A Self-Balanced Min-Cut Algorithm for Image Clustering

Xiaojun Chen, Joshua Zhexue Haung
College of Computer Science and Software, Shenzhen University
Shenzhen, 518060, P.R. China
xjchen@szu.edu.cn, zx.huang@szu.edu.cn

Feiping Nie (Corresponding author)
School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University
Xi′an 710072, Shanxi, P. R. China.
feipingnie@gmail.com

Renjie Chen, Qingyao Wu
School of Software Engineering, South China University of Technology
Guangzhou, P. R. China.
chen.renjie@mail.scut.edu.cn, qyw@scut.edu.cn

Abstract

Many spectral clustering algorithms have been proposed and successfully applied to image data analysis such as content based image retrieval, image annotation, and image indexing. Conventional spectral clustering algorithms usually involve a two-stage process: eigendecomposition of similarity matrix and clustering assignments from eigenvectors by k-means or spectral rotation. However, the final clustering assignments obtained by the two-stage process may deviate from the assignments by directly optimize the original objective function. Moreover, most of these methods usually have very high computational complexities. In this paper, we propose a new min-cut algorithm for image clustering, which scales linearly to the data size. In the new method, a self-balanced min-cut model is proposed in which the Exclusive Lasso is implicitly introduced as a balance regularizer in order to produce balanced partition. We propose an iterative algorithm to solve the new model, which has a time complexity of $O(n)$ where n is the number of samples. Theoretical analysis reveals that the new method can simultaneously minimize the graph cut and balance the partition across all clusters. A series of experiments were conducted on both synthetic and benchmark data sets and the experimental results show the superior performance of the new method.

1. Introduction

Over the past decades, many clustering algorithms have been proposed for cluster analysis of high-dimensional data, such as spectral clustering [21], subspace clustering [13, 8], multi-view clustering [4, 7], etc. Among them, spectral clustering is a popular method because it often shows good clustering performance due to the use of manifold information. Various spectral clustering algorithms have been proposed, such as Ratio Cut [12], k-way Ratio Cut [5], Normalized Cut [15], Spectral Embedded Clustering [19] and Constrained Laplacian Rank [18]. They have been successfully applied to image clustering in applications such as content based image retrieval, image annotation, and image indexing [10, 23, 3].

Spectral clustering methods usually transform the data into a weighted, undirected graph based on pairwise similarities. To obtain the final discrete clustering assignments, they often perform eigendecomposition on the similarity matrix first, and then carry out the final clustering assignments from eigenvectors by k-means or spectral rotation [24]. However, the final clustering assignments obtained by a two-stage process may deviate from the assignments by directly optimize the original objective function.

Moreover, since both graph construction and spectral analysis are time consuming, spectral clustering usually has a time complexity of $O(n^3)$ where n is the number of samples. In recent years, much effort has been devoted to accelerating the spectral clustering process. There are mainly three ways to handle the scalability issue of spectral clus-

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.227

tering. One way is to reduce the computational cost of the eigendecomposition step [11, 14], the second way is to sample the original data and perform clustering on the reduced data [22, 20], and the last way is to construct a small approximate affinity matrix and perform clustering on the small affinity matrix [2]. However, these methods are mainly based on sampling, and a lot of information of the data will be lost in the sampling step.

In this paper, we propose a novel Self-Balanced Min-Cut (SBMC) algorithm for image clustering. The main contributions of our work include:

1. We have proved that the Exclusive Lasso proposed in [25] can be used as a balance regularizer to produce balanced partition and avoid recovering a lot of small clusters.

2. We have proposed a new graph cut model, named the self-balanced min-cut model, in which the Exclusive Lasso is implicitly introduced as a balance regularizer. The regularization parameter, named the balance parameter, can be learnt.

3. We have proposed an iterative algorithm SBMC to solve the new model, in which the balance parameter is updated in each iteration. SBMC has a time complexity of $O(n)$ where n is the number of samples. Theoretical analysis reveals that the new method can simultaneously minimize the graph cut and balance the partition across all clusters. We also show that the conventional min-cut model can be considered as a special case of the new model.

4. Comprehensive experiments on both synthetic and benchmark data sets show the efficiency and effectiveness of the proposed method.

The rest of this paper is organized as follows. Notations and preliminaries are given in Section 2. We review the related work in Section 3 and the background in Section 4. The Self-Balanced Min-Cut algorithm is given in Section 5. We present experimental results and analysis in Section 6. Conclusions and future work are given in Section 7.

