be the label set and let $x \in X$ and $y \in Y$ denote values of the random variables $\mathbf{x}$ and $\mathbf{y}$. Consider a joint distribution $P_{\mathbf{x},\mathbf{y}}$ where the corresponding conditional distributions are $P_{\mathbf{x}|\mathbf{y}=1} := Q$ and $P_{\mathbf{x}|\mathbf{y}=-1} := \mu$, and the corresponding class marginals are $P(\mathbf{y} = 1) := 1/(1+\rho)$ and $P(\mathbf{y} = -1) := \rho/(1+\rho)$. In the supervised classification problem we seek a real valued function f that minimizes the average classification error $e(f) := E_{P_{\mathbf{x},\mathbf{y}}}[I(\mathrm{sign} f(\mathbf{x}) \neq \mathbf{y})]$. It is easy to show that $\mathcal{R}(f) = e(f)$ and therefore the goal of minimizing the risk is identical to minimizing the average classification error. To create a data set for this (artificial) classification problem we *collect* n_1 i.i.d. samples $(x_1, \ldots, x_{n_1})$ from Q and assign each of them the label $y = +1$, and we *synthesize* n_{-1} i.i.d. samples $(x_{n_1+1}, \ldots, x_{n_1+n_{-1}})$ from μ and assign each of them the label $y = -1$. This gives a data set $\mathcal{T} = ((x_1, y_1), ..., (x_n, y_n))$ of size $n = n_1 + n_{-1}$. In the *learning problem* the goal is to use $\mathcal{T}$ to choose a function $\hat{f}$ so that the average classification error is as small as possible. The empirical average classification error, which is equal to the empirical risk in (3), is given by

$$\sum_{i=1}^{n} u_i I(f(x_i \neq y_i)) \tag{4}$$

where

$$u_i = \begin{cases} \frac{1}{(1+\rho)n_1}, & y_i = 1 \\ \frac{\rho}{(1+\rho)n_{-1}}, & y_i = -1 \end{cases} . \tag{5}$$

The only difference between this learning problem and the standard supervised classification learning problem is that the class marginal probabilities are *known*.

3 Using Empirical Risk Minimization to Design Kernel Machines for Anomaly Detection

Empirical risk minimization (ERM) is a method that determines a detector $\hat{f}$ by minimizing an empirical risk $\hat{R}$. A natural choice for $\hat{R}$ is the empirical risk in (3) (or equivalently (4)), but it is also common to consider other risk functions that are easier to optimize. In this paper we compare two different ERM methods. The first minimizes (3) over a *simple data dependent hypothesis class* called LPC described in Cannon, Howse, Hush, and Scovel (2003) and the second is the *density level detection support vector machine* (DLD–SVM) described in Steinwart et al. (2005). In both cases the detectors f take the form of a *kernel machine*,

$$f_{\gamma,b}(x) = \sum_i \gamma_i K(x_i, x) + b$$

where $K : X \times X \to \mathbb{R}$ is a kernel function, meaning there exists a Hilbert space H and a map $\phi : X \to H$ such that $K(x_1, x_2) = \langle\, \phi(x_1), \phi(x_2) \,\rangle\ \forall\, x_1, x_2 \in X$. In this paper we are interested in problems where X is the space of graphs and therefore K is a so–called *graph kernel.*

4 Graph Kernels

Much of the existing work on graph kernels concentrates on the special cases where the graph is a string, e.g. (Joachims, 2002) and (Lodhi, Shawe-Taylor, Christianini, & Watkins, 2001), or the graph is a tree, e.g. (Vishwanathan & Smola, 2003). Most of the applications for this work are centered around various natural language processing tasks, such as text classification and deducing sentence meaning. Surprisingly little work has been done with more general graphical structures. Most of the work to date on general graph kernels is by Gärtner and Kashima, e.g. see (Gärtner, Flach, & Wrobel, 2003) and (Kashima, Tsuda, & Inokuchi, 2004). In this study we use the work of Gärtner et al. (2003) because it is both straightforward to implement and fairly easy to generalize. In Gärtner et al. (2003) kernels are described which compute certain graph features (*i.e.*, subgraph homomorphism and subgraph isomorphism). The computational complexity is analyzed for each kernel, and it is shown that computing kernels which represent some graph features (*i.e.*, subgraph isomorphism) is NP-hard, while computing kernels which represent other graph features (*i.e.*, subgraph homomorphism) may be more tractable. We now describe the kernel that we chose from Gärtner et al. (2003) for this work.

