The Graphlet Spectrum

Risi Kondor RISI@GATSBY.UCL.AC.UK
Gatsby Computational Neuroscience Unit, UCL, 17 Queen Square, London, WC1N 3AR, U.K.

Nino Shervashidze NINO.SHERVASHIDZE@TUEBINGEN.MPG.DE
Karsten M. Borgwardt KARSTEN.BORGWARDT@TUEBINGEN.MPG.DE
Interdepartmental Bioinformatics Group, Max Planck Institute for Biological Cybernetics, Max Planck Institute for Developmental Biology, Spemannstr. 41, 72076 Tübingen, Germany

Abstract

Current graph kernels suffer from two limitations: graph kernels based on counting particular types of subgraphs ignore the relative position of these subgraphs to each other, while graph kernels based on algebraic methods are limited to graphs without node labels. In this paper we present the *graphlet spectrum*, a system of graph invariants derived by means of group representation theory that capture information about the number as well as the position of labeled subgraphs in a given graph. In our experimental evaluation the graphlet spectrum outperforms state-of-the-art graph kernels.

1. Introduction

Over recent years, graph kernels have grown to become an important branch of graph mining. Their fundamental purpose is to represent a graph by features in a reproducing kernel Hilbert space. While most graph kernels arrive at these features by counting particular types of subgraphs, such as walks, shortest paths, subgraphs of a fixed size k, or subtrees (Kashima et al., 2003; Gärtner et al., 2003; Borgwardt & Kriegel, 2005; Shervashidze et al., 2009; Bach, 2008), we have recently proposed a group theoretical approach and found it to have state-of-the-art performance (Kondor & Borgwardt, 2008). However, both approaches have limitations: in counting subgraphs, the graph-theoretic approach ignores the relative position of subgraphs within the graph, while the algebraic approach suffers from the fact that it is restricted to unlabeled graphs, which are uncommon in applications.

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

In this paper, we overcome these two limitations by defining a new group-theoretic approach that allows both for labeled subgraphs and considers the relative position of subgraphs.

2. Graph invariants

In this paper $\mathcal{G}$ will be a directed weighted graph of n vertices. We represent $\mathcal{G}$ by its adjacency matrix $A \in \mathbb{R}^{n\times n}$, where $[A]_{i,j} \in \mathbb{R}$ is the weight of the edge from vertex i to vertex j. Unweighted graphs are special cases where $[A]_{i,j} \in \{0,1\}$, while in undirected graphs $A^\top = A$.

One of the key issues in learning on graphs is that whatever way we choose to represent $\mathcal{G}$, it must be invariant to vertex relabeling. Specifically, if we represent $\mathcal{G}$ by some sequence of features $c_1(A), c_2(A), \ldots, c_K(A)$, then for any permutation $\pi\colon \{1,2,\ldots,n\} \to \{1,2,\ldots,n\}$ these features computed from the permuted adjacency matrix

$$[A^\pi]_{\pi(i),\pi(j)} = A_{i,j}$$

must be the same, since A^π is just a different representation of the same graph. Such functions $c\colon \mathbb{R}^{n\times n} \to \mathbb{R}$ satisfying

$$c(A) = c(A^\pi), \qquad \forall \pi$$

are called **graph invariants** and they have a large literature both in pure mathematics and in applied domains (Wale & Karypis, 2006; Mikkonen, 2007).

2.1. The algebraic approach

Proponents of the algebraic approach to graph invariants focus not so much on the graph itself, but on the inherent structure of the permutations acting on the adjacency matrix. Recall that the natural way to define the product of two permutations σ_1 and σ_2 is by composition, i.e., $(\sigma_2\sigma_1)(i) = \sigma_2(\sigma_1(i))$, and that with

respect to this notion of multiplication the $n!$ different permutations of n objects form the **symmetric group** of degree n, denoted $\mathbb{S}_n$. Saying that $\mathbb{S}_n$ is a group means that it satisfies the following axioms:

G1 for any $\sigma_1, \sigma_2 \in \mathbb{S}_n$, $\sigma_2\sigma_1 \in \mathbb{S}_n$;

G2 for any $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{S}_n$, $\sigma_3(\sigma_2\sigma_1) = (\sigma_3\sigma_2)\sigma_1$;

G3 there is a unique $e \in \mathbb{S}_n$ satisfying $e\sigma = \sigma e = \sigma$ for any $\sigma \in \mathbb{S}_n$; and finally,

G4 for any $\sigma \in \mathbb{S}_n$ there is a unique $\sigma^{-1} \in \mathbb{S}_n$ such that $\sigma\sigma^{-1} = \sigma^{-1}\sigma = e$.

Given a function $f\colon \mathbb{S}_n \to \mathbb{R}$, the group structure suggests defining the **left–translate** of f by $\pi \in \mathbb{S}_n$ as

$$f^\pi\colon \mathbb{S}_n \to \mathbb{R}, \qquad f^\pi(\sigma) = f(\pi^{-1}\sigma).$$

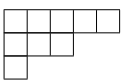
In (Kondor & Borgwardt, 2008) we have shown that if we encode the adjacency matrix in the function

$$f_A(\sigma) = A_{\sigma(n),\sigma(n-1)}, \tag{1}$$

then permuting the vertices of $\mathcal{G}$ by π transforms f_A exactly into $(f_A)^\pi$. This reduces the problem of constructing graph invariants to constructing left–translation invariant functionals of functions on $\mathbb{S}_n$. The specific invariants they use are grounded in the theory of non-commutative harmonic analysis.