2. Notations and Definitions

We summarize the notations and the definition of norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i-th row is denoted as $\mathbf{m}^i$, and its j-th column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $\mathbf{M} \in \mathcal{R}^{n\times m}$ is defined as $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} m_{ij}^2}$.

3. Related Work

Given a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we can construct an affinity matrix $\mathbf{A}$. $\mathbf{A}$ can be considered as a weighted undirected graph. Let $\mathbf{Y} \in \mathcal{B}^{n\times c}$ be the cluster indicator matrix, in which c is the number of clusters and $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. The classical Ratio Cut can be written as [12]

$$\min_{\mathbf{Y}^T\mathbf{Y}=\mathbf{I}} Tr(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) \tag{1}$$

and the Normalized Cut can be represented by [15]

$$\min_{\mathbf{Y}^T\mathbf{D}\mathbf{Y}=\mathbf{I}} Tr(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) \tag{2}$$

where $\mathbf{L}_A = \mathbf{D}_A - \mathbf{A}$ is Laplacian matrix, and $\mathbf{D}_A$ is the corresponding degree matrix which is a diagonal matrix with the i-th diagonal element as $d_{ii} = \sum_{j=1}^{n} a_{ij}$.

Problems (1) and (2) can be solved by a two-stage process: performing eigendecomposition on $\mathbf{L}_A$ first, and obtaining the final clustering assignments from eigenvectors by k-means or spectral rotation [24].

In 2011, Nie et al. proposed a spectral embedded clustering (SEC), which introduces a linearity regularization into the objective function to control the mismatch between the cluster assignment matrix and the low-dimensional embedding of the data [19]. To cluster $\mathbf{X}$ into c clusters, the objective function of SEC is as follows

$$\min_{\substack{\mathbf{Y},\mathbf{W},\mathbf{b}\\ \mathbf{Y}^T\mathbf{Y}=\mathbf{I}}} Tr(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) + \mu\left(\left\|\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{Y}\right\|^2 + \gamma_g Tr(\mathbf{W}^T\mathbf{W})\right) \tag{3}$$

where $\mathbf{W} \in \mathcal{R}^{n\times c}$ is the projection matrix, $\mathbf{b} \in \mathcal{R}^{n\times 1}$ is the bias vector, μ and γ_g are two regularization parameters. The above problem can be solved in a similar way as problems (1) and (2).

In 2014, Nie et al. proposed a clustering method CAN (Clustering with Adaptive Neighbors)[17]. CAN learns a probability matrix $\mathbf{S} \in \mathcal{R}^{n\times n}$, in which s_{ij} is the connected probability between $\mathbf{x}_i$ and $\mathbf{x}_j$. The objective function of CAN is as follows

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n}\left(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 s_{ij} + \gamma s_{ji}^2\right) \\ &\forall\, i,\ \mathbf{s}^i\mathbf{1} = 1,\ s_{ij} \in [0,1], rank(\mathbf{L}_S) = n - c \end{aligned} \tag{4}$$

where a rank constraint $rank(L_S) = n - c$ is imposed to the Laplacian matrix of $\mathbf{S}$ such that the connected components in $\mathbf{S}$ are exactly equal to c. The above problem can be solved with an iterative method. In each iteration, we have to perform eigendecomposion on $\mathbf{L}_S$.

In 2016, Nie et al. further improved CAN for a given affinity matrix to propose the Constrained Laplacian Rank

(CLR) method [18]. CLR learns $\mathbf{S} \in \mathcal{R}^{n\times n}$ that best approximates the initial affinity matrix $\mathbf{A}$. Two versions of CLR were proposed, one is with the ℓ_2 norm

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\|\mathbf{S}-\mathbf{A}\|_2^2 \right) \\ &\forall\, i,\ \mathbf{s}^i\mathbf{1}=1,\ s_{ij}\in[0,1], rank(\mathbf{L}_S)=n-c \end{aligned} \tag{5}$$

and the other one is with the ℓ_1 norm

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\|\mathbf{S}-\mathbf{A}\|_1 \right) \\ &\forall\, i,\ \mathbf{s}^i\mathbf{1}=1,\ s_{ij}\in[0,1], rank(\mathbf{L}_S)=n-c \end{aligned} \tag{6}$$

The above two problems can be solved with iterative methods, in which eigendecomposion is performed on $\mathbf{L}_S$ in each iteration.

To accelerate the spectral clustering process, Cai et al. proposed a landmarks-based spectral clustering (LSC) method [2]. Given a data set with n samples, LSC generates m ($m \ll n$) representative data points to compute a representative similarity matrix and the eigendecomposition can be performed on the low-size representative similarity matrix. The final discrete clustering result is obtained by performing k-means clustering on the eigenvectors. The overall time of LSC is $O(ndmt+nm^2)$ where t is the number of iterations of k-means for anchor generation, which is significant reduction from $O(n^3)$ considering $m \ll n$.

