Spectral Clustering and Transductive Learning with Multiple Views

Dengyong Zhou DENGYONG.ZHOU@MICROSOFT.COM
Christopher J.C. Burges CBURGES@MICROSOFT.COM
Microsoft Research, One Microsoft Way, Redmond, WA 98052

Abstract

We consider spectral clustering and transductive inference for data with multiple views. A typical example is the web, which can be described by either the hyperlinks between web pages or the words occurring in web pages. When each view is represented as a graph, one may convexly combine the weight matrices or the discrete Laplacians for each graph, and then proceed with existing clustering or classification techniques. Such a solution might sound natural, but its underlying principle is not clear. Unlike this kind of methodology, we develop multiview spectral clustering via generalizing the normalized cut from a single view to multiple views. We further build multiview transductive inference on the basis of multiview spectral clustering. Our framework leads to a mixture of Markov chains defined on every graph. The experimental evaluation on real-world web classification demonstrates promising results that validate our method.

1. Introduction

In the general machine leaning problem setting, we often assume that the data are represented in a single vector space or by a single graph. In many read-life problems, however, the same instances may be represented in several different vector spaces, or by several different graphs, or even as a mixture of vector spaces and graphs. Hence, we consider learning from instances which have multiple representations, and our goal is to effectively explore and exploit multiple representations simultaneously. This kind of machine learning issue is often called multiview learning (Rüping & Scheffer, 2005).

A typical example is web categorization in which the web can be described by either hyperlinks between web pages or the words occurring in web pages. In the former description, the web can be represented as a directed graph where each vertex represents a web page, and each directed edge a hyperlink. In the latter description, each web page is represented as a vector in Euclidian space, and each element in the vector typically responds to the occurrences of some word. For combing these two different representations, one can consider weighting the hyperlink graph by using the text vectors. Specifically, given a pair of linked web pages, the hyperlink is weighted by the similarity measure based on the dot product between the text vectors of those two given web pages. This methodology is overly link-centered since the dot product based similarity between two unlinked web pages are not taken into account.

Multiview learning occurs in many other situations. In scientific publication classification, we can build a citation network over the articles, where each node indicates an article, and each directed link a citation from one article to another. Moreover, we can also build a coauthor network over the articles, where there is a link between two articles if they have an author in common. In social network analysis, there are multiple types of relationships among individuals. For example, they can be email networks, organization hierarchy, collaboration and so on. As in web categorization, for clustering or classifying scientific publications or individuals, we need to consider how to utilize several networks together rather than a single network only.

A natural approach to multiview learning is to define a kernel for each type of data representation, and then convexly combine those kernels (Joachims et al., 2001; Zhang et al., 2006). In web categorization, the kernel for the link graph can be defined as the colink matrix, in which two web pages have a colink if a third web page points to both of them. In the kernel for

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

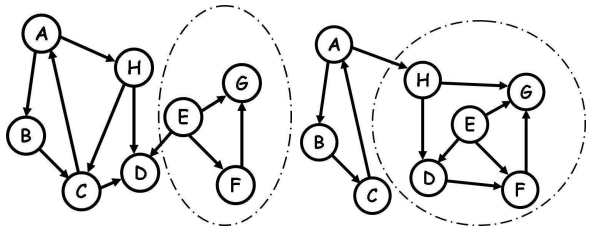

Figure 1. Two directed graphs sharing the same set of vertices. The large circle on each panel denotes the clustering result with respect to each graph. Obviously, the clustering is *good* for one graph while being *bad* for the other graph. Thus we consider how to find a clustering which is close to optimal for both graphs.

text, each entry can be given by the inner product between the corresponding document vectors. If the multiple views are represented by different graphs, the inverse of the graph Laplacian can be regarded as a kernel (Smola & Kondor, 2003), and, consequently, one may convexly combine graph Laplacians (Tsuda et al., 2005; Sindhwani et al., 2005; Argyriou et al., 2006). The underlying principle of this methodology is unclear however. It is well-known that the spectral clustering approach for a single graph is derived from a real-valued relaxation of the combinatorial normalized cut which naturally leads to the graph Laplacian (Shi & Malik, 2000), but we have not yet noticed any literature addressing the issue of which combinatorial cut can lead to the convex combination of graph Laplacians in the situation of multiple graphs.

