CLASSIFICATION BY CHEEGER CONSTANT REGULARIZATION

Hsun-Hsien Chang and José M. F. Moura

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

This paper develops a classification algorithm in the framework of spectral graph theory where the underlying manifold of a high dimensional data set is described by a graph. The classification on the data is performed on the graph. The classifier optimizes an objective functional that combines prior information with the Cheeger constant. We interpret this approach as a regularized version of the Cheeger constant based classifier that we introduced recently. Our derivation shows that Cheeger regularization removes noise like a Laplacian based classifier but preserves better sharp boundaries needed for class separation. Experimental results show good performance of our proposed approach for classification applications.

Index Terms— classification, spectral graph theory, regularization, Cheeger constant, Laplacian

1. INTRODUCTION

Many practical applications need to classify a given data set into groups. For example, given an image, clustering its pixels into regions, or given a set of fingerprint images, grouping them into different individuals.

The observed data points are usually high dimensional, but the underlying manifold where the data points lie on is low dimensional. For example, in a two-class fingerprint database, each 512×512 image is a data point with 512×512 dimensions; the intrinsic manifold of the whole data set might be a real line where the origin bisects the data points into binary classes. The first step when performing classification is to describe the manifold in a faithful way. Since the unknown manifold might be *nonlinear*, Roweis and Saul [1] propose to approximate the manifold by a graph. The data points are treated as vertices in the graph, and the graph edges capture locally the similarities between pairs of vertices. The classification is performed on the graph.

Belkin and Niyogi [2] develop a semisupervised classifier that utilizes the Laplacian eigenmap of the graph. They express a classifier as a linear combination of low order eigenfunctions of the graph Laplacian. Then, a partially labeled data trains the classifier to obtain the optimal linear combination coefficients, hence the optimal classifier. The operator selects how many eigenfunctions needed to describe the classifier; this selection might lead to a nonoptimal classifier. We propose to design a classifier in a single step that optimizes a functional consisting of several terms.

Belkin, Niyogi, and Sindhwani [3] develop a classifier regularized by the graph Laplacian. The Laplacian is a smoothing operator. It reduces noise but blurs sharp boundaries separating the classes. After running the algorithm, operators obtain a smooth classification function and have to set a threshold for classification. The operator dependent threshold easily leads to inconsistent results. To overcome this issue, we propose to adopt the Cheeger constant of the graph as a regularization term in our objective functional.

The objective functional consists of two terms. The first term considers the prior information provided by human experts; it penalizes the deviation of the classification function from prior labels. The second term is a regularization formulated from the Cheeger constant of the graph. We call this approach the *Cheeger regularization* method. The Cheeger constant originates from the problem of graph partitioning [4], which is to find a small as possible subset of edges, called *edge cut*, whose removal will separate out a large as possible subset of vertices. Cheeger regularization not only removes noise but also evaluates automatically the best boundary, i.e., the edge cut, between classes. We expect the Cheeger constant to regularize better than the graph Laplacian.

Well-defined classifiers are integer-valued. An integer denotes the class that a data point belongs to. The integer-valued classifier needs integer programming, so it increases the levels of difficulty. Existing approaches [2, 5] circumvent the difficulty by relaxing the classification function to be real-valued. In contrast with these methods, we add constraints to our minimization problem to force the classifiers to behave like integer-valued functions.

The organization of this paper is as follows. Section 2 derives the formation of our objective functional and details the Cheeger regularization method. In Section 3, we show the application of the Cheeger regularization method to a fingerprint database obtained from the National Institute of Stan-

This work was supported by NIH grants, R01EB/AI-00318 and P41EB001977.

dards and Technology (NIST). Finally, Section 4 concludes this paper.

2. METHOD

Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ denote a set of data points. We represent the whole data set as a graph which approximates the underlying manifold of the data. A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges linking the vertices. Each data point $\mathbf{x}_i$ corresponds to a vertex v_i. We next assign edges connecting the vertices. In the graph representation, the vertices with similar features are linked together. We define the distance ρ_{ij} between pairwise $\mathbf{x}_i$, $\mathbf{x}_j$ as $\rho_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$. When the distance ρ_{ij} is below a predetermined threshold τ_ρ, the vertices v_i, v_j are connected by an edge; otherwise, they are disconnected.

In graph theory, we usually consider *weighted* graphs. Since not all connected pairs of vertices have the same distances, we capture this fact by using a weight function on the edges. We adopt a heat kernel, suggested by Belkin and Niyogi [2], to compute the weight W_{ij} on edge e_{ij} connecting vertices v_i and v_j:

$$W_{ij} = \begin{cases} \exp\left(-\frac{\rho_{ij}^2}{\sigma^2}\right), & \text{if there is edge } e_{ij} \\ 0\,, & \text{if no edge } e_{ij} \end{cases}, \tag{1}$$

where σ is the heat kernel parameter. The weight is large when the features of two linked vertices are similar. The weighted graph now is equivalently represented by its *weighted adjacency matrix* $\mathbf{W}$ whose elements W_{ij} are the weighted edges in equation (1). The matrix $\mathbf{W}$ has a zero diagonal because we do not allow the vertices to be self-connected. It is symmetric since $W_{ij} = W_{ji}$.