4. Background

In this section, we introduce the Exclusive Lasso and Augmented Lagrangian multiplier optimization method which will be used in the next section. We also rewrite the min-cut problem for the following analysis.

4.1. Exclusive Lasso

Given a matrix $\mathbf{M} \in \mathcal{R}^{n\times m}$, Zhou et al. proposed the exclusive lasso for multi-task feature selection, which is defined as [25]

$$\|\mathbf{M}\|_e = \sum_{j=1}^{m} (\sum_{i=1}^{n} |m_{ij}|)^2 \tag{7}$$

The exclusive lasso was originally used for feature selection across multiple tasks. It models the scenario when variables in the same group compete with each other. With exclusive lasso, if one feature in a group is given a large weight, it tends to assign small or even zero weights to other features in the same group. From another point of view, the exclusive lasso can be considered as a combination of a ℓ_1-norm on the elements in the same row and a ℓ_2-norm on the ℓ_1-norm of each row. Since ℓ_1-norm tends to achieve a sparse solution, the construction in the exclusive lasso essentially introduces a competition among different columns for the same rows. In the following, we prove that the most balanced clustering can be obtained by minimizing the exclusive lasso.

Theorem 1. *Suppose $\mathbf{Y} \in \mathcal{B}^{n\times c}$ is a cluster indicator matrix, $\|\mathbf{Y}\|_e$ arrives its minimum when $\sum_{i=1}^{n} y_{ij}$ equals to $\frac{n}{c}$ if $\frac{n}{c}$ is an integer, or $\{\lfloor \frac{n}{c} \rfloor, \lceil \frac{n}{c} \rceil\}$ otherwise ($j \in [1,c]$).*

Proof. Let $\mathbf{u} \in \mathcal{R}^{c\times 1}$ be a column vector where $u_j = \sum_{i=1}^{n} y_{ij}$ and $\sum_{j=1}^{c} u_j = n$. Let $\mathbf{v} \in \mathcal{R}^{\times 1}$ be a constant column vector where $v_j = \frac{1}{c}$. According to the Cauchy-Schwarz inequality, we have $|<\mathbf{u},\mathbf{v}>|^2 \le \|\mathbf{u}\|^2 \|\mathbf{v}\|^2$ which indicates that

$$\sum_{j=1}^{c} u_j^2 \ge \frac{n^2}{c} \tag{8}$$

and the inequality holds when $u_j = \frac{n}{c}$ for $\forall j \in [1,c]$. Therefore, $\|\mathbf{Y}\|_e$ arrives its minimum when $\sum_{i=1}^{n} y_{ij} = \frac{n}{c}$ ($j \in [1,c]$). If $\frac{n}{c}$ is not an integer, we can verify that $\|\mathbf{Y}\|_e$ arrives its minimum when $\sum_{i=1}^{n} y_{ij} = \{\lfloor \frac{n}{c} \rfloor, \lceil \frac{n}{c} \rceil\}$ ($j \in [1,c]$). □

According to the above theorem, we can use the exclusive lasso as a balance constraint to obtain balanced clustering result.

4.2. Augmented Lagrangian multiplier (ALM)

Consider the constrained optimization problem

$$\min_{g(\mathbf{M})=0} f(\mathbf{M}) \tag{9}$$

where $\mathbf{M} \in \mathbf{R}^{n\times m}$. The algorithm using the augmented Lagrangian multiplier (ALM) method to solve problem (9) is described in Algorithm 1 [1].

Algorithm 1 Algorithm to solve problem (9)

1: **Input:** $\mathbf{X}$, $\rho \in (1,2)$.
2: Initialize $\mu > 0$, $\mathbf{\Lambda}$.
3: **repeat**
4: Update $\mathbf{M}$ by solving $\min_{\mathbf{M}} \left(f(\mathbf{M}) + \frac{\mu}{2} \left\| g(\mathbf{M}) + \frac{\Lambda}{\mu} \right\|_F^2 \right)$
5: Update Λ by $\Lambda = \Lambda + \mu g(\mathbf{M})$
6: Update μ by $\mu = \rho\mu$
7: **until** problem (9) converges
8: **Output:** the optimal solution of $\mathbf{M}$.

It has been proved that under some rather general conditions, Algorithm 1 converges Q-linearly to the optimal solution [1]. This property makes the ALM method very attractive.

