A kernel method for unsupervised structured network inference

Christoph Lippert
christoph.lippert
@tuebingen.mpg.de
MPI for Biol. Cybernetics
MPI for Dev. Biology
Tübingen, Germany

Oliver Stegle
os252@cam.ac.uk
Cavendish Laboratory
University of Cambridge
Cambridge, UK

Zoubin Ghahramani
zoubin@eng.cam.ac.uk
Department of Engineering
University of Cambridge
Cambridge, UK

Karsten M. Borgwardt
karsten.borgwardt
@tuebingen.mpg.de
MPI for Biol. Cybernetics
MPI for Dev. Biology
Tübingen, Germany

Abstract

Network inference is the problem of inferring edges between a set of real-world objects, for instance, interactions between pairs of proteins in bioinformatics. Current kernel-based approaches to this problem share a set of common features: (i) they are supervised and hence require labeled training data; (ii) edges in the network are treated as mutually independent and hence topological properties are largely ignored; (iii) they lack a statistical interpretation. We argue that these common assumptions are often undesirable for network inference, and propose (i) an unsupervised kernel method (ii) that takes the global structure of the network into account and (iii) is statistically motivated. We show that our approach can explain commonly used heuristics in statistical terms. In experiments on social networks, different variants of our method demonstrate appealing predictive performance.

1 Introduction

Graphs are the data structure of choice for modeling objects and their relationships. Applications span a large range from bioinformatics, systems biology, chemoinformatics to social network analysis and Internet studies. Here, graphs are used to model protein structures, cellular networks, chemical compounds, groups of individuals or groups of websites.

In these applications the structure of graphs captures different aspects. On the one hand, if the graph represents a *graphical model*, it describes a probabilistic model of the data, with nodes corresponding to random variables and the structure of the graph encoding their statistical dependence. On the other hand, the structure of the graph can correspond to a set of (physical) interactions between real-world objects represented by the nodes. These could be individuals in a social network who are communicating or proteins in a regular network that are interacting. In this latter case, a graph is often referred to as a *network*.

In this article, we are concerned with networks rather than graphical models. In particular, we are interested in learning problems where the task is to infer the structure, i.e. the edges, of a network from the set of nodes and their attributes. This problem is of great relevance in many fields. For example in bioinformatics we may want to predict the interactions between proteins solely based on a set of attributes.

We focus on kernel methods for network inference. They provide a unified framework for handling the variety of data types that we encounter as node attributes, such as vectors, strings or time series, by mapping them into a feature space. Previous kernel approaches to network inference exhibit at least one of the following three properties (Ben-Hur & Noble, 2005; Vert et al., 2007):

- They are *supervised*.
- They assume the likelihood of individual edges to be *independent*, and hence do not consider the global structure of the inferred graph and its topological properties.
- This independence assumption is *heuristic* and the network strategy it implies has no clear statistical interpretation.

We feel that these three characteristics are not necessarily beneficial for network inference:
First, ground truth network data from bioinformatics and other application domains are often hard to

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

obtain, noisy, or even contradictory (Jansen et al., 2003), For example the experimental determination of protein-protein interactions is expensive and prone to false positives. Similarly gathering information about links in social networks is a tedious and time-consuming task. Often, detailed data and information about the nodes in a network (for instance, protein sequence, structure, physical features) are available long before a reliable set of interactions is established. When such ground truth network data is available, supervised network inference has been shown to be superior to unsupervised approaches (Ben-Hur & Noble, 2005; Vert et al., 2007). But unsupervised approaches to network inference can indeed be appealing, when such reliable training data is not available for protein-protein interaction prediction in a specific species.

Second, the independence assumption of the existence of edges in a graph contradicts the nature of most real-world networks. Graphs commonly exhibit a particular global structure and topology, such as being small-world or scale-free (Barabási & Albert, 1999). Often these global properties of the network are reflected by node attributes in the graph, and it is desirable to exploit this correlation in network inference. For example, measurements on the essentiality of a protein — that is whether it is indispensable to the survival or reproduction of the organism — have been found to be correlated with its number of interaction partners, *i.e.* its degree in the interaction network (Jeong et al., 2001). Ignoring such information seems wasteful.

Third, as edges are treated as independent events, current methods predict edges solely based on the attributes of two nodes and ignore the rest of the graph; typically, the nodes which score highest according to a (dis-)similarity measure are predicted to be connected. From a theoretical point of view, it is unsatisfying that we do not understand which statistical criterion is optimized when following this strategy of drawing edges between the most (dis-)similar pairs of nodes.

In contrast to kernel methods, the statistical relational learning community employs graphical models for network inference and Bayesian methods such as Gaussian processes for link (edge) prediction ((Getoor & Taskar, 2007) and references therein). Although these methods have a clear statistical interpretation, they also tend to be based on supervised learning and independent link predictions.

In the following we will try to overcome these limitations. This article is structured as follows:

- In Section 2.2 we define an *unsupervised* kernel-framework for network inference based on statistical dependence maximization. We show that existing heuristic approaches to network inference can be *interpreted* as special cases of this statistical learning framework (see Section 3.1).
- In Section 3.2 we extend this class of existing approaches by introducing a new family of network inference algorithms that do not make assumptions about the independence of edges between nodes. Rather by considering non-local graph kernels such as based on node degrees, we can *relate complex network properties to node attributes*.
- In Section 4 we demonstrate the *practical usefulness* of our new approach in applications to social network analysis.

2 Network inference using HSIC

The tool at the core of our kernel method for unsupervised network inference is the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005), and the feature selection method BAHSIC (Song et al., 2007) building on this criterion. We first briefly review HSIC for feature selection and then explain how it can be adapted to the problem of network inference.