2.2. Fourier space invariants

Recall that a (finite dimensional, complex valued) **representation** of a group G is a matrix valued function $\rho\colon G \to \mathbb{C}^{d\times d}$ satisfying $\rho(xy) = \rho(x)\rho(y)$ for all $x, y \in G$. If G is finite, then one can find finite collections $\mathcal{R}$ of such representations that are fundamental in the sense that (a) no two representations in $\mathcal{R}$ are equivalent up to similarity transformation; (b) any representation of G reduces into a direct sum of representations in $\mathcal{R}$; (c) each $\rho \in \mathcal{R}$ is unitary, i.e., $\rho(x)^{-1} = \rho(x)^\dagger$. For more background in representation theory the reader is referred to (Serre, 1977). In the case of the symmetric group, a popular choice for $\mathcal{R}$ is **Young's Orthogonal Representation** (YOR), and the individual $\rho \in \mathcal{R}$ representations are usually labeled by the **integer partitions** $\{\lambda \vdash n\}$. An integer partition $\lambda \vdash n$ is a sequence of natural numbers $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ satisfying $\sum_{i=1}^k \lambda_i = n$ and $\lambda_i \geq \lambda_{i+1}$ for $i = 1, 2, \ldots k-1$. It is convenient to represent integer partitions by Young diagrams, such as



for $\lambda = (5, 3, 1)$. A special property of YOR is that each of the $\rho_\lambda(\sigma)$ matrices are real valued.

In terms of YOR (or indeed any other complete system of inequivalent irreducible representations of $\mathbb{S}_n$) the **Fourier transform** of a function $f\colon \mathbb{S}_n \to \mathbb{R}$ is defined as the sequence of matrices

$$\widehat{f}(\lambda) = \sum_{\sigma\in\mathbb{S}_n} f(\sigma)\,\rho_\lambda(\sigma) \qquad \lambda \vdash n. \tag{2}$$

Of the several properties of ordinary Fourier transformation inherited by such generalized Fourier transforms, we are particularly interested in the **translation theorem**, which states that

$$\widehat{f^\pi}(\lambda) = \rho_\lambda(\pi)\,\widehat{f}(\lambda) \qquad \lambda \vdash n. \tag{3}$$

Coupled with the unitarity of $\rho_\lambda(\pi)$, this immediately tells us that the matrices

$$\widehat{a}(\lambda) = \widehat{f}(\lambda)^\dagger \cdot \widehat{f}(\lambda) \qquad \lambda \vdash n \tag{4}$$

are translation invariant. This is called the **power spectrum** of f. In (Kondor & Borgwardt, 2008) we employed further, more powerful, invariants, in particular, the **skew spectrum**

$$\widehat{q}_\nu(\lambda) = \widehat{r}_\nu(\lambda)^\dagger \cdot \widehat{f}(\lambda), \qquad \lambda \vdash n, \qquad \nu \in \mathbb{S}_n, \tag{5}$$

where $r_\nu(\sigma) = f(\sigma\nu)f(\sigma)$.

In summary, the Fourier approach to computing graph invariants consists of the following steps:

1. compute f_A from A as in (1),
2. compute the Fourier transform $\widehat{f}_A$ by (2),
3. compute the skew spectrum (5) or any other system of left–translation invariant matrices and use these as graph invariants.

A fundamental issue in (Kondor & Borgwardt, 2008) was that the $\widehat{q}_\nu(\lambda)$ matrices turned out to be extremely sparse. Indeed, irrespective of the size of $\mathcal{G}$, the entire skew spectrum has only 87 independent nonzero elements, and its reduced, $O(n^3)$ computable version which we used in the experiments had just 49. While the skew spectrum is reported to have excellent performance on small and medium-sized graphs, this nonetheless casts a shadow on its representational power as n increases. Another limitation of the skew spectrum is that it is fairly rigid: most crucially for applications, there is no simple way of incorporating labels on the vertices or edges. The present work addresses both of these issues.

3. The graphlet spectrum

A common alternative to the algebraic approach described above is to characterize graphs in terms of the

frequency or position of certain elementary subgraphs embedded within them. Depending on the context these small subgraphs are usually called **graphlets** (which is the terminology that we will use in this paper) or **motifs**. Given a graphlet g of $k < n$ vertices whose adjacency matrix we denote with the same letter g, the **indicator**

$$\mu_g(v_1, v_2, \ldots, v_k) = \begin{cases} 1 & \text{if } \; g_{i,j} \leq A_{v_i,v_j} \quad \forall\, i,j, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

captures whether g is a subgraph of $\mathcal{G}$ at position $(v_1, v_2, \ldots, v_k)$, whereas

$$\mu_g^{\text{ind}}(v_1, v_2, \ldots, v_k) = \begin{cases} 1 & \text{if } \; g_{i,j} = A_{v_i,v_j} \quad \forall\, i,j, \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

captures whether g is an induced subgraph at $(v_1, v_2, \ldots, v_k)$.