Unlike the above methodology, we develop multiview spectral clustering via generalizing the usual single view normalized cut (Section 2) to the multiview case (Section 3). The basic motivation behind the multiview normalized cut is that we try to find a cut which is close to optimal on each graph (Figure 1). As in the single view case, this multiview normalized cut can be approximately optimized via a real-valued relaxation. The relaxation does not lead to convex combination of graph Laplacians. Instead, it results in a vertex-wise mixture of Markov chains associated with different graphs. In Section 5, the multiview spectral clustering approach is extended to multiview transductive classification via a regularization framework. The experimental results on web categorization are shown in Section 6. We conclude the paper in Section 7.

It is worth mentioning a popular method for multiview learning called co-training. In this method, multiple learning algorithms are trained on each view, and then each algorithm's prediction on new unlabeled examples are used to enlarge the training sets of others (Blum & Mitchel, 1998). This approach needs somewhat strong independence assumptions; a detailed discussion is beyond the scope of this paper, however.

2. Spectral Clustering with a Single Graph

In this section, we review the spectral clustering approach for directed graphs (Zhou et al., 2005). This approach naturally generalizes the spectral undirected graph clustering scheme (Shi & Malik, 2000; Meila & Shi, 2001; Ng et al., 2002). An undirected graph is a special case of a directed graph, that is, an edge of an undirected graph connecting vertex u to vertex v represents two directed edges, one from u to v, and the other from v to u.

The issue of spectral directed graph clustering has attracted a lot of research interest, in particular, in the web search engine community (Henzinger, 2003; Kleinberg, 1999). Compared with other directed graph clustering techniques, the key insight in the approach of Zhou et al. (2005) is to regard a directed graph as a Markov chain. In other words, their approach is built on random walks over a directed graph rather than directly manipulating the combinatorial structures of the directed graph. It is worth noting that there are a wide choice of random walks given a directed graph, and different random walk choices generally lead to different clustering results.

Given a directed graph $G = (V, E, w)$ with vertex set V, edge set E, and corresponding edge weights w, assume a random walk defined on G with transition probabilities p and stationary distribution π. Let S denote an arbitrary subset of V, and S^c the complement of S. Define the volumes

$$\mathrm{vol}\, S = \sum_{v \in S} \pi(v), \text{ and } \mathrm{vol}\, \partial S = \sum_{u \in S, v \in S^c} \pi(u) p(u, v). \tag{1}$$

We can verify that $\mathrm{vol}\, S + \mathrm{vol}\, S^c = 1$, and $\mathrm{vol}\, \partial S = \mathrm{vol}\, \partial S^c$. Then a clustering can be obtained by

$$\underset{\emptyset \neq S \subset V}{\mathrm{argmin}} \left\{ c(S) = \frac{\mathrm{vol}\, \partial S}{\mathrm{vol}\, S\, \mathrm{vol}\, S^c} \right\}. \tag{2}$$

The intuition behind this cut is as follows. Assume a random web surfer who browses web pages by following hyperlinks and occasionally jumping to a randomly chosen web page. Then the web surfer will regard a set of hyperlinked web pages as a community if the probability of leaving the web page set is small while the stationary probability mass of the same subset is large.

Note that, in the discrete optimization (2), the subset

S and its complement S^c are not themselves required to be connected. Hence, one may wonder if there exist subsets satisfying (2) while they are not connected. Although this question looks natural, it has not been addressed since spectral clustering was developed. Let us investigate this problem via showing the following argument.

Theorem 2.1. *Let G be a connected graph. Then for any partition $V = S \cup S^c$ satisfying (2), both S and S^c are connected.*

Proof. Assume S to be the union of $k \geq 2$ connected components S_i. Let $\tau = \min_i c(S_i)$. Then

$$\operatorname{vol} \partial S = \sum_{i \leq k} \operatorname{vol} \partial S_i \geq \tau \sum_{i \leq k} \operatorname{vol} S_i \operatorname{vol} S_i^c.$$

Note that $\operatorname{vol} S_i^c = 1 - \operatorname{vol} S_i$. Then

$$\begin{aligned} \operatorname{vol} \partial S &\geq \tau \sum_{i \leq k} \operatorname{vol} S_i - \operatorname{vol}^2 S_i \\ &= \tau \Big(\operatorname{vol} S - \sum_{i \leq k} \operatorname{vol}^2 S_i \Big). \end{aligned}$$