2.1 Hilbert-Schmidt Independence Criterion

Intuitively, the Hilbert Schmidt Independence Criterion (HSIC) is based on the concept that two random variables x and y are independent iff all functions $f(x)$ and $g(y)$ from 'large enough' function classes $\mathcal{F}$ and $\mathcal{G}$ are uncorrelated. The attractiveness of HSIC stems from the fact that an empirical estimate of the criterion can be expressed purely in terms of kernels.

In more detail, let sets of pairs of observations from two random variables, $X \sim x$ and $Y \sim y$, be drawn from a joint distribution $\Pr_{xy}$: $(X, Y) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. The Hilbert-Schmidt Independence Criterion (Gretton et al., 2005) measures the dependence between the two random variables, x and y, based on the empirical sample distribution of pairs from X and Y.

Let $\mathcal{F}$ and $\mathcal{G}$ be the reproducing kernel Hilbert Spaces (RKHS) on $\mathcal{X}$ and $\mathcal{Y}$ with associated kernels $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ respectively, and associated mappings $\phi : \mathcal{X} \to \mathcal{F}$ and $\psi : \mathcal{Y} \to \mathcal{G}$ to the feature spaces. The cross-covariance operator $\mathcal{C}_{xy} : \mathcal{G} \mapsto \mathcal{F}$ is defined as (Fukumizu et al., 2004)

$$\mathcal{C}_{xy} = \mathbb{E}_{xy}\left[(\phi(x) - \mu_x) \otimes (\psi(y) - \mu_y)\right], \qquad (1)$$

where $\mu_x = \mathbb{E}[\phi(x)]$ and $\mu_y = \mathbb{E}[\psi(y)]$. HSIC is then defined as the square of the Hilbert-Schmidt norm[1] of $\mathcal{C}_{xy}$, that is $\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, \Pr_{xy}) := \|\mathcal{C}_{xy}\|^2_{\mathrm{HS}}$.

[1] For a finite-dimensional matrix A, $\|A\|_{\mathrm{HS}} = \sqrt{\sum_{i,j} A^2_{i,j}}$, which is known as the Frobenius norm.

An empirical estimate of HSIC in terms of kernels on X and Y can be computed as (Gretton et al., 2005)

$$\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, Z) = (n-1)^{-2} \operatorname{tr} KHLH, \qquad (2)$$

where $K_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ and $L_{ij} = l(y_i, y_j) = \langle \psi(y_i), \psi(y_j) \rangle$ are kernel matrices on X and Y respectively, and $H_{ij} = \delta_{ij} - n^{-1}$ centers the kernel matrices in feature space. For a particular class of kernels, so-called *universal* kernels (Steinwart, 2002), HSIC can be shown to equal zero iff x and y are independent. In general, the larger HSIC, the larger the dependence between x and y. This property is exploited in BAHSIC for feature selection. In greedy forward selection, features are added to maximize the dependence between y (the class labels) and x (the data objects). We adapt this idea to network inference and refer to the resulting algorithm as NETHSIC.

2.2 NETHSIC

Let us now turn our attention to the question of how HSIC can be used as a tool for network inference. We are given the nodes V of a graph G and their attributes, but not its edges configuration E. In this case, X and Y have a different meaning than in feature selection: both represent the set of nodes in our network. $x_i \in X$ describes the attributes of node i in the network, y_i — intuitively speaking — its location within the network. $K_{ij} = k(x_i, x_j)$ is then a kernel (matrix) on the node attributes, which we refer to as the *attribute kernel*. $L_{ij} = l(y_i, y_j)$ is a kernel (matrix) on the locations of the nodes in the graph, which we call the *node kernel*. It will become clearer what is meant by 'location in the graph' in the following.

The location of a node within a graph is defined by its neighbors or other topological features. Hence the kernel l on these locations of nodes depends on the set of edges E of the graph. We make this dependence on E explicit by denoting the node kernel matrix by L_E.

After defining K and L_E, the network inference problem via HSIC (NETHSIC) can now be cast in the following form:

$$\underset{E \subset (V \times V) \wedge |E| = m}{\operatorname{argmax}} \frac{1}{(n-1)^2} \operatorname{tr} KHL_E H, \qquad (3)$$

where m is the number of edges of the graph that is to be learnt. In other words, we maximize over edge-configurations E, such that the attributes of the nodes and their locations in the graph maximally depend on each other (as determined by HSIC).

By maximizing the dependence between the attributes and the locations of nodes, we implicitly infer the set of edges E (because it defines the locations of the nodes). How the location of a node in a graph is defined depends on the node kernel l — we will elaborate on this point in the following sections, as our ability to learn complex graph structure relies on the freedom to pick l. Different choices for the node kernel l result in different solution to (3), and therefore correspond to different network structures.

Input: The set of nodes V,
number of edges m,
attribute kernel k and node kernel l
Output: A subset E of $V \times V$ of size m
$E \leftarrow \emptyset$
repeat
  $e = \arg\max_{e' \in V \times V} \operatorname{tr} KHL_{E \cup \{e'\}} H$
  $E \leftarrow E \cup \{e\}$
until $|E| = m$

Algorithm 1: NETHSIC forward selection

Search strategy A naive approach to solve problem (3) is exhaustive enumeration of all possible sets of $V \times V$ of size m, computing the objective function in (3) for each set. This would require an effort exponential in the size of the graph, n, however. To avoid exponential runtime, we employ greedy optimization strategies (Song et al., 2007). One approach is forward selection of edges (see Algorithm 1). Starting with a completely unconnected graph we iteratively add edges that increases HSIC the most, until the desired number of m target edges is reached. When inferring a sparse network, the number of edges m will be $m << n^2$ and often even $m \in O(n)$. The runtime of the resulting algorithm is $O(n^2 m R(L))$, where $R(L)$ is the effort of recomputing L, is appealing and renders medium scale graphs with hundreds of nodes tractable. For smaller scale graphs it might be desired to perform backward feature selection. It has been shown that backward elimination can lead to more accurate answers (Song et al., 2007), however at considerably higher computational cost scaling as $O(n^2(n^2 - m)R(L))$. Note that depending on the employed node kernel NETHSIC can be even faster. For instance on kernels where single edge additions or removals have only local effects, parts of the kernel can be cached. This principle allows the degree kernel to be evaluated in $O(n^2 m)$, which will be applied in our experiments (see Section 4).