The fundamental observation motivating the present paper is that (at least for unweighted graphs), f_A, as defined in (1), can be re-written as

$$f_A(\sigma) = \mu_e(\sigma(n), \sigma(n-1)),$$

where e stands for the elementary graphlet of two vertices and a single directed edge. In other words, f_A encodes where the edge e occurs in $\mathcal{G}$ as a subgraph. It is easy to extend this idea to larger graphlets by letting

$$f_{A,g}(\sigma) = \mu_g(\sigma(n), \sigma(n-1), \ldots, \sigma(n-k+1)), \tag{8}$$

or the analogous expression with μ^{ind} in the induced subgraph case. Crucially, $f_{A,g}$ defined in this way will obey the same transformation property as before, since if μ_g^π is the indicator of the permuted adjacency matrix A^π, then

$$\mu_g^\pi(\pi(v_1), \pi(v_2), \ldots, \pi(v_k)) = \mu_g(v_1, v_2, \ldots, v_k), \tag{9}$$

hence

$$\mu_g^\pi(v_1, \ldots, v_k) = \mu_g(\pi^{-1}(v_1), \ldots, \pi^{-1}(v_k)),$$

and therefore

$$f_{A^\pi,g}(\sigma) = \mu_g(\pi^{-1}\sigma(n), \ldots, \pi^{-1}\sigma(n-k+1)) = f_{A,g}(\pi^{-1}\sigma) = (f_{A,g})^\pi(\sigma). \tag{10}$$

This means that just as in Section 2, we can invoke the machinery of power spectra, skew spectra, etc. to derive graph invariants, but now these new invariants will be sensitive to the presence of entire subgraphs in $\mathcal{G}$ and not just individual edges.

An attractive feature of our approach is that given a small library $g_1, g_2, \ldots, g_m$ of graphlets we can compute a separate f_{A,g_i} function for each graphlet, and then form invariants from all possible combinations of these functions, capturing information about the relative position of different types of subgraphs as well as different subgraphs of the same type. Since in this case second order invariants such as (4) already yield a rich set of features, we forgo computing higher order, more expensive invariants, such as the skew spectrum. Our exact definition of the graphlet spectrum is as follows.

Definition 1 *Given a graph $\mathcal{G}$ of n vertices and adjacency matrix A, relative to a collection $g_1, g_2, \ldots, g_m$ of graphlets and an indicator function such as (6) or (7), the **graphlet spectrum** of $\mathcal{G}$ is defined to be the sequence of matrices*

$$\widehat{q}_{i,j}(\lambda) = \big(\widehat{f}_{A,g_i}(\lambda)\big)^\dagger \cdot \widehat{f}_{A,g_j}(\lambda), \qquad j \leq i, \qquad \lambda \vdash n, \tag{11}$$

where f_{A,g_i} is defined as in (8).

Proposition 1 *Each scalar component $[\widehat{q}_{i,j}(\lambda)]_{a,b}$ of the graphlet spectrum is a graph invariant.*

Proof. If we permute the vertices of $\mathcal{G}$ by $\pi \in \mathbb{S}_n$, then by (10) the graphlet spectrum becomes

$$\widehat{q}^\pi_{i,j}(\lambda) = \big(\widehat{f}_{A^\pi,g_i}(\lambda)\big)^\dagger \cdot \widehat{f}_{A^\pi,g_j}(\lambda) = \big(\widehat{f}^\pi_{A,g_i}(\lambda)\big)^\dagger \cdot \widehat{f}^\pi_{A,g_j}(\lambda),$$

which by the translation theorem (3) is further equal to

$$\big(\rho_\lambda(\pi)\, \widehat{f}_{A,g_i}(\lambda)\big)^\dagger \cdot \big(\rho_\lambda(\pi)\, \widehat{f}_{A,g_j}(\lambda)\big),$$

but by unitarity $\rho_\lambda(\pi)^\dagger \rho_\lambda(\pi) = I$, so these factors cancel, and $\widehat{q}^\pi_{i,j}(\lambda) = \widehat{q}_{i,j}(\lambda)$. ■

3.1. Generalizations

It should be clear from the above that (11) being invariant does not explicitly require μ to be an indicator for subgraphs. Indeed, any function of the vertices that depends purely on the graph structure and not the numbering of the vertices will transform according to (9), and hence the Fourier transform of the corresponding $f_\mu(\sigma) = (\sigma(n), \ldots \sigma(n-k))$ will be a candidate for inclusion in (11). This also includes real-valued μ.

At the most general level the graphlet spectrum is a system of invariants for k'th order weighted, directed **hypergraphs**, since (9) describes exactly the way that the "adjacency matrix" of such hypergraphs transforms under permutation. Equations (6) and (7) just define the "embedding hypergraphs" of g in A.