A *graph* is denoted by $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of n *vertices* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of $m \leq n^2$ *edges*. The *adjacency matrix* $\boldsymbol{E}$ is defined by $[\boldsymbol{E}]_{ij} = 1 \Leftrightarrow (v_i, v_j) \in \mathcal{E}$ and $[\boldsymbol{E}]_{ij} = 0 \Leftrightarrow (v_i, v_j) \notin \mathcal{E}$. For a *vertex label set* $\{\ell_1, ..., \ell_p\}$ the *vertex label matrix* $\boldsymbol{L}$ is defined by $[\boldsymbol{L}]_{ki} = 1 \Leftrightarrow \ell_k = \text{label}(v_i)$ and $[\boldsymbol{L}]_{ki} = 0 \Leftrightarrow \ell_k \neq \text{label}(v_i)$. Clearly $\boldsymbol{E}$ is an $n \times n$ square binary matrix and $\boldsymbol{L}$ is a $p \times n$ rectangular binary matrix where p is the number of vertex labels. The graph kernel that we use computes a weighted sum over walks of length k between vertices labeled ℓ_i and ℓ_j for all possible label pairs (ℓ_i, ℓ_j) and all possible walk lengths k. It is written formally as

$$K(G_1, G_2) = \left\langle \boldsymbol{L}_1 \left(\sum_{i=0}^{\infty} \eta_i \, \boldsymbol{E}_1^i \right) \boldsymbol{L}_1^{\mathsf{T}}, \boldsymbol{L}_2 \left(\sum_{j=0}^{\infty} \eta_j \, \boldsymbol{E}_2^j \right) \boldsymbol{L}_2^{\mathsf{T}} \right\rangle, \tag{6}$$

where the matrix inner product is defined by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i,j} [\boldsymbol{A}]_{ij} \, [\boldsymbol{B}]_{ij}$. At first glance it may appear that this kernel cannot be evaluated in practice because it contains infinite sums. However, if the weight sequence $\eta_0, \eta_1, \ldots$ is properly chosen, then Equation (6) can be evaluated in polynomial time. For example, if $\eta_0, \eta_1, \ldots$ is a geometric sequence, then $\eta_i = \beta^i$ and

$$\sum_{i=0}^{\infty} \beta^i \boldsymbol{E}^i = \left(\boldsymbol{I} - \beta \boldsymbol{E} \right)^{-1} \tag{7}$$

as long as $\beta < 1/\max_{\mathcal{G}}(\min_{G \in \mathcal{G}}(\Delta^+(G), \Delta^-(G)))$, where $\mathcal{G}$ is our collection of graphs, and $\Delta^+(G)$ and $\Delta^-(G)$ are the maximum out-degree and maximum in-degree respectively of the vertices in a particular graph. The right hand side of Equation (7) requires inverting an $n \times n$ matrix, which can be computed in $O(n^3)$ time. Therefore when $\eta_i = \beta^i$, the kernel in Equation (6) can be computed in $O(n^3 + p\,n^2 + p^2\,n)$ time.

5 Experimental Results

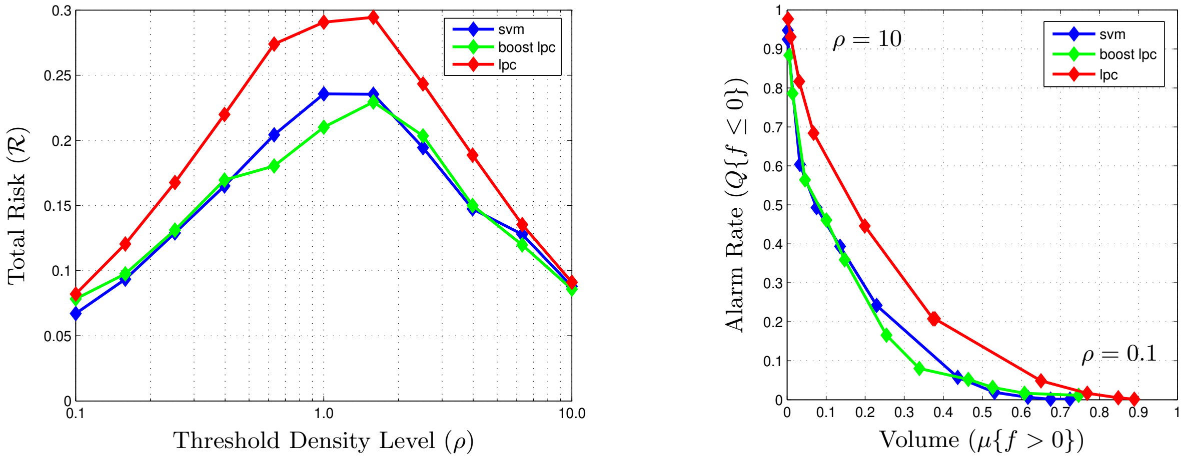
We now describe an experiment where X is the space of all simple undirected graphs having between 2 and 10 vertices with distinct vertex labels. In our example problem, the graphs drawn from the Q distribution represent the interactions of a *group of characters* within a *section of a book*. Vertices in these graphs correspond to a specific character (person) within a group and are labeled by the character's *relative rank* within the group, and an edge is present when the two characters interact within the section of the book represented by the graph. A *group of characters* consists of characters with consecutive rank (*e.g.*, characters with ranks 2 through 11), and a *section of a book* consists of consecutive book chapters (*e.g.*, chapters 16 through 20). Characters are ranked in decreasing order based on their frequency of appearance throughout the *entire* book. So the character with rank one is the character that appears most frequently in the book. Since the vertex label is the character's *relative* rank within a group, the vertex label set is *always* $\{1, \dots, 10\}$. To obtain samples from the distribution Q, graphs were generated from the following books: Anna Karenina, David Copperfield, The Iliad, Huckleberry Finn, and Les Misérables.