Alternative search strategies are also possible. We use greedy selection, because it yields the optimal solution for certain kernels, and it generates a ranking of edges in a graph that an be used as an evaluation criterion in experiments. This ranking can be assessed by comparison to a reference network yielding an ROC curve. However particular for global graph kernels alternative search strategies might be an interesting consideration. One may think of include, for instance, a random walk or Simulated Annealing or iterations of

adding and deleting edges during the greedy search.

Objective of greedy selection Assume k is a kernel on node attributes, giving rise to a kernel matrix K. Let $\widetilde{K} = HKH$ be the centered version of K. Then since $\operatorname{tr} KHL_EH = \operatorname{tr} HKHL_E = \operatorname{tr} \widetilde{K}L_E$ in each iteration of greedy NETHSIC we maximize

$$\operatorname*{argmax}_{e' \in V \times V} \operatorname{tr}(\widetilde{K}L_{E\cup\{e'\}}) - \operatorname{tr}(\widetilde{K}L_E) = \tag{4}$$
$$\operatorname*{argmax}_{e' \in V \times V} \sum_{i=1}^{n}\sum_{j=1}^{n} \widetilde{K}(i,j)\left(L_{E\cup\{e'\}}(i,j) - L_E(i,j)\right)$$

over possible edges $e' \in V \times V$. The node attributes are fixed and hence $\widetilde{K}$ is unchanged whatever E currently looks like. To solve problem (4), we hence have to look only at those entries (i,j) of $L_E(i,j)$ and $L_{E\cup\{e\}}(i,j)$ that are not identical, or in other terms: at those pairs of nodes i and j whose kernel value $l(i,j)$ changes when we remove edge e.

How to determine m, the number edges A key difference between NETHSIC and the common approach to graph inference — which draws an edge between nodes whose similarity exceeds a threshold θ — is the setting of parameters. While in direct approaches the number of inferred edges is implicitly defined by means of the threshold value θ, NETHSIC requires the number of edges m as a pre-specified parameter.

If this property is undesired, it is also possible to determine m by means of statistical significance testing. This procedure is based on idea recently proposed in (Gretton et al., 2008): By randomly permuting the attributes of the nodes and recomputing the kernel matrix $\tilde{K}$ we can generate an artificial false dataset. Dependence estimations between any such randomized $\tilde{K}$ and L_E are purely random and can be used to calculate a p-value for the dependence test. This approach has the advantage that we only have to choose a statistical significance level (usually 0.05 or 0.01) rather than an explicit number of edges that we wish to infer.

3 Special instances of NETHSIC

Network inference via NETHSIC requires two kernel functions: an attribute kernel k and a node kernel l. The choice of l allows us to influence the structure of the network that we aim to learn. Here we present a selection of node kernels that possess interesting properties for network inference.

3.1 1-step random walks

The (unnormalized) Laplacian of a graph G is defined as $\mathcal{L} = D - A$ where A is the adjacency matrix of G, and $D(i,i) = \sum_j A(i,j)$. A p-step random walk on the normalized graph Laplacian $\tilde{\mathcal{L}} = D^{-\frac{1}{2}}\mathcal{L}D^{-\frac{1}{2}}$ gives rise to a kernel on nodes $L = (aI - \tilde{\mathcal{L}})^p$, where $a \geq 2$ and $p \in \mathbb{N}$ and I is the identity matrix of size $n \times n$ (Smola & Kondor, 2003). The eigenspectrum of $\tilde{\mathcal{L}}$ is upper-bounded by 2 and lower-bounded by 0, and positive semi-definiteness is hence guaranteed by choosing $a \geq 2$. It will turn out to be useful to deal with 1-step random walks on the *unnormalized* graph Laplacian. A p-step random walk on the unnormalized graph Laplacian can be defined analogously as $L = (aI - \mathcal{L})^p$, however for odd choices of p, $a \geq 2$ does not guarantee positive semi-definiteness. For a 1-step random walk via $(aI - \mathcal{L})$ we therefore have to set $a \geq 2n - 2$ to guarantee positive semi-definiteness, as stated by Theorem 1:

Theorem 1 *Let $\mathcal{L}$ be the unnormalized graph Laplacian on graph G of size n. Then $(aI - \mathcal{L})$ is positive semi-definite if $a \geq 2n - 2$.*

Proof As $\mathcal{L}$ is positive semi-definite, its eigenvalues $\lambda_i[\mathcal{L}]$ are non-negative. It is a well-known fact from linear algebra that the eigenvalues of $(\mathcal{L} - aI)$ are $\lambda_i[\mathcal{L} - aI] = \lambda_i[\mathcal{L}] - a$, and the eigenvalues of $(aI - \mathcal{L})$ are $\lambda_i[aI - \mathcal{L}] = a - \lambda_i[\mathcal{L}]$. As a consequence, the eigenvalues of $(aI - \mathcal{L})$ are nonnegative iff $a \geq \lambda_{max}[\mathcal{L}]$. It follows from Gershgorin's circle theorem, however, that $\lambda_{max}[\mathcal{L}] \leq 2n - 2$ for a graph of size n. Hence $a \geq 2n - 2$ guarantees the positive semi-definiteness of $(aI - \mathcal{L})$. ■