This generalization is particularly useful for taking into account label information, the absence of which

is probably the most severe limitation of (Kondor & Borgwardt, 2008). For example, if $\mathcal{G}$ is the structure of a molecule and g is a small chemical feature such as a functional group in organic chemistry, then we can redefine (6) to indicate a match only when both the topology and the labeling (i.e., what kind of atom occupies each vertex) match up between g and A. Edge labels may be incorporated in a similar way.

4. Computational considerations

More often than not, the biggest challenge in applying representation theoretical ideas to real world problems is making the necessary computations scalable. In the case of the graphlet spectrum at first sight it appears that computing the Fourier transform (2) already demands $O((n!)^2)$ time, which is clearly forbiddingly expensive. There are two key ingredients to reducing this computational burden to a level that is feasible in a practical algorithm: sparsity and the theory of fast Fourier transforms. We aim for applications involving medium sized graphs (few hundred nodes), and a handful of graphlets with k in the range 2 to 6.

4.1. Sparsity

Since the Fourier transform $f \mapsto \widehat{f}$ is a unitary transformation, the combined size of the $\widehat{f}(\lambda)$ matrices appearing in (2) is $n!$. However, any f defined by (8) is a so-called right $\mathbb{S}_{n-k}$–invariant function. For such functions most $\widehat{f}(\lambda)$ Fourier components turn out to be identically zero, and (at least in YOR) even the remaining components will have a characteristic column-sparse structure. To describe this structure we need the following facts from representation theory:

1. The individual rows/columns of the $\rho_\lambda(\sigma)$ representation matrices are indexed by so-called **standard Young tableaux**, which we get by bijectively filling the boxes of the Young diagram of λ with the numbers $1, 2, \ldots, n$ according to the rule that the numbers must increase left to right in each row and top to bottom in each column. For example,

1	3	4	5	8
2	6			
7				

   is a standard tableau of shape $(5, 2, 1)$. The set of standard tableaux of shape λ we denote by $\mathcal{T}_\lambda$ and the set of standard tableaux of n boxes by $\mathcal{T}_n$. There is only one standard tableau of shape (n), and we will depict it as $\boxed{1}\boxed{2}\boxed{}\boxed{}\boxed{n}$.

2. The $\rho_{(n)}$ representation is the one-dimensional **trivial representation** $\rho_{(n)}(\sigma) = (1)\ \forall \sigma \in \mathbb{S}_n$.

3. There is a natural partial order on partitions in which $\lambda' \geq \lambda$ if and only if $\lambda' = (\lambda_1, \lambda_2, \ldots, \lambda_{k'})$ can be derived from $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ by adding boxes, i.e, $\lambda'_j \geq \lambda_i$ for $i = 1, 2, \ldots, k$.

4. There is a corresponding partial order on standard tableaux in which $t' \geq t$ (with $t' \in \mathcal{T}_n$ and $t \in \mathcal{T}_m$) if and only if t' can be derived from t by adding boxes containing the numbers $m+1, m+2, \ldots, n$.

5. For $m < n$, the permutations that fix $m+1$, $m+2, \ldots, n$ form a subgroup in $\mathbb{S}_n$, which we identify with $\mathbb{S}_m$.

6. YOR has the special property that if we restrict σ to $\mathbb{S}_{n-1}$, then ρ_λ decomposes into a direct sum of YOR representations of $\mathbb{S}_{n-1}$ in the form

$$\rho_\lambda(\sigma) = \bigoplus_{\substack{\lambda^- \leq \lambda \\ \lambda^- \vdash n-1}} \rho_{\lambda^-}(\sigma), \qquad \sigma \in \mathbb{S}_{n-1}, \quad (12)$$

   where the ρ_{λ^-} block is at the intersection of rows/columns that are indexed by standard tableux that yield a tableau of shape λ^- on removal of the box containing n.

Together these facts lead to the following result, which is fairly well-known in the world of computational non-commutative harmonic analysis.

Proposition 2 *If $f\colon \mathbb{S}_n \to \mathbb{R}$ is defined as in (8), then in YOR its Fourier transform (2) will be identically zero except for those columns indexed by $\{\, t \in \mathcal{T}_n \mid t \geq \boxed{1}\boxed{2}\boxed{}\boxed{}\boxed{*} \,\}$, where $*$ stands for $n-k$. Furthermore, the total number of scalar entries in these columns is exactly $n!/(n-k)!$.*

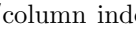
Proof. The sets $\sigma\mathbb{S}_m = \{\, \sigma\tau \mid \tau \in \mathbb{S}_m \,\} \subset \mathbb{S}_n$ are called left $\mathbb{S}_m$–cosets and by $\mathbb{S}_n/\mathbb{S}_m$ we mean a set of permutations with exactly one permutation from each such coset. If f (dropping the A, g indices) is defined as in (8), then it is constant on each left $\mathbb{S}_{n-k}$–coset, therefore its Fourier transform can be written as

$$\widehat{f}(\lambda) = \sum_{\sigma \in \mathbb{S}_n/\mathbb{S}_{n-k}} \sum_{\tau \in \mathbb{S}_{n-k}} f(\sigma)\, \rho_\lambda(\sigma\tau) = \sum_{\sigma \in \mathbb{S}_n/\mathbb{S}_{n-k}} f(\sigma)\, \rho_\lambda(\sigma) \sum_{\tau \in \mathbb{S}_{n-k}} \rho_\lambda(\tau). \quad (13)$$

