Cheeger Cut Model for the Balanced Data Classification Problem

Yan-zhou Zhang [1, a], Yan Jiang [2,b], Zhi-Feng Pang [3,c,*]

[1] Basic Courses Department, Henan Polytechnic, Zhengzhou, 450046, China

[2] Foundation department, Zhengzhou Tourism College, Zhengzhou, 45009, China

[3] College of Mathematics and Information Science, Henan University, Kaifeng, 475004, China

[a] zhyanzh2003@yahoo.com.cn , [b] yanjiang0628@163.com ,

[c] zhifengpang@163.om (* Corresponding Author)

Keywords: Cheeger Cut, Penalty Method, Weighted Function, Scaling Parameter

Abstract: In this paper we propose a numerical method based on the splitting strategy to solve the Cheeger cut model. In order to improve the classification results, we propose a new self-tuning strategy to choose a robust scaling parameter. Some numerical examples are arranged to illustrate the efficiency of our proposed method.

Introduction

Data classification is of an important topic in machine learning and computer vision. Some original clustering techniques aiming to data classification were based on the combinatorial normalize/ratio graph cut problem [10]. These techniques tend to find the second eigenvector or the first k th eigenvectors of the unnormalized and normalized graph Laplacians based on some suitable relaxations. Recent ideas were extended to the standard graph Laplacian such as graph p -Laplacian [4,9], where they showed that using a ratio of p -homogeneous functions leads quite naturally to a nonlinear eigenvalue problem associated to a certain nonlinear operator. Furthermore, Hein and Buhler [7] noticed that the graph p -Laplacian approximates to the Cheeger cut when $p \to 1$. Unlike other graph-based approximation/relaxation techniques, this approximation can be obtained any arbitrarily exact. This observation theoretically and practically thus starts a direction for spectral clustering techniques based applications.

Following the work in the Cheeger cut problem [7], Bresson et al. [12,3] recently proposed to first introduce some constrained variables and then to use an operator splitting method. Since these subproblems could be efficiently solved by using the classical optimization methods such as the splitting method and the augmented Lagrangian method, they gave some efficient numerical results. In this paper we also consider the relaxation strategy following the work in [12, 3]. Differentially, we use the penalty method to solve it by introducing a variable substitution. Furthermore, we propose a new and more robust scaling parameter to improve the effectiveness of classification. Some numerical compares are arranged to illustrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. In section 2 we proposed a numerical method to solve the Cheeger cut model. We give some numerical comparisons to illustrate the efficiency of our proposed method

Cheeger Cut Model

Set $V = \{v_1, v_2, \cdots, v_N\}$ be N vertices and $w: V \times V \to R^+$ represents the similarity between its vertices, the Cheeger cut value of a binary partition is defined by

$$C(S) = \frac{cut(S, S^c)}{\min(|S|, |S^c|)} \tag{2.1}$$

Published by Atlantis Press, Paris, France.
© the authors, 2013

with $S \cup S^c = V$ and $S \cap S^c = \phi$, where $cut(S, S^c) := \sum_{i \in S, j \in S^C} \omega_{i,j}$ and $|\cdot|$ denotes the number of the points in a given set. The problem (2.1) is an NP-hard problem, however we can find an exact continuous relaxation

$$\min_{x \in [0,1]^N} \frac{\|Dx\|_{\ell^1}}{\|X - m(x)\|_{\ell^1}} \tag{2.2}$$

to find its solution, where D is a bounded operator and $m(x)$ denotes the median value of the vector x, $[0,1]^N = [0,1] \times [0,1] \times \cdots \times [0,1]$ for $N = |V|$. If the global optimal solution x^* of the problem (2.2) is obtained, then we can use the natural relationship of $x^* = 1_s$ to get the exact partition S and S^c. For simplification, we set the minimization value of (2.2) as $\bar{\lambda}$ and the solution $\bar{x}$.

Dinkelbach type method.

The problem (2.2) is actual the fractional programming problem. However, as the usual drawbacks of the fractional programming problem, there is no direct algorithm for guaranteeing to get the global solution of (2.2), which urges us to consider the following parametric problem

$$F(\lambda) := \min_{X \in [0,1]^N} \|Dx\|_{\ell^1} - \lambda \|x - m(x)\|_{\ell^1} \tag{2.3}$$

where $\lambda \in R$. This is based on the equivalence of getting the minimized value of (2.2) by finding the root of the equation $F(\lambda) = 0$ [6].

Algorithm 2.1. Compute the minimization problem (2.3).