Let us now assume our kernel matrix L_E is $aI - \mathcal{L}$ on our graph $G = (V, E)$, where $a \geq 2n-2$. When we add edge $e = (i,j)$ to E, the following entries change in L_E: $L_{E\cup\{e\}}(i,i) = L_E(i,i) - 1$; $L_{E\cup\{e\}}(j,j) = L_E(j,j) - 1$; $L_{E\cup\{e\}}(i,j) = L_E(i,j) + 1$; $L_{E\cup\{e\}}(j,i) = L_E(j,i) + 1$. All other entries remain unchanged. Hence we can rewrite (4) as

$$\operatorname*{argmax}_{e'=(i,j)} -\widetilde{K}(i,i) - \widetilde{K}(j,j) + 2\widetilde{K}(i,j) =$$
$$\operatorname*{argmax}_{e'=(i,j)} -\widetilde{d}(i,j)^2 = \operatorname*{argmin}_{e=(i,j)} \widetilde{d}(i,j)^2, \tag{5}$$

where $\widetilde{d}$ is the distance induced by kernel $\widetilde{K}$. Hence the edge $e = (i,j)$ that maximizes the objective in (4) is the one for which the distance between i and j is minimal according to $\widetilde{K}$ — this is the edge we add to E in each iteration of NETHSIC.

Theorem 2 *Greedy network inference using NETHSIC on $L = aI - \mathcal{L}$ finds the optimal solution to problem (3).*

Proof For this kernel, problem (3) is equivalent to $\text{argmin}_{\mathcal{I}} \sum_{(i,j)\in\mathcal{I}\subset V\times V} \widetilde{d}(i,j)^2$, which decomposes into an optimization problem over individual edges. By greedily selecting the m edges whose nodes are closest to each other (distance induced by $\widetilde{K}$), we obtain the optimal solution to (3). ■

As a consequence NETHSIC with 1-step random walks is exactly the same as the common ad-hoc approach to graph inference, which is to compute a distance function d between pairs of nodes and to then threshold these distances compared to some score: If and only if the distance $d(i,j) < \theta$, an edge is predicted to exist between nodes i and j. While the statistical foundations of this strategy were somehow obscure so far, NETHSIC now allows us to interpret this strategy:

Lemma 3 *By predicting edges between nodes whose attributes are most similar, one maximizes the dependence, as measured by the Hilbert-Schmidt Independence Criterion, between the node attributes and a 1-step random walk on the predicted graph.*

As a direct consequence, NETHSIC with a 1-step random walk kernel on the unnormalized graph Laplacian is optimally solvable in $O(n^2 \log n)$, which is the cost of computing the kernel matrices L and K and sorting their entries.

We note that the exact opposite strategy — connecting nodes with most *dissimilar* attributes — is equivalent to choosing the Laplacian of a graph as the node kernel in NETHSIC. Analogous to the 1-step random walk, greedy selection is again optimal and the runtime is $O(n^2 \log n)$ (proofs omitted). This kernel is useful when inferring a network in which nodes with dissimilar roles (and hence attributes) tend to interact: for instance, professor/student or positively/negatively charged molecules. Our experiments on social networks show that such a dissimilar kernel indeed often describes social relationships (Section 4).

3.2 Node kernels with *global* effects

The two kernels we have presented so far possess the property that (3) decomposes into an optimization problem over individual edges. We refer to such a kernel as a kernel with *local effects*. In general, if adding the edge (i,j) affects kernel values other than $L(i,j)$, $L(j,i)$, $L(i,i)$, or $L(j,j)$, then we observe more than local effects. Indeed, the selection of an edge then has a *global effect*, *i.e.* it changes the similarities between other pairs of nodes in the graph, not just i and j. While kernels with global effects make it more difficult to solve (3), as we have to update L in each iteration of NETHSIC, they also exhibit a desirable property, namely that the existence of edges are not independent events any more. Hence by choosing a node kernel with global effects, we make NETHSIC learn graphs with interdependencies between edges.

As a first example, let our kernel matrix L_E be the squared adjacency matrix A^2 of our graph $G = (V, E)$. Let $\mathcal{N}(i)$ denote the neighbours of node i in graph G. Intuitively, this kernel on nodes counts the number of common neighbours of two nodes i and j: $A^2(i,j) = |\mathcal{N}(i) \cap \mathcal{N}(j)|$.

When we add edge $e = (i,j)$ to E, the following entries change in L_E:
$\forall k \in \mathcal{N}(j) : L_{E\cup\{e\}}(k,i) = L_E(k,i) + 1;$
$\forall k \in \mathcal{N}(j) : L_{E\cup\{e\}}(i,k) = L_E(i,k) + 1;$
$\forall k \in \mathcal{N}(i) : L_{E\cup\{e\}}(k,j) = L_E(k,j) + 1;$
$\forall k \in \mathcal{N}(i) : L_{E\cup\{e\}}(j,k) = L_E(j,k) + 1;$
$L_{E\cup\{e\}}(i,i) = L_E(i,i) + 1;$
$L_{E\cup\{e\}}(j,j) = L_E(j,j) + 1.$

The intuitive interpretation of these changes is that by adding edge (i,j), we create walks of length 2 from node i via j to the neighbours of j and from node j via i to the neighbours of i.