Hence

$$\begin{aligned} c(S) &= \frac{\operatorname{vol} \partial S}{\operatorname{vol} S \operatorname{vol} S^c} = \frac{\operatorname{vol} \partial S}{\operatorname{vol} S - \operatorname{vol}^2 S} \\ &\geq \tau \frac{\operatorname{vol} S - \sum_{i \leq k} \operatorname{vol}^2 S_i}{\operatorname{vol} S - \operatorname{vol}^2 S}. \end{aligned}$$

Note that, for $k \geq 2$,

$$\operatorname{vol}^2 S = \big(\sum_{i \leq k} \operatorname{vol} S_i \big)^2 \geq \sum_{i \leq k} \operatorname{vol}^2 S_i.$$

Therefore, $c(S) > \tau$, which is in contradiction with our initial assumption that S is a solution to (2). □

The combinatorial optimization problem (2) can be approximately solved by relaxing it into a real-valued problem

$$\begin{aligned} &\operatorname*{argmin}_{f \in \mathbb{R}^{|V|}} \left\{ \sum_{u \in V, v \in V} \pi(u) p(u,v) \left(f(u) - f(v) \right)^2 \right\} \quad (3) \\ &\text{subject to } \sum_{v \in V} f^2(v) \pi(v) = 1, \ \sum_{v \in V} f(v) \pi(v) = 0. \end{aligned}$$

Let P denote the transition probability matrix with its elements being $p(u,v)$, and Π the diagonal matrix with its diagonal elements being $\pi(u)$. Define a matrix

$$L = \Pi - \frac{\Pi P + P^T \Pi}{2}. \quad (4)$$

Then the clustering which satisfies the cut criterion can be approximately obtained via solving the generalized eigenvector system

$$Lf = \lambda \Pi f,$$

where λ is the second smallest eigenvalue.

It is easy to extend binary partition to k-partition. Assume a k-partition to be $V = V_1 \cup V_2 \cup \cdots \cup V_k$, where $V_i \cap V_j = \emptyset$ for all $1 \leq i, j \leq k$. Let P_k denote a k-partition. Then we can obtain a k-partition by minimizing

$$c(P_k) = \sum_{1 \leq i \leq k} \frac{\operatorname{vol} \partial V_i}{\operatorname{vol} V_i}. \quad (5)$$

We can check that Equation (2) is a special case of Equation (5) with $k = 2$. Moreover, the solution of the corresponding relaxed optimization problem of (5) can be any orthonormal basis for the linear space spanned by the generalized eigenvectors of L pertaining to the k smallest eigenvalues.

3. Spectral Clustering with Multiple Graphs

Assume two directed graphs $G_i = (V, E_i, w_i), i = 1, 2$, which share the same set of vertices while having different edges and weights. Suppose S to be a nonempty subset of V. Define

$$\operatorname{mvol} S = \alpha \operatorname{vol}_1 S + (1 - \alpha) \operatorname{vol}_2 S, \quad (6)$$

and

$$\operatorname{mvol} \partial S = \alpha \operatorname{vol}_1 \partial S + (1 - \alpha) \operatorname{vol}_2 \partial S, \quad (7)$$

where α is a parameter in $[0, 1]$. Then we may cluster the vertex set V into two subsets by

$$\operatorname*{argmin}_{\emptyset \neq S \subset V} \left\{ c(S) = \frac{\operatorname{mvol} \partial S}{\operatorname{mvol} S \operatorname{mvol} S^c} \right\}. \quad (8)$$

Clearly, the case of $\alpha = 0$ or 1 reduces to the cut for a single graph.

The basic motivation of defining such a multiple graph cut is that we want to obtain a cut which is good on average while it may not be the best for a single graph (Figure 1). The parameter α is used to specify the relative importance of each graph in clustering. It is not hard to imagine that the relative importance measure varies across different clustering goals. Let us consider a somewhat extreme example to illustrate this point. When we seek to cluster scientists into different groups such that in each group scientists have some research interest in common, then the co-author relationship

will dominate over other kinds of relationships. However, if we hope the scientists in the same group share the same political point of view, then the coauthor relationship may not be helpful, and in fact, could be misleading.

Algorithm 1 Spectral clustering with multiple graphs

Given k graphs $G_i = (V, E_i, w_i), 1 \le i \le k$, which are directed or undirected, and share the same vertex set V, the vertices in V can be clustered into two subsets as follows.