Step 0 Choose an original value x^0 and set $\lambda^1 = \frac{\|Dx^0\|_{\ell^1}}{\|x^0 - m(x^0)\|_{\ell^1}}$. Let $k = 1$;

Step 1 Compute x^k by

$$F(\lambda^k) := \min_{x \in [0,1]^N} \|Dx\|_{\ell^1} - \lambda^k \|x - m(x)\|_{\ell^1} \tag{2.4}$$

Step 2 If $F(\lambda^k) = 0$, then stop. Otherwise, set

$$\lambda^{k+1} = \frac{\|Dx^k\|_{\ell^1}}{\|x^k - m(x^k)\|_{\ell^1}} \tag{2.5}$$

and let $k := k + 1$ go to Step 1.

The problem (2.4) in Algorithm 2.1 includes a nonlinear operator $m(x)$. Since the original vector $x \in \{0,1\}^N$, we can assume that $m(x)$. Then the minimization problem (2.4) can be rewritten as

$$\min_{x \in C_1} \|Dx\|_{\ell^1} - \lambda^k \|x\|_{\ell^1}, \tag{2.6}$$

where $C_1 := \{x | x \in [0,1]^N\} \cap \{x | x^T 1 \le N/2 + 1\}$. Obviously the set C_1 is a bounded close convex set, then the problem (2.6) has at least one global solution denoted by $\tilde{x}$. However, the problem (2.6) includes two non-smoothing terms, which is not to be solved. With the help of an auxiliary variable y, we can transfer to consider

$$\begin{cases} \min\limits_{x,y \in C} \|Dx\|_{\ell^1} - \lambda^k \|y\|_{\ell^1} \\ s.t. \quad x = y \end{cases} \tag{2.7}$$

where $C := \{y | y \in [0,1]^N\} \cap \{y | y^T 1 \le N/2 + 1\}$. Using the penalty method, the minimization problem (2.7) can be written as

$$\min_{x,y \in C} \underbrace{\|Dx\|_{\ell^1} + \frac{1}{2\tau}\|x - y\|^2_{\ell^2} - \lambda^k \|y\|_{\ell^1}}_{=F(x,y)}. \tag{2.8}$$

So we can solve the problem (2.8) by the splitting strategy as follows

© the authors, 2013

$$\begin{cases} R\left(y^{m-1}\right) := x^m = \arg\min\limits_{x} \ \|Dx\|_{\ell^1} + \dfrac{1}{2\tau}\left\|x - y^{m-1}\right\|^2_{\ell^2} & (2.9a) \\ S\left(x^m\right) := y^m = \arg\min\limits_{Y\in C} \ \dfrac{1}{2\tau}\left\|x^m - y\right\|^2_{\ell^2} - \lambda^k \|y\|_{\ell^1} & (2.9b) \end{cases}$$

Obviously, the subproblem (2.9a) corresponds to the special $\ell^1 - \ell^2$ model, which solution can be obtained by using the classic projection gradient method such as the Bermudez-Moreno algorithm [1, 2]. That is to say, let ξ^0 be a suitable original value, we can consider the iterative scheme as

- Choose the original value ξ^{m_0} and set $j = 1$;
- Compute ξ^{m_j} by $\xi^{m_j} = P_\beta(\xi^{m_{j-1}} + \frac{1}{\delta} D(D^T \xi^{m_{j-1}} - y^{m-1}))$,

where P is a projection operator and $\beta_\delta = \{\xi : |\xi| \le 1\}$;

- If the stop condition is satisfied, set $x^m = y^{m-1} - \tau D^T \xi^{m_j}$; Otherwise, set $j = j+1$ and go to the second step.

For the linear constraint subproblem (2.9b), we first focus on the constrained condition $0 \le y^T 1 \le \frac{N}{2} + 1$ for the cost function to get the optimal solution y^0. Specifically, we first solve

$$\begin{cases} \min\limits_{Y} \ \dfrac{1}{2\tau}\left\|y - x^m\right\|^2_{\ell^2} - \lambda_k \|y\|_{\ell^1} \\ s.t. \ \ y^T 1 \le 1 + \dfrac{N}{2} \quad . \end{cases} \tag{2.10}$$