All other entries remain unchanged. Hence the greedy optimization criteria (4) can be rewritten as

$$\underset{e=(i,j)}{\text{argmax}}[2\sum_{k\in\mathcal{N}(j)} \widetilde{K}(i,k) + 2\sum_{k\in\mathcal{N}(i)} \widetilde{K}(j,k)] + \widetilde{K}(i,i) + \widetilde{K}(j,j).$$

This adds edges (i,j) where i is similar to the neighbourhood of j, and j is similar to the neighbourhood of i, whereas the direct similarity of (i,j) is ignored. Greedy selection is not optimal in this case, as the neighbourhoods of nodes change in consecutive iterations of NETHSIC and the objective above cannot be evaluated for each edge irrespective of the existence of other edges.

Other examples include kernels defined on the normalized graph Laplacian (see Equations (17)–(20) in (Smola & Kondor, 2003)). For this kernel $L(i,j)$ depends on the degree of node i and node j, hence all kernel values $L(i,k)$ between i and $k \in \mathcal{N}(i)$, and $L(j,k)$ between j and $k \in \mathcal{N}(j)$ change (and correspondingly $L(k,i)$ and $L(k,j)$), when removing edge (i,j).

A simple, yet useful class of node kernels with global effects can be defined by looking at classic topological descriptors: These are scalars or vectors describing some topological property of a node in a graph. An associated node kernel can easily be derived, for instance, in terms of a linear kernel on the topological properties of two nodes. In our experiments, we will use linear kernels on four such topological properties of nodes: *node degree*, *closeness centrality*, *betweenness centrality*, and *shortest path vectors*[2].

[2]We provide the definitions of these topological properties for *connected* graphs here.

Degree As commonly known, the degree δ of a node i in an undirected graph G is defined as $\delta(i) = |\mathcal{N}(i)|$. We define an associated kernel k_δ between vertices i and j as $k_\delta(i,j) = \langle \delta(i), \delta(j) \rangle$, which is referred to as `degr` in our experiments.

Closeness centrality is defined as the mean shortest path length between a vertex i and all other vertices reachable from it: $C_C(i) = (n-1)^{-1} \sum_{t \in V \setminus \{i\}} d_G(i,t)$ where d_G is the shortest path length between all pairs of nodes in G. We define an associated kernel k_{C_C} between vertices i and j as $k_{C_C}(i,j) = \langle C_C(i), C_C(j) \rangle$, called `close` in our experiments.

Betweenness centrality is defined as the number of shortest paths passing through node i: $C_B(i) = \sum_{\substack{s \neq i \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(i)}{\sigma_{st}}$ where σ_{st} is the number of shortest paths from node s to t, and $\sigma_{st}(i)$ is the number of shortest paths from s to t passing through vertex i. We define an associated kernel k_{C_B} between vertices i and j as $k_{C_B}(i,j) = \langle C_B(i), C_B(j) \rangle$, which is referred to as `betw` in our experiments.

The **shortest path vector** $s(i)$ of size $1 \times n$ of a node i is simply its row in the matrix of shortest path distances: $s(i) = [d_G(i,v_1), \ldots, d_G(i,v_n)]$, where $V = \{v_1, \ldots, v_n\}$. We define an associated kernel k_s between vertices i and j as $k_s(i,j) = \langle s(i), s(j) \rangle$, called `sp` in our experiments.

4 Experiments and discussion

In this section, we apply NETHSIC to social network analysis. We compare to three unsupervised network inference methods: correlation relevance networks (Butte et al., 2000), mutual information relevance networks (Butte & Kohane, 2000), and graphical Gaussian models (Toh & Horimoto, 2002).

4.1 Social network analysis

For this evaluation we obtained four social network data sets: Krackhardt's High-tech managers data set, Padgett's Florentine Families, Freeman's EIES network, and Countries trade data set (Wasserman & Faust, 1994). The Countries data set contains information on 24 countries, comprising 4 numeric attributes: population size, the GNP per capita, the average education level (as expressed by scalars), as well as energy consumption. We used NETHSIC to learn the network of trade relationships between these countries, based on these 4 numeric node attributes.

For experiments, NETHSIC requires:

- An *attribute kernel*: for each data set we computed $n \times n$ kernel matrices for all of the numeric attributes, using a linear kernel.
- A *node kernel*: we ran NETHSIC with eight different node kernels: the graph Laplacian, the 1-step random walk on the unnormalized graph Laplacian (Section 3.1), the squared adjacency matrix, a linear kernel on shortest path vectors, closeness, degrees and betweenness of the nodes (all from Section 3.2), and a Δ-kernel on the degrees of the nodes.
- The *number of desired edges* m: We set $m = \frac{1}{2}(n^2 - n)$, where n is the number of nodes in the respective graph, such that all possible edges are consecutively added to the graph. This approach yields a ranking over all edges, namely the inverse order in which they were added to the graph.

To evaluate the quality of NETHSIC's output we need:

- A gold reference network that we can compare to. The Countries trade data includes five such reference networks, the network of trade of 1) food, 2) crude materials, 3) mineral fuels, 4) manufactured goods, and 5) exchange of diplomats (Wasserman & Faust, 1994) .
- A strategy for comparing NETHSIC's output to the gold reference network: To compare the two, we transformed the gold reference network into a ranking of pairs of nodes. All pairs of nodes that are connected by an edge are ranked higher than those that are not connected. In this way, we obtain a ranking of edges from NETHSIC and from the gold reference. The quality of the NETHSIC output can be measured in terms of an ROC curve and AUC value (where we treat the gold reference as the true "class labels" and NETHSIC's ranking as the predictions).