To generate samples from the reference distribution μ, we first considered the choice of reference distribution. When $X \subset \mathbb{R}^d$ it is common to choose the uniform distribution for μ, but a uniform distribution may be a poor choice for our graph space. Since there are far more graphs with 10 vertices than graphs with less than 10 vertices, a uniform distribution will be highly concentrated around graphs with 10 vertices. Thus for any non–uniform distribution Q, such as the distribution of our book data, it is likely that almost all graphs containing less than 10 vertices will be labeled anomalous. Thus choosing a uniform reference distribution is likely to produce a very uninformative solution. Therefore, for this study we choose a non-uniform reference distribution. Our reference data consists of samples from X containing $2 \leq n \leq 10$ vertices and $1 \leq m \leq n(n-1)/2$ edges. The probability of n is uniform on $\{2, \dots, 10\}$, and the probability of m is uniform on $\{1, \dots, n(n-1)/2\}$. The edge locations are determined by drawing m samples without replacement from the set of $\binom{n}{2}$ possible locations. The vertex labels are determined by drawing n samples without replacement from the set $\{1, \dots, 10\}$. Note that there are many other ways to generate reference data for this problem.

For our anomaly detection experiments we generated 3170 book graphs from Q and 30,000 random graphs from μ. All of the data and software needed to generate these graphs is in a package called *GraphBase* by Knuth (1994). We split this graph data into three disjoint sets: training $\mathcal{T}$, validation $\mathcal{A}$, and testing $\mathcal{S}$. The set $\mathcal{T}$ contains 1600 book graphs and 10,000 reference graphs, and the sets $\mathcal{A}$ and $\mathcal{S}$ each contain 785 book graphs and 10,000 reference graphs. We used three different algorithms to design anomaly detectors with this data, the DLD–SVM, the DLD–LPC and the boosted DLD–LPC [1]. We constructed the detector $\hat{f}$ using the training set $\mathcal{T}$. Tuning parameters, such as β for the kernel and λ for the SVM, were chosen to (approximately) minimize the empirical risk in (4) on the validation set $\mathcal{A}$. Note that for simple undirected graphs containing at most 10 vertices $\beta < \frac{1}{9}$. Estimates of the generalization performance (*i.e.*, future performance) were obtained using the testing set $\mathcal{S}$. We estimate three different performance measures, the total risk in Equation (2), the alarm rate $Q(f \leq 0)$, and the volume $\mu(f > 0)$.

The estimates of these performance measures for the DLD–SVM, DLD–LPC and boosted DLD–LPC anomaly detectors are shown in Figure 1. Figure 1(a) plots the estimated total risk $\mathcal{R}$ versus the threshold density level ρ for the three anomaly detectors. Notice that the SVM significantly outperforms the single LPC, but that the boosted LPC has comparable performance to the SVM. Also note that the minimum total risk in this plot is about 6%, which suggests that the conditional distributions $P_{\mathbf{x}|\mathbf{y}=1}$ and $P_{\mathbf{x}|\mathbf{y}=-1}$ have a fair amount of overlap. Figure 1(b) plots the alarm rate $Q\{f \leq 0\}$ versus the volume $\mu\{f > 0\}$. Again notice that the SVM and the boosted LPC have similar performance and the single LPC is

[1] For the boosted DLD–LPC we used the standard AdaBoost algorithm with base classifiers designed using the DLD–LPC algorithm.

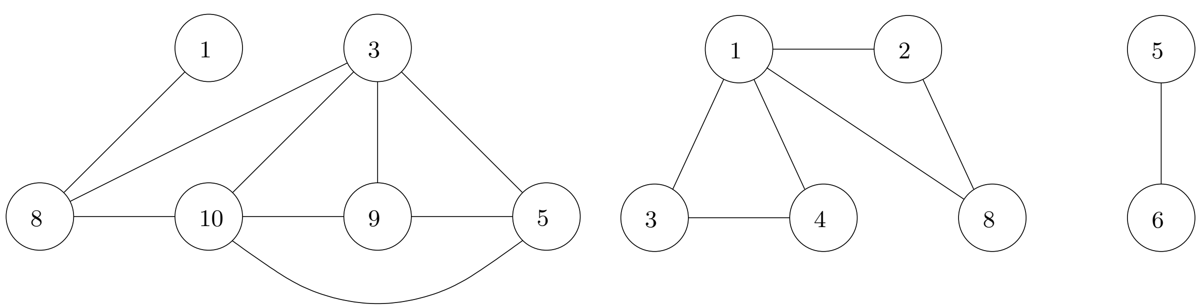


(a) (b)