Based on the numerical optimization method, the solution of the problem (2.10) satisfies that

$$y - x^m - \tau\lambda^k W + \tau p = 0, \tag{2.11}$$

where $p = \max\left(0, p + c_1\left(y^T 1 - \frac{N}{2} - 1\right)\right)$ and $W = (w_1, w_2, \cdots, w_N)^T$ such that

$$w_i : \begin{cases} = \dfrac{y_i}{|y_i|} sign(y_i), & if \ y_i \ne 0 \\ \in [0,1], & if \ y_i = 0 \end{cases} \tag{2.12}$$

for $i = 1,2,\cdots,N$. Furthermore, we get the explicit solution $y^{o,m} = \left(y_1^{o,m}, y_2^{o,m}, \cdots, y_N^{o,m}\right)^T$ of the minimization problem (2.10) based on the relationships of (2.11) and (2.12) as

$$y_i^{o,m} = \begin{cases} r_i^m + \lambda^k \tau sign\left(r_i^m\right), & if \ r_i^m \ne 0 \\ \pm\lambda^k \tau, & if \ r_i^m = 0 \end{cases} \tag{2.13}$$

for $i = 1,2,\cdots,N$, where $r^m{}_i = x^m{}_i - p^m \tau$ and $p^m = \max\left(0, p^{m-1} + c_1\left(\left(y^{m-1}\right)^T 1 - \frac{N}{2} - 1\right)\right)$. Once we get $y^{o,m}$, then the solution $y^m = \left(y_1^m, y_2^m, \cdots, y_N^m\right)^T$ of the minimization problem (2.9b) can be obtained by truncating every subvariables $y^{o,m}$ into $[0,1]$ with the following strategy

$$y^m = \Pi_{[0,1]^N}\left(y^{o,m}\right) =: \begin{cases} 1, & if \ y_i^{o,m} > 1 \\ y_i^{o,m}, & if y_i^{o,m} \in [0,1] \\ 0, & if \ y_i^{o,m} < 1 \end{cases} \tag{2.14}$$

for $i = 1,2,\cdots,N$. Then we have the following strategy to solve the problem (2.11) as

$$\begin{cases} \bullet \ Choose \ original \ value \ p^{m_0} and \ set \ r = 1; \\ \bullet \ Compute \ y^m by \ the \ strategy \ (2.14); \\ \bullet \ Update \ p^m := p^m 1 \ by \ p^m = \max\left\{0, p^{m-1} + c_1\left(\left(y^m\right)^T 1 - \dfrac{N}{2} - 1\right)\right\} \\ \bullet \ If \ the \ stop \ conditon \ is \ satisfied, \\ set \ x^m = y^{m-1} - \dfrac{\mu}{\delta} D^* \xi^{m_j}; \quad Otherwise, \ set \ j = j+1 \ and \ continuously \ compute \ \xi^{m_j}. \end{cases} \tag{2.15}$$

Based on the strategies (2.10) and (2.14) and choosing some suitable values, so we can get the solution of the subproblem (2.4).

Published by Atlantis Press, Paris, France.
© the authors, 2013

Numerical Implementation

In this section we consider the numerical implementation based on the Cheeger-cut model. In the Cheeger-cut model, the quality of the linear operator D is confirmed by the weighted function. A popular weighted function is the Gaussian kernel which defines the similarity between two points as $w_{i,j} = \exp(-d_{i,j}^2 / \sigma^2)$, where $d_{i,j}$ is the distance between datum points x_i and x_j, the scaling parameter σ determines the similarity of these two datum points. However, the fixed scaling parameter σ is not suitable for more complex data. To circumvent this problem, Zelnik-Manor and Perona [13] proposed to consider an adapted parameter $\sigma^2{}_1 = \sigma_i \sigma_j$ with the assumption of σ_i / σ_j as the distance to the kth nearest neighbor for the datum point x_i / x_j. Unfortunately, this method maybe suffer from a drawback when one scaling parameter is significantly larger than the other, multiplying the scaling parameters together could cause the points to become more similar than desired. A possible choice is to set $\sigma_2 = \min(\sigma_i, \sigma_j)$ in [11] or $\sigma_1 = \max(\sigma_i, \sigma_j)$ in [8], but these choices obviously separate the relationship between σ_i and σ_j. So here we propose a new scaling parameter $\sigma^2{}_4 = \frac{1}{3}(\sigma^2{}_i + \sigma_i \sigma_j + \sigma^2{}_j)$. Obviously, when the gap between σ_i and σ_j is smaller, the property of the proposed new parameter tends to the original parameter σ^2 or $\sigma_i \sigma_j$. Inversely, the scaling property tends to the dominated parameter.