We ran NETHSIC (in its forward selection variant) for all combinations of attributes in each dataset and the eight node kernels, and compared the inferred network to each of the reference networks. For comparison we applied three unsupervised network inference methods from the literature: Mutual information relevance networks connect the nodes with a high pairwise mutual information (Butte & Kohane, 2000). Correlation relevance networks draw edges between nodes that have correlated features (Butte et al., 2000). Graphical Gaussian models are similar to relevance networks but account for indirect or partial correlation due to existing edges (Toh & Horimoto, 2002). As both correlation relevance networks and Graphical Gaussian models rely on correlations and partial correlations between the attributes, respectively, these methods do not provide meaningful results for a single node attribute. For this reason, we also ran all the methods on a combination of all attributes of each data set. For NETHSIC we chose an attribute kernel which is a sum over linear kernels on the individual attributes.

node attribute	**reference network**	**NETHSIC with node kernel**								**method**			**random**
input	**output reference**	`lapl`	`1step`	`sp`	`betw`	`close`	`adj`	`degr`	`Δdegr`	GGM	corr	mi	$\alpha = 5\%$
		Krackhardt's High-tech managers (21 nodes)											
age	advise relation	41.5%	**58.8%**	**57.6%**	47.6%	**59.3%**	42.6%	49.9%	**62.2%**	—	—	**57.6%**	57.1%
age	friendship	42.7%	**57.1%**	**57.7%**	55.0%	52.2%	43.4%	39.9%	47.7%	—	—	51.2%	57.0%
age	who reports whom	52.7%	47.4%	42.9%	54.5%	44.9%	47.9%	57.5%	58.5%	—	—	52.4%	61.2%
tenure	advise relation	43.5%	56.6%	51.8%	45.9%	55.5%	43.6%	54.5%	**59.7%**	—	—	55.3%	57.1%
tenure	friendship	49.7%	50.3%	46.2%	**58.0%**	48.6%	51.3%	56.1%	55.9%	—	—	51.2%	57.0%
tenure	who reports whom	51.4%	48.6%	46.5%	48.3%	44.7%	52.1%	61.7%	59.0%	—	—	48.0%	61.2%
level	advise relation	50.5%	46.8%	49.2%	45.0%	40.6%	49.1%	**60.9%**	47.9%	—	—	**57.5%**	57.1%
level	friendship	40.8%	**61.0%**	56.5%	45.1%	49.7%	39.1%	40.1%	**61.3%**	—	—	54.4%	57.0%
level	who reports whom	**79.3%**	18.4%	30.4%	**66.1%**	28.6%	**66.2%**	**78.4%**	29.5%	—	—	50.3%	61.2%
department	advise relation	40.5%	**59.2%**	57.9%	47.0%	42.5%	37.9%	**57.8%**	**60.7%**	—	—	44.8%	57.1%
department	friendship	39.4%	**60.4%**	56.5%	47.6%	47.7%	43.6%	51.8%	52.2%	—	—	50.8%	57.0%
department	who reports whom	22.0%	**80.7%**	**76.2%**	51.9%	57.8%	30.0%	42.1%	55.1%	—	—	46.5%	61.2%
		Freeman's EIES network (32 nodes)											
citations	aquaintanceship at time 1	54.4%	45.4%	42.5%	50.2%	42.1%	36.4%	**63.0%**	**58.8%**	—	—	**59.7%**	54.7%
citations	aquaintanceship at time 2	53.7%	46.3%	46.0%	55.1%	38.4%	38.1%	**60.7%**	**56.7%**	—	—	**64.2%**	55.5%
citations	messages	38.9%	**61.3%**	**57.8%**	48.5%	**56.6%**	**59.0%**	41.1%	**60.1%**	—	—	52.7%	54.2%
discipline	aquaintanceship at time 1	40.4%	**58.2%**	**56.7%**	45.0%	49.3%	40.3%	44.5%	51.1%	—	—	**57.7%**	54.7%
discipline	aquaintanceship at time 2	48.5%	53.5%	51.2%	47.7%	47.7%	46.7%	48.0%	51.6%	—	—	52.5%	55.5%
discipline	messages	51.3%	49.0%	50.8%	47.0%	44.9%	46.4%	**55.5%**	51.3%	—	—	51.6%	54.2%
		Countries trade data (24 nodes)											
population size	food and live animals	**63.1%**	37.4%	31.2%	**59.2%**	26.2%	**61.0%**	**79.4%**	**62.9%**	—	—	54.7%	56.4%
population size	crude materials	**57.4%**	43.1%	32.0%	**60.3%**	22.7%	**59.8%**	**84.5%**	**70.4%**	—	—	56.2%	56.3%
population size	mineral fuels	54.1%	46.6%	40.3%	**61.1%**	34.1%	54.0%	**68.9%**	**57.3%**	—	—	48.5%	56.0%
population size	basic manufactured goods	**66.4%**	33.9%	23.4%	**58.0%**	20.7%	**66.7%**	**88.2%**	**68.8%**	—	—	**62.1%**	56.3%
population size	exchange of diplomats	54.3%	45.7%	38.8%	56.3%	30.1%	54.4%	**76.3%**	**67.0%**	—	—	56.2%	56.3%
GNP per capita	food and live animals	39.8%	**60.5%**	53.7%	54.3%	**62.3%**	48.2%	35.4%	**57.1%**	—	—	47.3%	56.4%
GNP per capita	crude materials	42.6%	**57.6%**	50.6%	52.0%	**62.4%**	53.5%	38.4%	**60.3%**	—	—	48.7%	56.3%
GNP per capita	mineral fuels	45.4%	54.8%	52.8%	**56.6%**	**57.0%**	50.2%	43.8%	**57.6%**	—	—	42.3%	56.0%
GNP per capita	basic manufactured goods	38.9%	**61.3%**	**57.0%**	54.6%	**58.4%**	48.6%	40.7%	**64.7%**	—	—	49.7%	56.3%
GNP per capita	exchange of diplomats	41.0%	**59.1%**	52.4%	49.3%	**62.8%**	50.6%	36.3%	55.9%	—	—	48.7%	56.3%
education	food and live animals	56.2%	44.1%	46.4%	**62.0%**	29.2%	**59.4%**	**81.5%**	**62.7%**	—	—	49.5%	56.4%
education	crude materials	**61.6%**	38.6%	43.8%	**59.5%**	26.0%	**61.0%**	**82.0%**	**58.6%**	—	—	48.5%	56.3%
education	mineral fuels	**58.0%**	42.2%	47.7%	**60.8%**	29.7%	53.3%	**74.0%**	52.4%	—	—	48.4%	56.0%
education	basic manufactured goods	**59.2%**	40.9%	44.7%	**59.6%**	27.2%	**58.9%**	**79.5%**	**57.0%**	—	—	50.0%	56.3%
education	exchange of diplomats	54.7%	45.3%	47.1%	**56.7%**	32.0%	**56.5%**	**71.0%**	**60.9%**	—	—	48.5%	56.3%
energy use	food and live animals	**73.1%**	26.9%	25.5%	**57.2%**	38.4%	**62.0%**	**81.9%**	**61.1%**	—	—	**65.1%**	56.4%
energy use	crude materials	**72.2%**	27.8%	27.9%	**56.6%**	31.7%	**57.3%**	**81.3%**	**62.5%**	—	—	**63.8%**	56.3%
energy use	mineral fuels	**62.6%**	37.4%	39.1%	**65.6%**	40.1%	50.9%	**69.3%**	54.3%	—	—	49.2%	56.0%
energy use	basic manufactured goods	**76.1%**	23.9%	23.6%	**61.3%**	30.5%	**61.7%**	**83.8%**	**57.4%**	—	—	**65.5%**	56.3%
energy use	exchange of diplomats	**63.5%**	36.5%	37.8%	53.1%	40.3%	52.4%	**70.0%**	**59.1%**	—	—	**63.8%**	56.3%
		Padgett's Florentine Families (16 nodes)											
wealth	business relation	44.5%	55.6%	54.6%	47.4%	54.3%	45.7%	48.0%	58.3%	—	—	47.9%	61.8%
wealth	marriage relation	52.8%	46.5%	42.0%	48.3%	43.6%	54.3%	58.4%	53.2%	—	—	51.3%	63.2%
priors	business relation	55.0%	44.2%	43.2%	48.0%	55.5%	60.1%	39.9%	33.4%	—	—	41.7%	61.8%
priors	marriage relation	46.6%	51.0%	48.0%	39.1%	**66.2%**	51.5%	35.8%	41.7%	—	—	48.9%	63.2%
ties	business relation	49.7%	51.2%	51.6%	50.2%	50.9%	58.6%	45.5%	58.6%	—	—	54.0%	61.8%
ties	marriage relation	43.6%	54.7%	55.4%	44.3%	53.1%	44.2%	53.7%	**71.4%**	—	—	**68.8%**	63.2%
mean AUC		**51.7%**	**48.3%**	**46.5%**	**53.1%**	**44.0%**	**51.1%**	**59.3%**	**56.9%**	—	—	**53.1%**	
percentage of values better than random		**27.3%**	**27.3%**	**15.9%**	**36.4%**	**18.2%**	**27.3%**	**50.0%**	**54.6%**	—	—	**25.0%**	
rooted mean L_2-deviation from best AUC		**0.187**	**0.271**	**0.280**	**0.165**	**0.312**	**0.187**	**0.135**	**0.156**	—	—	**0.176**	