1. For each graph G_i, associate it with a random walk which has a unique stationary distribution. Denote by p_i the transition probabilities, and π_i the stationary distribution satisfying
$$\sum_{u\in V} \pi_i(u)p_i(u,v) = \pi_i(v).$$
2. Define a mixture of those random walks by
$$p(u,v) = \sum_{i\le k} \beta_i(u)p_i(u,v),$$
where
$$\beta_i(u) = \frac{\alpha_i\pi_i(u)}{\sum_{j\le k}\alpha_j\pi_j(u)}, \text{ and } \sum_{j\le k}\alpha_j = 1,\ \alpha_j \ge 0.$$
The random walk mixture has a unique stationary distribution given by
$$\pi(v) = \sum_{i\le k}\alpha_i\pi_i(v).$$
3. Denote by P the matrix with the elements $p(u,v)$, and Π the diagonal matrix with the diagonal elements $\pi(u)$. Form the matrix
$$L = \Pi - \frac{\Pi P + P^T\Pi}{2}.$$
4. Compute the generalized eigenvector satisfying
$$Lf = \lambda\Pi f$$
where λ is the second smallest eigenvalue, and cluster the vertex set V into the two parts $S = \{v \in V | f(v) \ge 0\}$ and $S^c = \{v \in V | f(v) < 0\}$.

In what follows, we construct a Markov mixture model, and explain the multiple graph cut in terms of a random walk. Define functions

$$\beta_1(u) = \frac{\alpha\pi_1(u)}{\alpha\pi_1(u) + (1-\alpha)\pi_2(u)}, \tag{9}$$

and

$$\beta_2(u) = \frac{(1-\alpha)\pi_2(u)}{\alpha\pi_1(u) + (1-\alpha)\pi_2(u)}. \tag{10}$$

So that $\beta_1(u)+\beta_2(u) = 1$ and $\beta_i \ge 0$. Then define new transition probabilities among vertices as

$$p(u,v) = \beta_1(u)p_1(u,v) + \beta_2(u)p_2(u,v). \tag{11}$$

Note that β_1 and β_2 vary from vertex to vertex rather than being a constant. Therefore the above formula is not simply a linear combination of the transition probability matrices on each graph.

From a straightforward computation, we can check that the stationary distribution of the Markov mixture model is

$$\pi(v) = \alpha\pi_1(v) + (1-\alpha)\pi_2(v). \tag{12}$$

Consequently, we have

$$\begin{aligned} \operatorname{mvol} S &= \alpha\sum_{v\in S}\pi_1(v) + (1-\alpha)\sum_{v\in S}\pi_2(v) \\ &= \sum_{v\in S}\pi(v) = P(S). \end{aligned}$$

Similarly, $\operatorname{mvol} S^c = P(S^c)$. Moreover,

$$\begin{aligned} \operatorname{mvol}\partial S &= \alpha\sum_{(u,v)\in\partial_1 S}\pi_1(u)p_1(u,v) + \\ &\quad (1-\alpha)\sum_{(u,v)\in\partial_2 S}\pi_2(u)p_2(u,v) \\ &= \sum_{u\in S, v\in S^c}(\alpha\pi_1(u) + (1-\alpha)\pi_2(u))\cdot \\ &\quad \left(\frac{\alpha\pi_1(u)}{\alpha\pi_1(u)+(1-\alpha)\pi_2(u)}p_1(u,v)\right. \\ &\quad \left. + \frac{(1-\alpha)\pi_2(u)}{\alpha\pi_1(u)+(1-\alpha)\pi_2(u)}p_2(u,v)\right) \\ &= \sum_{u\in S, v\in S^c}\pi(u)p(u,v) \\ &= P(S\to S^c). \end{aligned}$$

Similarly, $\operatorname{mvol}\partial S^c = P(S^c \to S)$. It can be verified that

$$P(S^c\to S) = P(S\to S^c).$$

Thus we have $\operatorname{mvol}\partial S^c = \operatorname{mvol}\partial S$. Hence

$$\begin{aligned} c(S) &= \frac{\operatorname{mvol}\partial S(\operatorname{mvol} S + \operatorname{mvol} S^c)}{\operatorname{mvol} S \operatorname{mvol} S^c} \\ &= \frac{\operatorname{mvol}\partial S}{\operatorname{mvol} S} + \frac{\operatorname{mvol}\partial S^c}{\operatorname{mvol} S^c} \\ &= P(S\to S^c|S) + P(S^c\to S|S^c). \end{aligned}$$

Now the multiple graph cut can be understood as follows. Assume a random walk with the current position being at a vertex in one graph. Then, in the next step, the walker may continue his random walk in the same graph with a certain probability, or jump to the other graph with the remaining probability and continue his random walk there. A subset of vertices is regarded as a cluster if during the random walk the probability of leaving this subset is small while the stationary probability mass of the same subset is large. It is obvious how to extend the above analysis to more than two graphs. Finally, we summarize the multiple graph spectral clustering in Algorithm 1.