2.1. Prior Knowledge from Labeling

Assume that the first ℓ data points are labeled by an expert. For simplicity, this paper considers binary classification. Let $\{y_i\}_{i=1}^{\ell}$ be the labels, where $y_i = 1$ denotes one class and $y_i = -1$ denotes the other class. The goal is to find on the graph $\mathcal{G}$ a classification function $f : \mathcal{V} \to \{-1, +1\}$. Although we can choose any two real numbers to represent the classes, the merit of this choice will be clearer when we discuss the optimization in Section 2.3.

To estimate the classifier f, we penalize the average quadratic errors between the desired classifier f and the labels. Hence, the first term $J_1(f)$ of our objective is

$$J_1(f) = \frac{1}{\ell}\sum_{i=1}^{\ell}(f_i - y_i)^2\,, \tag{2}$$

where $f_i = f(v_i)$.

The domain $\mathcal{V}$ of the classification function f is discrete, so we can represent f by a vector

$$\mathbf{f} = [f_1, f_2, \cdots, f_N]^T\,. \tag{3}$$

The labels and the function on the first ℓ labeled data points are denoted by vectors

$$\mathbf{y}_{\mathrm{L}} = [y_1, y_2, \cdots, y_\ell]^T \tag{4}$$

and

$$\mathbf{f}_{\mathrm{L}} = [f_1, f_2, \cdots, f_\ell]^T\,, \tag{5}$$

respectively. In the sequel, the first objective term $J_1(f)$ becomes

$$J_1(\mathbf{f}) = \frac{1}{\ell}\|\mathbf{f}_{\mathrm{L}} - \mathbf{y}_{\mathrm{L}}\|^2\,, \tag{6}$$

which captures the prior provided by human experts.

2.2. Cheeger Regularization

The study of the Cheeger constant originates from the problem of graph partitioning [4]. The task of graph partitioning is to look for a subset E_0 of edges, i.e., an edge cut, whose removal will partition the vertex set $\mathcal{V}$ into two subsets V_1 and V_2. The vertex subsets are associated with a measure called *volume*. The Cheeger constant $C(V_1)$ is defined as the minimal cut-to-volume ratio

$$C(V_1) = \min_{V_1 \subset \mathcal{V}} \frac{|E_0(V_1, V_2)|}{\mathrm{vol}(V_1)}\,, \tag{7}$$

assuming that $\mathrm{vol}(V_1) \le \mathrm{vol}(V_2)$. In equation (7), $|E_0(V_1, V_2)|$ is the sum of the edge weights in the cut E_0:

$$|E_0(V_1, V_2)| = \sum_{v_i \in V_1, v_j \in V_2} W_{ij}\,. \tag{8}$$

The volume $\mathrm{vol}(V_1)$ of V_1 is defined as the sum of the vertex degrees in V_1:

$$\mathrm{vol}(V_1) = \sum_{v_i \in V_1} d_i\,, \tag{9}$$

where the degree d_i of the vertex v_i is defined as

$$d_i = \sum_{v_j \in V} W_{ij}\,. \tag{10}$$

Let χ denote the characteristic vector of V_1; its components χ_i are defined as

$$\chi_i = \begin{cases} 1, & \text{if } v_i \in V_1 \\ 0, & \text{if } v_i \notin V_1 \end{cases}. \tag{11}$$

We can express the Cheeger constant (7) in terms of the characteristic vector χ, see [5,6] for details,

$$C(\chi) = \min_{\chi} \frac{\chi^T \mathbf{L} \chi}{\chi^T \mathbf{D} \mathbf{1}}\,, \tag{12}$$

where $\mathbf{D} = \mathrm{diag}\,(d_1, d_2, \cdots, d_N)$ is a diagonal matrix of vertex degrees, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian, and $\mathbf{1}$ is a vector with all components 1.