Recursively applying (12) gives

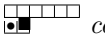
$$\rho_\lambda(\tau) = \bigoplus_{\lambda^- \in \Lambda^-} \rho_{\lambda^-}(\tau), \quad (14)$$

where Λ^- is a multiset of partitions of $n-k$, the multiplicity of each $\lambda^- \in \Lambda^-$ being determined by how

many distinct ways there are of arriving at λ^- by successive "legal" removals of boxes from λ. In particular, the trivial representation $\rho_{(n-k)}(\tau) = (1)$ occurs in (14) exactly at those locations on the diagonal where the row/column index satisfies $t \geq$  . At these diagonal locations we will have an entry of $\sum_{\tau \in \mathbb{S}_{n-k}} \rho_\lambda(\tau) = (n-k)!$. By the unitarity of the Fourier transform (on $\mathbb{S}_{n-k}$), all other representations must be orthogonal to $\rho_{(n-k)}$ in the sense that for these representations $\sum_{\tau \in \mathbb{S}_{n-k}} \rho_{\lambda^-}(\tau) = 0$, and hence every other entry in (14) is zero.

Since the set of right $\mathbb{S}_{n-k}$–invariant functions spans a space of dimension $n!/(n-k)!$, and the Fourier transform is unitary (hence, linear and invertible), the total size of the non-zero columns of $\widehat{f}$ must be at least $n!/(n-k)!$. Examining (13) reveals that any function that is orthogonal to this space will not contribute to the columns in question, so again by unitarity, the size of the columns must, in fact, be exactly $n!/(n-k)!$. ■

Example 1 *(c.f., (Kondor & Borgwardt, 2008)) In the simplest case of graphlets which are just single edges ($k=2$) Proposition 2 tells us that the only non-zero components of $\widehat{f}$ are*

1. *the single scalar component $\widehat{f}_{(n)}$;*
2. *the*  *column of $\widehat{f}_{(n-1,1)}$;*
3. *the*  *column of $\widehat{f}_{(n-1,1)}$;*
4. *the*  *column of $\widehat{f}_{(n-2,2)}$;*
5. *the*  *column of $\widehat{f}_{(n-2,1,1)}$,*

where ■ denotes n and • denotes $n-1$.

Proposition 3 *If the number of vertices of each of the graphlets $g_1, g_2, \ldots, g_m$ is k, and $n \geq 2k$, then the graphlet spectrum (11) has*

$$\binom{m}{2} \sum_{s=0}^{k} \binom{k}{s}^2 s!$$

non-zero scalar components.

Proof. Clearly, there are $\binom{m}{2}$ ways of choosing i and j in (11). Now for fixed i and j, by Proposition 2, non-zeros can only occur in $\widehat{q}_{i,j}(\lambda)$ matrices indexed by λ that have $0 \leq s \leq k$ boxes in the second and higher rows. Within such a matrix the non-zeros are at the intersection of rows/columns indexed by standard tableaux from $T^k_\lambda = \{\, t \in \mathcal{T}_\lambda \mid t \geq$ $\}$, so in total there are $|\, T^k_\lambda \,|^2$ non-zero matrix elements. In enumerating the $t \in T^k_\lambda$ there are $\binom{k}{s}$ ways of choosing which $k-s$ of the numbers $n-k+1, \ldots, n$ should go in the first row, and let us say, g_λ ways of arranging the remaining s numbers in rows two and higher. This means that for fixed i, j and s there are $\binom{k}{s}^2 \sum_{\lambda \vdash n,\ \lambda_1 = n-s} g^2_\lambda$ non-zeros to account for. However, if we let λ^* be the partition that we get by stripping away the entire first row of λ, then g_λ is exactly the number of "legal" ways of arranging $1, 2, \ldots, s$ in a partition of shape λ^*, i.e., $g_\lambda = |\, \mathcal{T}_{\lambda^*} \,|$. Since by unitarity the number of scalar components of a function (on $\mathbb{S}_s$) and its Fourier transform must be the same, $\sum_{\lambda \vdash n,\ \lambda_1 = n-s} g^2_\lambda = \sum_{\lambda^* \vdash s} |\, \mathcal{T}_{\lambda^*} \,|^2 = s!$. ■

Table 1. The size of the graphlet spectrum (in terms of scalar components) induced by a single graphlet of k vertices. The cases $k = 3, 4, 5$ are probably the most interesting since they extract "higher order structure" from the graph; $k = 2$ is useful in the context of multiple graphlets encoding different labeled features, otherwise it just reproduces a subset of the skew spectrum; $k = 6$ and higher are typically too expensive to compute.

k	2	3	4	5	6
size	7	34	209	1,546	13,327

4.2. Fast Fourier Transforms

The reason that $\widehat{f}$ can be efficiently computed is not just that it is sparse, but that its sparsity structure is closely matched to the structure of the non-commutative fast Fourier transforms that are gaining popularity in the non-commutative harmonic analysis community (Rockmore, 1997; Clausen, 1989). In general, such FFTs reduce computing $\widehat{f}$ to computing n separate transforms on $\mathbb{S}_{n-1}$, which are in turn each reduced to $n-1$ transforms on $\mathbb{S}_{n-2}$, and so on. To describe the specialized version of Clausen's FFT that we developed to compute the graphlet spectrum, we need the following concepts.