4. Multiple Undirected Graphs

In this section, we discuss the special case of multiple undirected graphs. Assume two undirected graphs $G_i = (V, E_i, w_i), i = 1, 2$. Given a vertex $v \in V$, denote by $d_i(v) = \sum_u w_i(u, v)$. With respect to each graph G_i, define the boundary of S as $\partial_i S = \{(u, v) \in E_i | u \in S, v \in S^c\}$, and the volume of S as $\mathrm{vol}_i\, S = \sum_{u\in S, v\in S} w_i(u, v) = \sum_{v\in S} d_i(v)$, and the volume of ∂S as $\mathrm{vol}_i\, \partial_i S = \sum_{(u,v)\in\partial_i S} w_i(u, v)$. All of those definitions are taken from (Chung, 1997).

We then define the multiview boundary of S with respect to both G_1 and G_2 by $\partial S = \partial_1 S \cup \partial_2 S$, and the multiview volume of S as

$$\mathrm{mvol}\, S = \alpha \frac{\mathrm{vol}_1\, S}{\mathrm{vol}_1\, V} + (1-\alpha)\frac{\mathrm{vol}_2\, S}{\mathrm{vol}_2\, V},$$

and the multiview volume of ∂S as

$$\mathrm{mvol}\, \partial S = \alpha \frac{\mathrm{vol}_1\, \partial_1 S}{\mathrm{vol}_1\, V} + (1-\alpha)\frac{\mathrm{vol}_2\, \partial_2 S}{\mathrm{vol}_2\, V}.$$

It is easy to check that $\mathrm{mvol}\, S + \mathrm{mvol}\, S^c = 1$.

Note the volume based normalization in the above definitions. The normalization is necessary because the weights on different graphs can be measured in very different scales. For instance, in web categorization, we can form at least two undirected graphs. One is the colink graph (Joachims et al., 2001), and the other is a fully connected undirected graph weighted by the textual kernel. In addition, the normalization leads to the Markov mixture model. Let us consider the natural random walk on undirected graphs. The transition probabilities are $p_i(u, v) = w_i(u, v)/d_i(u)$, and the stationary probabilities $\pi_i(u) = d_i(u)/\mathrm{vol}_i\, V$. Then

$$\beta_1(u) = \frac{\alpha d_1(u)/\mathrm{vol}_1\, V}{\alpha d_1(u)/\mathrm{vol}_1\, V + (1-\alpha)d_2(u)/\mathrm{vol}_2\, V},$$

and

$$\beta_2(u) = \frac{(1-\alpha) d_2(u)/\mathrm{vol}_1\, V}{\alpha d_1(u)/\mathrm{vol}_1\, V + (1-\alpha)d_2(u)/\mathrm{vol}_2\, V}.$$

Thus

$$\begin{aligned} p(u,v) &= \beta_1(u)p_1(u,v) + \beta_2(u)p_2(u,v) \\ &= \frac{\alpha w_1(u,v)/\mathrm{vol}_1\, V}{\alpha d_1(u)/\mathrm{vol}_1\, V + (1-\alpha)d_2(u)/\mathrm{vol}_2\, V} \\ &\quad + \frac{(1-\alpha) w_2(u,v)/\mathrm{vol}_1\, V}{\alpha d_1(u)/\mathrm{vol}_1\, V + (1-\alpha)d_2(u)/\mathrm{vol}_2\, V} \end{aligned}$$

and $\pi(u) = \alpha d_1(u)/\mathrm{vol}_1\, V + (1-\alpha)d_2(u)/\mathrm{vol}_2\, V$.

Introducing

$$w(u, v) = \alpha \frac{w_1(u, v)}{\mathrm{vol}_1\, V} + (1-\alpha)\frac{w_2(u, v)}{\mathrm{vol}_2\, V},$$

and $d(u) = \pi(u)$, we then have $p(u, v) = w(u, v)/d(u)$. This means that, in the special case of multiple undirected graphs, the Markov mixture model reduces to a linear combination of adjacency matrices or a convex combination of normalized adjacency matrices. Hence, it is different from the approaches which convexly combine undirected graph Laplacians via $L = \alpha L_1 + (1-\alpha)L_2$ without any stated reasons (Tsuda et al., 2005; Sindhwani et al., 2005; Argyriou et al., 2006). In that literature, the Laplacian matrix for undirected graphs is defined to be $L_i = D_i - W_i$, where D_i is a diagonal matrix with its diagonal elements being $d_i(u)$, and W_i is the weight matrix with its each element being $w_i(u, v)$.