Now, we can replace the characteristic vector χ with the classifier $\mathbf{f}$ by noting that

$$\chi = \frac{1}{2}(\mathbf{f} + \mathbf{1}) \,. \tag{13}$$

The Cheeger constant becomes

$$C(\mathbf{f}) = \min_{\mathbf{f}} \frac{\frac{1}{4}(\mathbf{f}+\mathbf{1})^T\mathbf{L}(\mathbf{f}+\mathbf{1})}{\frac{1}{2}(\mathbf{f}+\mathbf{1})^T\mathbf{D}\mathbf{1}} \,. \tag{14}$$

The evaluation of the Cheeger constant is equivalent to minimizing the numerator and the negative denominator. This leads to the regularization term $J_2(\mathbf{f})$ of our objective functional

$$J_2(\mathbf{f}) = \frac{1}{2}(\mathbf{f}+\mathbf{1})^T\mathbf{L}(\mathbf{f}+\mathbf{1}) - \beta(\mathbf{f}+\mathbf{1})^T\mathbf{D}\mathbf{1} \,, \tag{15}$$

where β is a non-negative parameter controlling the weight on the negative denominator. Using the fact that $\mathbf{L}\mathbf{1} = \mathbf{0}$, the regularization reduces to

$$J_2(\mathbf{f}) = \frac{1}{2}\mathbf{f}^T\mathbf{L}\mathbf{f} - \beta(\mathbf{f}+\mathbf{1})^T\mathbf{D}\mathbf{1} \,. \tag{16}$$

Equation (16) shows that Cheeger regularization is different from the Laplacian regularization suggested by Belkin, Niyogi and Sindhwani [3]. If we adopted the Laplacian regularization, the second objective term $J_2(\mathbf{f})$ would be $\mathbf{f}^T\mathbf{L}\mathbf{f}$. In equation (16), we can clearly see that the Cheeger regularization has one more term than the Laplacian regularization. The additional term plays the role of seeking for the optimal edge cut; equivalently, it contributes to sharp boundary preservation.

2.3. Constrained Optimization

Putting together the prior knowledge $J_1(\mathbf{f})$ and the Cheeger regularization $J_2(\mathbf{f})$, we have the objective

$$J(\mathbf{f}) = \gamma_1 J_1(\mathbf{f}) + \gamma_2 J_2(\mathbf{f}) \,, \tag{17}$$

where γ_1 and γ_2 are the non-negative weights.

There are two issues for the optimization. First, an important concern in $J_2(\mathbf{f})$ is to determine the value of β. Ideally, we would like to set β equal to the Cheeger constant $C(\mathbf{f})$, and let the minimization solve $J_2(\mathbf{f}) = 0$. However, the Cheeger constant varies with respect to the classifier $\mathbf{f}$, so we can not determine its value before running the minimization. To overcome this difficulty, we use the bounds of the Cheeger constant in the minimization. Spectral graph theory [4] upper and lower bounds the Cheeger constant as

$$\frac{1}{2}\lambda_1 \leq C(\mathbf{f}) < \sqrt{2\lambda_1} \,, \tag{18}$$

where λ_1 is the first nonzero eigenvalue of the graph Laplacian $\mathbf{L}$. Hence, we can constrain $\frac{1}{2}\lambda_1 \leq \beta < \sqrt{2\lambda_1}$ using the bounds (18) in the optimization formulation. We expect that β is equivalent to the Cheeger constant when the minimization reaches the optimal classifier.

Second, the desired classifier is a binary, integer-valued function. The minimization becomes an integer programming problem, which is a difficult task. Existing methods [2,5] suggest relaxing the classifier into a real-valued function, but the relaxation sacrifices the classifier's accuracy. In other words, the classifier can take any possible real value, and the operator has to determine and tune afterwards the threshold for different classes. We propose to handle this trade-off by adding constraints on the relaxed classifier. Recall from Section 2.1 that the classifier components f_i should be $+1$ or -1. We can enforce the components f_i to have absolute value one, namely $f_i^2 = 1$.

We now recast the optimization problem as the following:

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{f}) = \gamma_1 J_1(\mathbf{f}) + \gamma_2 J_2(\mathbf{f}), \quad \mathbf{f} \in \mathbb{R}^N \\ \text{subject to} \quad & \forall i,\ f_i^2 = 1 \\ & \beta < \sqrt{2\lambda_1} \\ & \beta \geq \frac{1}{2}\lambda_1 \,. \end{aligned} \tag{19}$$

In the optimization setting (19), $J_1(\mathbf{f})$ captures the prior information (6) and $J_2(\mathbf{f})$ is the Cheeger regularization (16). The formulation (19) generalizes [2, 3, 5]. When we remove all the constraints and set $\beta = 0$, the optimization reduces to the Laplacian regularization in [2, 3]. When $\gamma_1 = 0$ and $\gamma_2 = 1$, we need no prior knowledge and rely on the Cheeger constant to classify the data; the algorithm becomes fully automatic, similar to but more sophisticated than the isoperimetric approach in [5].

3. EXPERIMENTAL RESULTS

We implement our algorithm with MATLAB® on a computer with a 2.6 GHz CPU and 512 MB RAM. We adopt the optimization algorithm developed in [7] for minimization.