1. Complexity is measured in **scalar operations**, which is a single scalar multiplication followed by a scalar addition. Copying information and rearranging matrices is assumed to be free.
2. An **adjacent transposition** τ_i is a special permutation that swaps i with $i+1$ and leaves everything else fixed.
3. YOR has the special property that the representation matrices of adjacent transpositions are very sparse: $\rho_\lambda(\tau_i)$ has at most 2 non-zero entries in each row and in each column.
4. Defining $[\![i, n]\!]$ as the permutation

$$[\![i, n]\!](j) = \begin{cases} j & \text{for } \; j = 1, \ldots, i-1 \\ j+1 & \text{for } \; j = i, \ldots, n-1 \\ i & \text{for } \; j = n, \end{cases}$$

the cosets $\{[\![i,n]\!]\,\mathbb{S}_{n-1}\}_{i=1}^{n}$ form a partition of $\mathbb{S}_n$, therefore any $\sigma \in \mathbb{S}_n$ can be written as $\sigma = [\![i,n]\!]\tau$ for some $i \in \{1,2,\ldots,n\}$ and $\tau \in \mathbb{S}_{n-1}$. Furthermore, $[\![i,n]\!] = \tau_i\,\tau_{i+1}\ldots\tau_{n-1}$.

Proposition 4 *If $f\colon \mathbb{S}_n \to \mathbb{R}$ is defined as in (8), then in YOR its Fourier transform can be computed in*

$$\left[\frac{(n+1)\,n\,(n-1)}{3} - \frac{(m+1)\,m\,(m-1)}{3}\right]\frac{n!}{m!} \quad (15)$$

scalar operations, where m is a shorthand for $n-k$. For fixed k this expression grows as $O(n^{2+k})$.

Proof. (Sketch) Clausen's fast Fourier transform (Clausen, 1989) for general functions on f is based on the observation that thanks to (12) and the coset decomposition $\bigcup_{i=1}^{n}[\![i,n]\!]\,\mathbb{S}_{n-1} = \mathbb{S}_n$, if we define the restricted functions $f_i\colon \mathbb{S}_{n-1} \to \mathbb{R}$ as $f_i(\tau) = f([\![i,n]\!]\,\tau)$, then

$$\widehat{f}(\lambda) = \sum_{i=1}^{n}\sum_{\tau\in\mathbb{S}_{n-1}} \rho_\lambda([\![i,n]\!]\,\tau)\,f([\![i,n]\!]\,\tau) = \sum_{i=1}^{n}\rho_\lambda([\![i,n]\!])\sum_{\tau\in\mathbb{S}_{n-1}}\rho_\lambda(\tau)\,f_i(\tau) = \sum_{i=1}^{n}\rho_\lambda([\![i,n]\!])\bigoplus_{\substack{\lambda^-\vdash n-1\\ \lambda^-\le\lambda}}\widehat{f}_i(\lambda^-), \quad (16)$$

where $\widehat{f}_i$ are now Fourier transforms on the smaller group $\mathbb{S}_{n-1}$. If t is a standard tableau of shape λ, letting $[\rho_\lambda(\sigma)]_t$ denote the column of $\rho_\lambda(\sigma)$ indexed by t, and similarly for $[\widehat{f}(\lambda)]_t$, specializing (16) to column t gives

$$[\widehat{f}(\lambda)]_t = \sum_{i=1}^{n}\rho_\lambda([\![i,n]\!])\,\widehat{h}_t^{(i)},$$

where $h_t^{(i)}$ is a column vector of all zeros, except for the block encoding $[\widehat{f}_i(\lambda^-)]_{t^-}$, where t^- is the tableau that we get from t by removing n. If $h_i^{(t)}$ is d_t dimensional, then multiplying it by $\rho_\lambda([\![i,n]\!])$ can be done in $2d_t(n-i)$ operations, since $\rho_\lambda([\![i,n]\!]) = \rho_\lambda(\tau_i)\ldots\rho_\lambda(\tau_{n-1})$, and in YOR each of the matrices in this product has at most two non-zero entries in each row. Summing over i gives $d_t n(n-1)$, and then summing over over all $\{\, t\in\mathcal{T}_n \mid t \ge \begin{array}{|c|c|c|c|c|}\hline 1&2&&&*\\\hline\end{array}\,\}$ and using Proposition 2 gives a total of $n(n-1)\,n!/(n-k)!$ operations for computing $\widehat{f}$ from the lower level transforms $\{\widehat{f}_i\}$.

Now we can recurse on the $\mathbb{S}_{n-1}$–transforms $\{\widehat{f}_i\}$ and show that computing each of them from the appropriate $\mathbb{S}_{n-2}$–transforms takes $(n-1)(n-2)\,(n-1)!/(n-k)!$ operations, and so on, down to level $n-k$. Starting the flow of information at the bottom, in total this recursive algorithm requires $\sum_{j=n-k+1}^{n} j\,(j-1)\,\frac{n!}{(n-k)!}$ scalar operations, which evaluates to (15). ■

We remark that using the FFT described in (Maslen, 1997) could further reduce the asymptotic complexity of computing $\widehat{f}$ by a factor of n.