node attribute	**reference network**	**NETHSIC with node kernel**								**method**			**random**
input	**output reference**	`lapl`	`1step`	`sp`	`betw`	`close`	`adj`	`degr`	`Δdegr`	GGM	corr	mi	$\alpha = 5\%$
		Krackhardt's High-tech managers (21 nodes)											
all normalized	advise relation	42.5%	**57.5%**	48.2%	45.0%	43.7%	47.5%	45.6%	49.0%	51.6%	56.5%	50.4%	57.1%
all normalized	friendship	34.8%	**65.2%**	**60.5%**	45.1%	54.9%	36.2%	53.4%	**60.9%**	55.3%	**61.5%**	44.1%	57.0%
all normalized	who reports whom	26.6%	**73.4%**	**77.4%**	**66.1%**	37.3%	12.2%	58.2%	**70.9%**	55.0%	56.5%	55.7%	61.2%
		Freeman's EIES network (32 nodes)											
all normalized	aquaintanceship at time 1	42.9%	**57.1%**	**55.6%**	50.2%	**60.7%**	40.8%	45.9%	53.9%	53.3%	**54.9%**	**58.1%**	54.7%
all normalized	aquaintanceship at time 2	48.3%	51.8%	51.1%	55.1%	**61.4%**	46.6%	47.9%	55.5%	46.7%	49.2%	**63.0%**	55.5%
all normalized	messages	47.3%	52.7%	51.5%	48.5%	49.3%	47.8%	**55.0%**	52.6%	48.6%	**57.0%**	53.1%	54.2%
		Countries trade data (24 nodes)											
all normalized	food and live animals	**58.9%**	41.2%	37.8%	**56.5%**	27.5%	**62.5%**	**84.5%**	**58.0%**	**59.0%**	34.5%	38.1%	56.4%
all normalized	crude materials	**59.9%**	40.1%	40.6%	55.0%	25.9%	**60.6%**	**86.3%**	**60.9%**	**58.4%**	34.1%	40.0%	56.3%
all normalized	mineral fuels	**59.1%**	40.9%	44.2%	53.8%	31.1%	53.8%	**74.0%**	51.0%	**63.1%**	42.3%	47.3%	56.0%
all normalized	basic manufactured goods	**62.3%**	37.7%	37.6%	**59.7%**	24.1%	**63.7%**	**87.9%**	**57.5%**	54.1%	29.0%	43.8%	56.3%
all normalized	exchange of diplomats	54.0%	46.0%	43.1%	45.0%	33.6%	54.7%	**76.0%**	**57.3%**	**58.4%**	34.1%	40.0%	56.3%
		Padgett's Florentine Families (16 nodes)											
all normalized	business relation	49.3%	50.7%	49.0%	47.0%	54.2%	57.3%	40.1%	42.6%	52.7%	45.4%	49.2%	61.8%
all normalized	marriage relation	44.7%	55.2%	51.9%	45.5%	60.6%	53.5%	41.0%	44.6%	38.9%	51.9%	50.8%	63.2%
mean AUC		**48.5%**	**51.5%**	**49.9%**	**51.7%**	**43.4%**	**49.0%**	**61.2%**	**55.0%**	**53.5%**	**46.7%**	**48.7%**	
percentage of values better than random		**30.8%**	**30.8%**	**23.1%**	**23.1%**	**15.4%**	**23.1%**	**46.2%**	**46.2%**	**30.8%**	**23.1%**	**15.4%**	
rooted mean L_2-deviation from best AUC		**0.238**	**0.260**	**0.266**	**0.200**	**0.357**	**0.257**	**0.117**	**0.173**	**0.186**	**0.308**	**0.269**	