Application to a Fingerprint Database: We apply our classification algorithm to a NIST fingerprint database. The database contains 10 classes and each class has 200 images. Figure 1 displays one sample fingerprint drawn from each class. To test our algorithm, we run ten trials by randomly choosing two classes for each trial. The 400 images in each trial is treated as a given data set. For each class, we label 10 images before running the algorithm. Then, our classifier automatically determines the classes to which the images belong. In this application, the parameter σ^2 is 0.1 for computing the edge weights (1). The weighting parameters γ_1, γ_2 in the objective functional (19) are set to 1.

The criterion for performance evaluation is defined as the

Fig. 1. Ten sample images in the NIST fingerprint database.

success rate P_s for correctly recognized data:

$$P_\text{s} = \frac{\text{number of correctly classifed fingerprints}}{\text{number of all fingerprints}}. \tag{20}$$

Figure 2 shows the success rates for the ten experiments. The success rates are above 99%. This demonstrates that our developed classifier performs well.

Comparisons with other Classifiers: We compare our proposed method with two other schemes developed in the framework of spectral graph theory. The first method we compare is developed in [2, 3], which is a semi-supervised algorithm with the Laplacian regularization. The second method is the isoperimetric algorithm developed in [5], which is an automatic algorithm minimizing only the numerator of the Cheeger constant (12). The results of these two algorithms are displayed in Figure 2 as well. The plots show that our proposed Cheeger regularization method has performance better than or equal to the Laplacian regularization scheme. The isoperimetric algorithm always has the least success rates in all the experiments.

We use running time to evaluate algorithm complexity. In three methods, the graph representations of fingerprint database are identical. This part needs to compute all pairwise distances between all images and takes 951 seconds. Performing classification takes 2.00, 1.95, and 1.11 seconds for Cheeger regularization, Laplacian regularization, and isoperimetric algorithms, respectively. Although our proposed method has the longest running time, the successful classification rate trades off the speed.

4. CONCLUSIONS

This paper proposes a classification algorithm regularized by the Cheeger constant. Given a data set, we model its underlying manifold by a graph and define a binary classifier on the graph. The derivation of the classifier is formulated in the optimization framework. The objective functional we minimize consists of two terms. The first term considers the prior labeling by a human expert. The second term is a regularization term derived from the Cheeger constant. Cheeger regularization is a generalization of the Laplacian regularization. The Cheeger regularization not only includes the Laplacian regularization but also preserves the optimal boundary between classes.

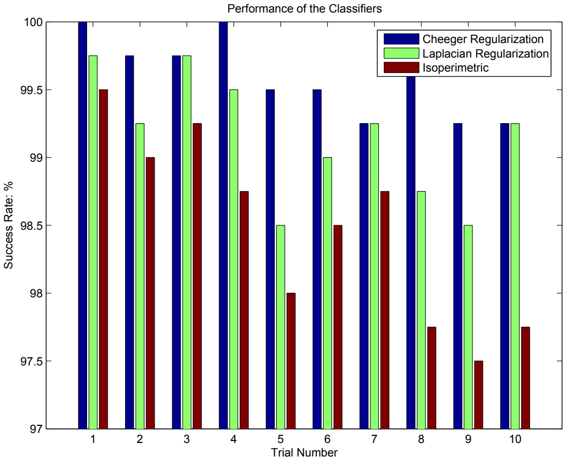

Fig. 2. Success rate of classifying fingerprint images using our proposed algorithm.

The integer-valued classifier introduces addition difficulties for optimizing the objective functional. We relax the classifier but add constraints to force it behave like an integer-valued function. The experimental results applied to a NIST fingerprint database show that our proposed classifier determines with high accuracy the different classes. The evaluation study demonstrates that the classifier regularized by the Cheeger constant outperforms other types of classifiers.

5. REFERENCES

[1] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, December 2000.

[2] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, no. 1-3, pp. 209–239, June 2004.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "On manifold regularization," in *Proceedings of International Workshop on Artificial Intelligence and Statistics*, Barbados, UK, January 2005, pp. 17–24.

[4] F. R. K. Chung, *Spectral Graph Theory*, vol. 92 of *CBMS Regional Conference Series in Mathematics*, American Mathematical Society, 1997.

[5] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 469–475, March 2006.

[6] H. H. Chang, J. M. F. Moura, Y. L. Wu, and C. Ho, "Automatic detection of regional heart rejection in USPIO-enhanced MRI," submitted to *IEEE Transactions on Medical Imaging*.

[7] H. H. Chang, J. M. F. Moura, Y. L. Wu, and C. Ho, "Immune cells detection of *in vivo* rejecting hearts in USPIO-enhanced magnetic resonance imaging," in *Proceedings of IEEE International Conference of Engineering in Medicine and Biology Society*, New York, NY, August 2006, pp. 1153–1156.