5. Classification with Multiple Graphs

In some sense, it is straightforward to build a transductive inference algorithm from a clustering approach. Let us first consider classification on a single graph. Assume we have a directed graph $G = (V, E, w)$, and a discrete label set $\mathcal{L} = \{-1, 1\}$. The vertices in a subset $S \subset V$ have been classified as 1 or -1. Our task is to predict the labels of the remaining unclassified vertices.

Let $f : V \to S$ denote the classification function. Define a function y with $y(v) = 1$ or -1 if $v \in S$, and 0 if v is unlabeled. Then we can choose a classification function via

$$\begin{aligned} \underset{f\in\mathbb{R}^{|V|}}{\mathrm{argmin}} \Bigg\{ &\sum_{u\in V, v\in V} \pi(u)p(u,v)\left(f(u) - f(v)\right)^2 \\ &+ C\sum_{v\in V} \pi(v)\left(f(v) - y(v)\right)^2 \Bigg\}. \end{aligned} \tag{13}$$

where $C > 0$. Note that the first term in the optimization problem is the objective function for clustering in Equation (3). Intuitively, the objective function for classification forces the classification function to

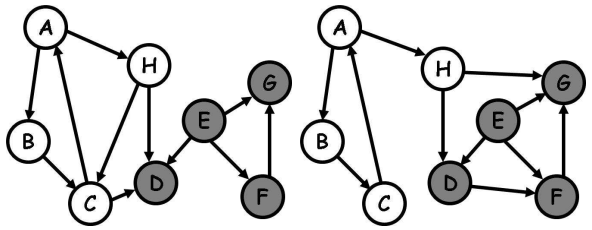

Figure 2. Classification with multiple graphs. There are two directed graphs over the same set of vertices. Those vertices belong to two different classes respectively denoted by gray and white circles.

change as slowly as possible on densely connected subgraphs. The second term in the above objective function forces the classification function to fit the given labels as well as possible. The tradeoff between these two requirements is measured by the parameter C.

If each function in $\mathbb{R}^{|V|}$ is scaled with a factor $\pi^{-1/2}$, then Equation (13) will be transformed into

$$\operatorname*{argmin}_{f\in\mathbb{R}^{|V|}}\Big\{\sum_{u\in V,v\in V}\pi(u)p(u,v)\Big(\frac{f(u)}{\sqrt{\pi(u)}}-\frac{f(v)}{\sqrt{\pi(v)}}\Big)^2 +C\sum_{v\in V}\left(f(v)-y(v)\right)^2\Big\}. \tag{14}$$

This is the classification approach proposed in (Zhou et al., 2005). However, Equation (13) somehow looks much more natural than Equation (14). For undirected graphs, Equation (14) reduces to the approach in (Zhou et al., 2004). There are several pieces of theoretic work around this approach (El-Yaniv & Pechyony, 2006; Ando & Zhang, 2007; El-Yaniv & Pechyony, 2007). See also (Chapelle et al., 2003; Joachims, 2003; Zhu et al., 2003; Smola & Kondor, 2003; Belkin et al., 2004) which are closely related to (Zhou et al., 2004).

To solve the optimization problem (13), we differentiate the objective function with respect to φ and obtain

$$(C\Pi+L)f=C\Pi y,$$

This linear system has the closed-form solution

$$f=C(C\Pi+L)^{-1}\Pi y.$$

However, since this linear system is positive definite and even diagonally dominant, we can avoid computing the inverse and instead use a fast solver such as (Spielman & Teng, 2003).

Classification with multiple graphs can be formalized as follows. Given a set of graphs $G_i=(V,E_i,w_i),\ 1\le i\le k$, with a vertex set V in common, and with the vertices in a subset $S\subset V$ labeled as 1 or -1, our goal is to predict the labels of the remaining unlabeled vertices in S^c (Figure 2). To extend the single graph based transduction to multiple graphs, the only thing we need to do is to construct the Markov mixture model used in the multiview spectral clustering in Section 3. For completeness, we summarize the multiview transduction in Algorithm 2. Note that in the algorithm we use a parameter $\gamma\in(0,1)$ instead of $C\in(0,\infty)$. The relationship between α and C can be

Algorithm 2 Classification with multiple graphs

Given k graphs $G_i=(V,E_i,w_i), 1\le i\le k$, which are directed or undirected, and which share the same vertex set V, assume that the vertices in a subset $S\subset V$ have been labeled as 1 or -1. The remaining unlabeled vertices can be classified as follows.