Figure 1: Performance measures estimated using the testing set $\mathcal{S}$ for the DLD–SVM, DLD–LPC and boosted DLD–LPC anomaly detectors. (a) The total risk R given by Equation (2) versus the threshold density level ρ. (b) The alarm rate $Q\{f \leq 0\}$ versus the volume $\mu\{f > 0\}$.

noticeably worse. This plot is implicitly a function of ρ, with the points in the lower right corresponding to small values of ρ and those in the upper left corresponding to large ρ. The curves in Figure 1(b) clearly illustrate the trade-off between alarm rate and volume. Ideally one would like to operate at a point on these curves which has both a small alarm rate and a small volume. The point $\rho \approx 1$ is near the knee of the curves and therefore appears to give a reasonable trade-off between small alarm rate and small volume for all three detectors. Note that $\rho \approx 1$ is near the maximum total error for the three curves in Figure 1(a).

Finally we present two examples of book graphs, one was labeled anomalous by the boosted DLD-LPC detector and the other was labeled non-anomalous. The two graphs are shown in Figure 2. The graph in Figure 2(a) was labeled anomalous and the one in Figure 2(b) was



(a) (b)

Figure 2: Two examples of book graphs, (a) was labeled as anomalous, and (b) was labeled as non-anomalous by the boosted DLD–LPC detector.

labeled non-anomalous by the boosted DLD–LPC detector. The graph in Figure 2(a) comes from "David Copperfield" chapters 3–8 and characters 5–14, and the one in Figure 2(b) comes from "Les Misérables" chapters 310–345 and characters 2–11. This characterization of these two graphs appears reasonable in the following sense. In the non-anomalous graph in Figure 2(b) most of the connections go from high ranked characters to low ranked characters, and there is very little interaction between low ranked characters. However, in the anomalous graph in Figure 2(a) this situation is essentially reversed. Note that the non-anomalous graph has two distinct components (*i.e.*, it is not fully connected), so apparently the number of components does not strongly influence whether a graph is labeled as anomalous.

6 Conclusion

We have described a recent discovery that allows us to make quantitative comparisons of different anomaly detection solutions using estimates of the total risk in Equation (2). Furthermore we have applied ERM learning algorithms from supervised classification to design anomaly detectors. Our formalism allows us to make an explicit choice for the reference measure μ, which appears to be very important in graph spaces. We have also described solution methods that perform anomaly detection directly in graph space using the graph kernel in Equation (6). Our methods solve the density level detection problem *directly* (in contrast to indirect methods based on e.g. density estimation techniques which may be more difficult or produce inferior results). Our methods have both proven performance bounds and guaranteed computational efficiency. We have demonstrated these methods on a synthetic problem where the graphs represent the interactions between characters in novels.

References

Ben-David, S., & Lindenbaum, M. (1997). Learning distributions by their density levels: A paradigm for learning without a teacher. *J. Comput. System Sci.*, *55*, 171–182.

Cannon, A., Howse, J., Hush, D., & Scovel, C. (2003). *Simple classifiers* (LANL Technical Report LA-UR-03-0193). Los Alamos, NM: Los Alamos National Laboratory.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf & M. Warmuth (Eds.), *Proceedings of the 16th annual conference on computational learning theory* (Vol. 2777, pp. 129–143). Berlin, Germany: Springer-Verlag.

Joachims, T. (2002). *Learning to classify text using support vector machines* (Vol. 668). Berlin, Germany: Springer-Verlag.

Kashima, H., Tsuda, K., & Inokuchi, A. (2004). Kernels for graphs. In B. Schölkopf, K. Tsuda, & J. Vert (Eds.), *Kernel methods in computational biology* (pp. 155–170). Cambridge, MA: MIT Press.

Knuth, D. (1994). *The Stanford GraphBase.* Reading, MA: Addison-Wesley Publishing Co.

Lodhi, H., Shawe-Taylor, J., Christianini, N., & Watkins, C. (2001). Text classification using string kernels. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems* (Vol. 13, pp. 563–569). Cambridge, MA: MIT Press.

Steinwart, I., Hush, D., & Scovel, C. (2005). A classification framework for anomaly detection. *Journal of Machine Learning Research*, *6*, 211–232.

Vishwanathan, S., & Smola, A. (2003). Fast kernels for string and tree matching. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15). Cambridge, MA: MIT Press.