Table 1: NETHSIC in social network analysis. Shown are the AUC values reached by NETHSIC for various combinations of attribute kernel (column 1), node kernel (one from column 3-10), and gold reference network (column 2). For comparison we applied correlation relevance networks (corr), graphical Gaussian models (GGM) and mutual information relevance networks (mi) (column 11-13). Note that correlation relevance networks and graphical Gaussian models are not applicable for individual attributes as these methods rely on correlations respective partial correlations within multiple attributes. Results significantly better than random ($\alpha = 0.05$) (column 14) are shown in bold. (Abbreviations of node kernels: `1step` 1-step random walk on unnormalized Laplacian, `lapl` Laplacian; `sp` linear kernel on shortest path vectors; `betw` linear kernel on betweenness centrality; `close` linear kernel on closeness centrality; `adj` squared adjacency matrix; `degr` linear kernel on node degrees; `Δdegr` delta kernel on node degrees)

4.2 Results

We report AUC values on the reference networks for both individual features (top) and the combination of features (bottom) in Table 1. We assessed whether an AUC-value is significant by randomly shuffling its predictions 10^4 times and checking whether its result was among the best 5% of these random permutations which corresponds to a significance level of $\alpha = 5\%$. In the tables significant results are listed in bold. Further, the rooted mean L_2-deviation from the best AUC on each data set for each of the methods was determined. This statistics rewards methods that perform close to the best-performing method across all data sets.

What we found most interesting is that NETHSIC did not reach its highest or most significant AUC values when we used the 1-step random walk kernel (`1step`), but for the linear and delta kernels on the degree (`degr`, Δ`degr`). This means that especially on the Countries data set, it is **not** a good strategy to assume that nodes with similar attributes must be connected, but rather that they share a similar node degree in the underlying network. Across all 4 data sets, the degree kernel, which is a node kernel with global effects, reached AUC values that were better than random. The general message is: In these social networks, similar node attributes mostly indicate that these nodes have a similar degree in the network. Note also that neither mutual information or correlation based network inference nor graphical Gaussian models could reach the same mean AUC score as NETHSIC in combination with the degree kernel. The number of measurements per node (1-4 attributes) seems to be insufficient for accurate network inference using these methods, and at the same time, they cannot model an explicit link between node attributes and particular topological properties of the nodes.

5 Conclusions

In this paper we define a kernel-based framework for unsupervised structured network inference. We can interpret existing heuristic-based algorithms in our statistical framework and demonstrate the utility of advanced graph scoring functions taking the graph topology into account. In future work, we will focus on extending this framework to semi-supervised settings: Given some edges, the task is to complete the network, while observing constraints on network topology. Speeding up runtime for kernels with global effects is another algorithmic challenge that will be addressed in future work. The theoretical computer science community has studied the problem of updating centrality scores and shortest paths in a dynamically changing network, which will alleviate this problem for kernels based on these types of topological indices.

References

Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 509–512.

Ben-Hur, A., & Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, *21*, 38–46.

Butte, A. J., & Kohane, I. S. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing* (pp. 415–426).

Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., & Kohane, I. S. (2000). Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks. *PNAS*, *97*, 12182–12186.

Fukumizu, K., Bach, F. R., & Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *JMLR*, *5*, 73–99.

Getoor, L., & Taskar, B. (2007). *Introduction to statistical relational learning*. MIT Press.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *ALT*.

Gretton, A., Fukumizu, K., Teo, C.-H., Song, L., Schölkopf, B., & Smola, A. (2008). A kernel test of statistical dependence. *NIPS 20*.

Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., & Gerstein, M. (2003). A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, *302*, 449–53.

Jeong, H., Mason, S. P., Barabási, A. L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, *411*, 41–42.

Smola, A. J., & Kondor, I. R. (2003). Kernels and regularization on graphs. *COLT*.

Song, L., Smola, A., Gretton, A., Borgwardt, K., & Bedo, J. (2007). Supervised feature selection via dependence estimation. *ICML*.

Steinwart, I. (2002). The influence of the kernel on the consistency of support vector machines. *JMLR*, *2*.

Toh, H., & Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical modeling. *Bioinformatics*, *18*, 287–297.

Vert, J.-P., Qiu, J., & Noble, W. (2007). A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, *8*, S8.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge University Press.