1. For each graph G_i, associate it with a random walk which has a unique stationary distribution. Denote by p_i the transition probabilities, and π_i the stationary distribution.

2. Define a mixture of those random walks by

$$p(u,v)=\sum_{i\le k}\beta_i(u)p_i(u,v),$$

where

$$\beta_i(u)=\frac{\alpha_i\pi_i(u)}{\sum_{j\le k}\alpha_j\pi_j(u)},\ \text{and}\ \sum_{j\le k}\alpha_j=1,\ \alpha_j\ge 0.$$

The random walk mixture has a unique stationary distribution given by

$$\pi(v)=\sum_{i\le k}\alpha_i\pi_i(v).$$

3. Denote by P the matrix with the elements $p(u,v)$, and Π the diagonal matrix with the diagonal elements $\pi(u)$. Form the matrix

$$M=\Pi-\gamma\frac{\Pi P+P^T\Pi}{2},$$

where γ is a parameter in $(0,1)$.

4. Define a function y on V with $y(v)=1$ or -1 if vertex v is labeled, and 0 if v is unlabeled. Solve the linear system

$$Mf=\Pi y,$$

and classify each unlabeled vertex v as sign $f(v)$.

Table 1. Precisions for four different approaches: content features based transduction; link structure based transduction; linearly combining graph Laplacians; and the Markov mixture model. The numbers in the fist line denote the proportion of labeled instances. Each precision result is averaged over 100 trials.

Labeled instances (%)	0.15	0.20	0.25	0.30	0.35	0.40	0.45
recall = 50%							
link	0.39	0.45	0.53	**0.61**	0.67	0.70	0.81
Content	0.37	0.43	0.45	0.50	0.62	0.72	**0.86**
Markov mixture	**0.43**	**0.47**	**0.55**	**0.61**	**0.68**	**0.73**	0.85
combining Laplacian	0.41	0.46	0.54	0.61	0.67	0.71	0.82
recall = 60%							
link	0.33	0.37	0.43	0.48	**0.57**	0.60	0.67
Content	0.31	0.35	0.37	0.40	0.47	0.51	0.59
Markov mixture	**0.36**	**0.41**	**0.45**	**0.50**	**0.57**	**0.62**	**0.68**
combining Laplacian	**0.36**	0.39	0.44	0.49	**0.57**	0.61	0.67

expressed as $\gamma = 1/(1 + C)$.

6. Experiments

We address the spam detection issue by using the dataset of `webspam-uk2006-1.2` (Castillo et al., 2006). This collection includes 77.9 million web pages over $11,452$ hosts. We consider the spam detection issue at the host level. In other words, we consider if a host is spam or not. The hosts in the dataset have been manually labeled as `normal`, `borderline`, `spam`, and `cannot judge`. Overall, 5.91% hosts are labeled as `spam`, and 43.45% hosts are labeled as `normal`. The remaining 50.69% hosts are `borderline` or `cannot judge`.

We build a directed graph over hosts as follows. Each host can be regarded a collection of web pages. Given two hosts, if there exists a hyperlink from some page on one host to some page on the other host, then we say that there is a directed edge between these two hosts. Moreover, it is weighted by the number of such edges. We can also describe each host by its content features which are useful in detecting if a host is spam or not. Each content feature of a host is built on the content features of the web pages contained by the host. The content features of a web page can be the fraction of anchor text, the fraction of visible text, the average word length, and so on. Hence each host can be represented as a feature vector. Then we normalize each feature vector, and the similarity between two hosts is measured as the inner product between the two corresponding feature vectors. Consequently a similarity graph is built over the hosts. Obviously the similarity graph is undirected.

In this dataset, $8,944$ hosts have been set up with content features. We further remove the hosts which are labeled as `borderline` or `cannot judge`. In other words, we only consider the hosts which are clearly judged as `normal` or `spam`. Then we extract the largest strongly connected subgraph from the subgraph consisting of those hosts. It has $2,922$ nodes, in which 156 are spam. The second largest strongly connected subgraph contains 21 nodes. In fact, over 96% of the strongly connected subgraphs have only a single node.