4.3. Further sparsification and implementation

For small k Propositions 2–4 reduce computing the graphlet spectrum into the realm of possibility. However, in realistic scenarios some further computational savings are possible.

In the "f domain" we can exploit the fact that real world graphs are generally sparse, the locations where graphlets match them are even sparser, hence we never have to store a dense indicator function (6) in memory, and crucially, in performing the FFT we can ignore entire branches in the implied tree of cosets.

As a counterpart in Fourier space, for $k \ge 4$ it is helpful to ignore some of the "higher order" Fourier matrices, which quickly become infeasibly large. In general, the Fourier matrices that we are concerned with are of shape $\lambda = (n-s, \lambda_2, \lambda_3, \ldots \lambda_j)$ with $s << n$, and the dimensionality of such matrices grows with n^s. Imposing a cutoff at $s = 3$ or $s = 4$ not only reduces the storage requirements of $\widehat{f}$, and consequently of the graphlet spectrum $\widehat{q}$, but also allows us to eliminate all branches leading to these components in the computation of the FFT.

In practice, exploiting these dual sources of sparsity in tandem affords much better scaling behavior (albeit with no theoretical guarantees) than suggested by Propositions 2 and 4, and opens the door to exploring larger graphlets. From an implementation point of view this was made possible by significantly extending the $\mathbb{S}_n$`OB` FFT library (Kondor, 2006). We plan to make all the code necessary for computing the graphlet spectrum available on the $\mathbb{S}_n$`OB` web site.

5. Experiments

We assess the performance of the graphlet spectrum features on four benchmark datasets of chemical structures of molecules: MUTAG, ENZYMES, NCI1, and NCI109. Of these, MUTAG (Debnath et al., 1991) is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds labeled according to whether or not they have a mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*. ENZYMES is a dataset of protein tertiary structures which we ob-

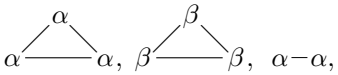
Table 2. Prediction accuracy in percent for the graphlet spectrum features and state-of-the-art graph kernels on four classification benchmarks in 10 repetitions of 10-fold cross-validation. Standard errors are indicated in parentheses. Best results for each datasets are in bold.

	MUTAG	ENZYMES	NCI1	NCI109
Number of instances/classes	188/2	600/6	4110/2	4127/2
Max. number of nodes	28	126	111	111
Graphlet spectrum	88.11 (0.46)	**35.42** (0.58)	**65.0** (0.09)	**65.31** (0.08)
Reduced skew spectrum	**88.61** (0.21)	25.83 (0.34)	62.72 (0.05)	62.62(0.03)
Graphlet count kernel	81.7 (0.67)	23.94 (0.4)	54.34 (0.04)	52.39 (0.09)

tained from (Borgwardt et al., 2005) consisting of 600 enzymes from the BRENDA enzyme database (Schomburg et al., 2004). In this case the task is to correctly assign each enzyme to one of the 6 EC top level classes. The average number of nodes of the graphs in this dataset is 32.6 and the average number of edges is 124.3. Finally, we also conducted experiments on two balanced subsets of NCI1 and NCI109, which classify compounds based on whether or not they are active in an anti-cancer screen ((Wale & Karypis, 2006) and `http://pubchem.ncbi.nlm.nih.gov`). Since in these datasets the number of vertices varies from graph to graph, while the graphlet spectrum requires a fixed n, we set n to be the maximum over the entire dataset and augment the smaller graphs with the appropriate number of unconnected "phantom" nodes.

The experiments consisted of running SVMs on the above data using a linear kernel on top of the the graphlet spectrum features. For comparison, we applied a linear kernel on the reduced skew spectrum features from (Kondor & Borgwardt, 2008) and a graphlet count kernel that counts the number of common graphlets in two graphs (Shervashidze et al., 2009). Both these kernels had been shown to outperform the classic random walk kernel (Gärtner et al., 2003) in earlier studies.

One of the strengths of the graphlet spectrum is that it allows the practitioner to use graphlets specifically designed to pick out salient features, such as functional groups in molecules. In our experiments we started with a minimal set of graphlets and saw performance increase as we added further ones one by one. The actual graphlets used in our experiments were the following:

- MUTAG: C−C, C−C−C, C−C−C−C, C−N, O−N, ∗−∗;
- NCI1 and NCI109: C−C, C−N, C−O, O−N, O−O, N−N;
- ENZYMES: ∗−∗,  , α−α, α−β, β−β;

where $* - *$ denotes an edge with arbitrary node labels. For fair comparison in the graphlet count kernel we used the same graphlets. Further experimentation and incorporating more knowledge from chemistry could lead to a siginificantly more powerful system of graphlets for organic molecules. It is important to stress that computational time, while a constraint, was not the limiting factor here: computing the spectrum with the above graphlets on MUTAG took about a second per graph on a desktop machine, and the system could easily handle several more graphlets of up to 4 or even 5 vertices. For enzymes α and β denote α-helices and β-sheets, respectively.