Examples. In numerical examples, we consider three data show in Figure3.1 for the data classification, where data are first added to the Gaussian noise with $\sigma = 0.025$ and then embedded in R^{100}. For parameters, we first fix the parameter $\sigma = 0.4$ and then tune other parameters by repeating the numerical algorithms in order to choose more suitable parameters. The numerical results including the error points (also the percentage) between the original data and the experimental results are shown Table 3.1. As we can see from Table Ⅰ, our numerical method can efficiently cluster the data, especially for choosing scales σ_1 and σ_4. It's worth noting that we can get worst classification results when the structure of datum such as Data (III) is complex.

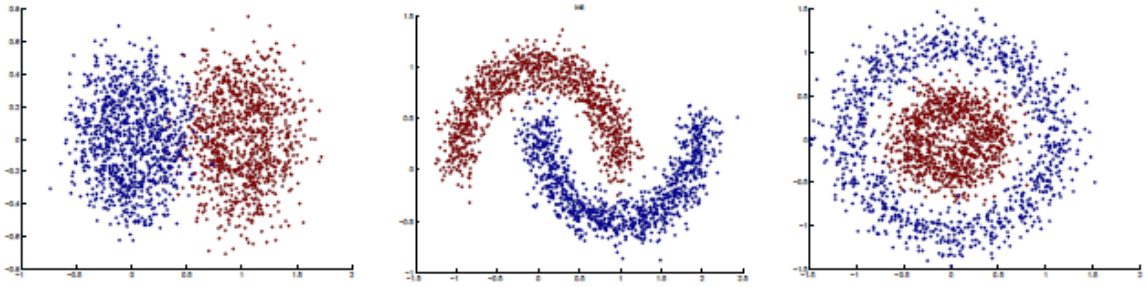

Figure 3.1: Left to right: (a): Data(I);(b): Data(II);(c)Data(III)

Table 3.1 Numerical results in Examples

Scale\ Erro	Data Set (I)		Data Set (II)		Data Set (III)	
	Error (%)	Poins	Error (%)	Points	Error (%)	Points
σ_1	3.25	65	9.45	189	31.40	628
σ_2	3.30	66	9.50	190	34.15	683
σ_3	3.30	66	9.55	191	34.1	682
σ_4	3.25	65	9.10	182	33.35	647

Published by Atlantis Press, Paris, France.
© the authors, 2013

Conclusion

In this paper we proposed a splitting method to solve a balanced data classification problem. In order to improve the numerical results, we also propose a new self-tuning strategy for choosing suitable scale. Some numerical examples were arranged to illustrate the efficiency of our proposed method. However, we also noticed that our method did not efficiently deal with the more complicated data. In the future we will further consider some new numerical models or methods to improve the classification results.

Acknowledgment

The third author acknowledges the financial support by the university research fund of Henan University (2011YBZR003). The author is grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] J. Aujol. Some first-order algorithms for total variation based image restoration. Journal of Mathematical Imaging and Vision, 34(3):307-327, 2009.

[2] A. Bermudez and C. Moreno. Duality methods for solving variational inequalities. Computers \& Mathematics with Applications, 7(1):43-58, 1981.

[3] X. Bresson, X. Tai, T. Chan, and A. Szlam. Multi-class transductive learning based on L1 relaxations of Cheeger cut and Mumford-Shah-Potts model. UCLA CAM Report 12-03.

[4] T. Buehler and M. Hein. Spectral clustering based on the graph p-Laplacian. In Proceedings of the 26th International Conference on Machine Learning, 81-88, 2009.

[5] F. Chung. Spectral Graph Theory. America Mathematics Society, 1997.

[6] W. Dinkelbach. On nonlinear fractional programming. Management Science, 13(7):492-498, 1967.

[7] M. Hein and T. Buehler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In Advances in Neural Information Processing Systems, 847-855, 2010.

[8] M. Hein and S. Setzer. Beyond spectral clustering - tight relaxations of balanced graph cuts.In Advances in Neural Information Processing Systems, 24:2366--2374, 2011

[9] D. Luo, H. Huang, C. Ding, and F. Nie. On the eigenvectors of p-Laplacian. Machine Learning, 81(1):37-51, 2010.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transactions Pattern Analysis and Machine Intelligence, 22(8):888-905, 2000.

[11] K. Streib and J. Davis. Using Ripley’s K-function to improve graph-based clustering techniques.IEEE Conference on Computer Vision and Pattern Recognition, 2305-2312, 2011.

[12] A. Szlam and X. Bresson. Total variation and Cheeger cuts. In Proceedings of the 27th International Conference on Machine Learning, 1039-1046, 2010.

[13] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In Advances in Neural Information Processing Systems, 2:1601-1608, 2004.

Published by Atlantis Press, Paris, France.
© the authors, 2013