For both the host graph and the similarity graph, we use the natural random walk following the links uniformly at random. We compare the Markov mixture model with three other transductive classifications respectively based on the host graph, the similarity graph, and the convex combination of graph Laplacians. For the Markov mixture model and the graph Laplacian combination, each graph is treated equally, i.e. $\alpha_i = 0.5,\ i = 1, 2$. In further work, we will investigate how to choose α to measure the importance of different graphs.

Spam detection is a highly unbalanced classification problem. In our dataset, only 5.3% hosts are spam. Hence we measure algorithmic performances via precision/recall which is widely used in the Information Retrieval (IR) community. Precision is the ratio of the number of retrieved and relevant documents to the number of documents retrieved, and recall is the proportion of the number of relevant documents that are retrieved to the total number of the relevant documents available. In addition, classifying a normal host as spam is much worse than classifying a spam host as normal. That means precision is more crucial than recall. In our experiments we focus on the situation of low recall (Table 1). Since those approaches

are closely related to each other with subtle differences, their standard deviations are almost the same and around 0.005. From the experimental results, the Markov mixture model consistently performs better than the single view based approaches, and also better than convexly combining graph Laplacians. The improvement looks small, but it is still quite significant since spam detection is challenging in practice. Moreover, at web scale, even a one-point improvement can result in the discovery of a large amount of spam.

7. Conclusion

We have developed multiview spectral clustering and transductive inference approaches. The essential ingredient of this work is to form a mixture of Markov chains defined on different views. The experimental evaluation on real-world web classification problems demonstrates encouraging results that validate our approach. In addition to the web categorization issue addressed in this paper, the present methodology can be applied to a wide range of other practical problems including social network analysis and bioinformatics.

References

Ando, A., & Zhang, T. (2007). Learning on graph with Laplacian regularization. *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Argyriou, A., Herbster, M., & Pontil, M. (2006). Combining graph Laplacians for semi–supervised learning. *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA.

Belkin, M., Matveeva, I., & Niyogi, P. (2004). Regression and regularization on large graphs. *Proc. 17th Annual Conference on Learning Theory*.

Blum, A., & Mitchel, T. (1998). Combining labeled and unlabeled data with co-training. *Proc. Workshop on on Computational Learning Theory*.

Castillo, C., Donato, D., Becchetti, L., Boldi, P., Santini, M., & Vigna, S. (2006). A reference collection for web spam. *SIGIR Forum*, *40*.

Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA.

Chung, F. (1997). *Spectral graph theory*. No. 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI.

El-Yaniv, R., & Pechyony, D. (2006). Stable transductive learning. *Proc. 19th Annual Conference on Computational Learning Theory*.

El-Yaniv, R., & Pechyony, D. (2007). Transductive rademacher complexity and its applications. *Proc. 20th Annual Conference on Computational Learning Theory*.

Henzinger, M. (2003). Algorithmic challenges in web search engines. *Internet Mathematics*, *1*, 115–123.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proc. 20th IInternational Conference on Machine Learning*.

Joachims, T., Cristianini, N., & Shawe-Taylor, J. (2001). Composite kernels for hypertext categorisation. *Proc. 18th International Conference on Machine Learning*.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*, 604–632.

Meila, M., & Shi, J. (2001). A random walks view of spectral segmentation. *Proc. 8th International Workshop on Artificial Intelligence and Statistics*.

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA.

Rüping, S., & Scheffer, T. (2005). Learning with multiple views. *Proc. ICML Workshop on Learning with Multiple Views*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). A co-regularization approach to semi-supervised learning with multiple views. *Proc. ICML Workshop on Learning with Multiple Views*.

Smola, A., & Kondor, R. (2003). Kernels and regularization on graphs. *Proc. 16th Annual Conference on Learning Theory*.

Spielman, D., & Teng, S. (2003). Solving sparse, symmetric, diagonally-dominant linear systems in time $o(m^{1.31})$. *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science*.

Tsuda, K., Shin, H., & Schölkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, *21*, ii59–ii65.

Zhang, T., Popescul, A., & Dom, B. (2006). Linear prediction models with graph regularization for web-page categorization. *Proc. 12th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Zhou, D., Huang, J., & Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. *Proc. 22th International Conference on Machine Learning*.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proc. 20th International Conference on Machine Learning*.