To evaluate performance, we tested prediction accuracy on independent evaluation sets which we obtained as follows. We split each dataset into 10 folds of identical sizes. We then split 9 of these folds again into 10 parts, trained a C-SVM (implemented by LIBSVM (Chang & Lin, 2001)) on 9 parts, and predicted on the 10th part. We repeated this training and prediction procedure for $C \in \{10^{-7}, 10^{-6}, \ldots, 10^{7}\}$, and determined the C reaching maximum prediction accuracy on the 10th part. We then trained an SVM with this best C on all 9 folds (= 10 parts), and predicted on the 10th fold, which acts as an independent evaluation set. We repeated the whole procedure 10 times so that each fold acts as independent evaluation set exactly once. For each dataset and each method, we repeated the whole experiment 10 times and report mean accuracy levels and standard errors in Table 2.

On the whole, the graphlet spectrum outperforms both its comparison partners in our experiments. Its accuracy is more than 2% higher than that of the reduced skew spectrum on both NCI datasets, and almost 10% better on ENZYMES. Only on MUTAG is the skew spectrum's accuracy slightly better than that of the graphlet spectrum (88.61% vs. 88.11%).

Most interestingly, the graphlet spectrum always outperforms the graphlet count-based kernels. Its ability to consider relative positions between graphlets seems to lead to a much more sophisticated measure of structural graph similarity than pure subgraph frequencies. Even when the graphlet count based approach's performance is barely better than random, as on NCI109,

the graphlet spectrum still achieves state-of-the-art results based on the same graphlets.

6. Discussion

In this paper we have presented a new, efficiently computable system of graph invariants for use in graph kernels called the *graphlet spectrum*. The graphlet spectrum is based on k'th order subgraphs ("graphlets"), and to the best of our knowledge it is the first practical system of graph invariants that not only counts subgraphs, but also takes their relative position into account. A further advantage of the new approach is that it can encode vertex and edge labels in addition to the graph topology.

Experiments show that on graphs of medium size (up to a few hundred vertices) the graphlet spectrum is comparable in performance with state-of-the-art graph kernels, and in several cases outperforms all other methods. Theoretical results from non-commutative harmonic analysis and the representation theory of $\mathbb{S}_n$, together with a custom-built FFT library allow the graphlet spectrum to scale up to real-world problems with relative ease.

One of the sources of flexibility as well as one of the burdens associated with the graphlet spectrum is having to specify a library of graphlets. In our experiments we solved this by using domain knowledge, defining a system of graphlets specifically tailored to organic molecules. However, automatic graphlet selection approaches are also conceivable, leading to the issue of efficient feature selection on graphs, such as the work by Tsuda (2007) on feature selection on frequent subgraphs.

References

Bach, F. (2008). Graph kernels between point clouds. *Proc. Intl. Conf. Machine Learning* (pp. 25–32).

Borgwardt, K. M., & Kriegel, H.-P. (2005). Shortest-path kernels on graphs. *Proc. Intl. Conf. Data Mining* (pp. 74–81).

Borgwardt, K. M., Ong, C. S., Schonauer, S., Vishwanathan, S. V. N., Smola, A. J., & Kriegel, H. P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, *21*, i47–i56.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Clausen, M. (1989). Fast generalized Fourier transforms. *Theor. Comput. Sci.*, 55–63.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., & Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J Med Chem*, *34*, 786–797.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *Proc. Annual Conf. Computational Learning Theory* (pp. 129–143). Springer.

Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *Proc. Intl. Conf. Machine Learning* (pp. 321–328). San Francisco, CA: Morgan Kaufmann.

Kondor, R. (2006). $\mathbb{S}_n$`ob`: a C++ library for fast Fourier transforms on the symmetric group. Available at `http://www.gatsby.ucl.ac.uk/~risi/Snob/`.

Kondor, R., & Borgwardt, K. (2008). The skew spectrum of graphs. *Proc. Intl. Conf. Machine Learning* (pp. 496–503).

Maslen, D. K. (1997). The computation of Fourier transforms on the symmetric group. *Math. Comp*, *67*, 1121–1147.

Mikkonen, T. (2007). The ring of graph invariants – graphic values. arXiv: 0712/0146.

Rockmore, D. N. (1997). Some applications of generalized FFTs. *Proceedings of the DIMACS workshop on groups and computation*.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., & Schomburg, D. (2004). Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research*, *32D*, 431–433.

Serre, J.-P. (1977). *Linear representations of finite groups*, vol. 42 of *Graduate Texts in Mathamatics*. Springer-Verlag.

Shervashidze, N., Vishwanathan, S. V. N., Petri, T., Mehlhorn, K., & Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. *Proceedings of International Conference on Artificial Intelligence and Statistics*.

Tsuda, K. (2007). Entire regularization paths for graph data. *Proc. Intl. Conf. Machine Learning* (pp. 919–926).

Wale, N., & Karypis, G. (2006). Comparison of descriptor spaces for chemical compound retrieval and classification. *Proc. Intl. Conf. Data Mining* (pp. 678–689).