4.3. Min-Cut revisited

In this section, we rewrite the min-cut problem which partitions the vertices in the affinity matrix $\mathbf{A}$ into c disjoint sets so that the total weight of the set of edges with endpoints in different sets is minimized. Let $\mathbf{Y} \in \mathcal{B}^{n\times c}$ be the cluster indicator matrix, in which $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. The objective function of min-cut clustering is formulated as follows

$$\min_{\mathbf{Y}\in Ind} \mathbf{1}^T A\mathbf{1} - Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) \tag{10}$$

Note that $\mathbf{1}^T\mathbf{A}\mathbf{1} = \mathbf{1}^T\mathbf{D}_A\mathbf{1} = Tr(\mathbf{Y}^T\mathbf{D}_A\mathbf{Y})$, the above problem can be rewritten as

$$\min_{\mathbf{Y}\in Ind} Tr(\mathbf{Y}^T\mathbf{L}_A\mathbf{Y}) \tag{11}$$

Problem (11) is difficult to solve. A well known method is to relax $\mathbf{Y}$ from the discrete values to the continuous ones, and add different constraints to form the Ratio Cut problem in Eq. (1) and the Normalized Cut problem in Eq. (2).

In this paper, we want to directly obtain discrete solution $\mathbf{Y}$. Obviously, problem (10) can be rewritten as

$$\max_{\mathbf{Y}\in Ind} Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) \tag{12}$$

However, directly solving the above problem results in a degenerated solution in which all objects belong to one cluster. In the next section, we will propose a new min-cut algorithm.

5. Self-Balanced Min-Cut Algorithm

Given a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we can construct an affinity matrix $\mathbf{A}$. Suppose we want to cluster $\mathbf{X}$ into c clusters. Let $\mathbf{Y} \in \mathcal{B}^{n\times c}$ be the cluster indicator matrix, in which $y_{il} = 1$ indicates that $\mathbf{x}_i$ is assigned to the l-th cluster. Denoting $\mathbf{B} = \mathbf{Y}\mathbf{Y}^T$, we know that $b_{ij} = 1$ if $\mathbf{y}^i = \mathbf{y}^j$, and $b_{ij} = 0$ otherwise. To obtain a good partition $\mathbf{Y}$, we hope that $b_{ij} = 0$ if a_{ij} is small and $b_{ij} = 1$ if a_{ij} is big. Intuitively, we can obtain the clustering assignments by minimizing the difference between $\mathbf{A}$ and $\mathbf{Y}\mathbf{Y}^T$. A natural way is to obtain the clustering assignments by solving the following problem

$$\min_{\mathbf{Y}\in Ind} \|\mathbf{A} - \mathbf{Y}\mathbf{Y}^T\|_F^2 \tag{13}$$

However, the above problem may only work for data with perfect cluster structure which rarely exists in real life data. We will discuss the disadvantage of problem (13) later. In this paper, we propose to solve the following objective function

$$\min_{\mathbf{Y}\in Ind, s} \|\mathbf{A} - s\mathbf{Y}\mathbf{Y}^T\|_F^2 \tag{14}$$

where $s > 0$ is a balance parameter.

Problem (14) seems difficult to solve. Fortunately, we can rewrite problem (14) as a new problem which is much easier to solve according to the following theorem

Theorem 2. *Solving problem (14) is equivalent to solving the following problem*

$$\max_{\mathbf{Y}\in Ind, s>0} 2sTr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2\|\mathbf{Y}\|_e \tag{15}$$

where $\|\mathbf{Y}\|_e = Tr(\mathbf{Y}^T\mathbf{1}\mathbf{1}^T\mathbf{Y})$ *is the exclusive lasso [25].*

Proof. Since $\mathbf{Y} \in Ind$, it can be verified that $Tr(\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T) = \sum_{j=1}^{c}(\sum_{i=1}^{n} y_{ij})^2$. Then we know $Tr(\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T) = Tr(\mathbf{Y}^T\mathbf{1}\mathbf{1}^T\mathbf{Y}) = \|\mathbf{Y}\|_e$.

Problem (14) can be rewritten as follows

$$\begin{aligned}
&\min_{\mathbf{Y}\in Ind, s>0} \|\mathbf{A} - s\mathbf{Y}\mathbf{Y}^T\|_F^2 \\
\Leftrightarrow &\max_{\mathbf{Y}\in Ind, s} 2sTr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2Tr(\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T) \\
\Leftrightarrow &\max_{\mathbf{Y}\in Ind, s} 2sTr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2Tr(\mathbf{Y}^T\mathbf{1}\mathbf{1}^T\mathbf{Y}) \\
\Leftrightarrow &\max_{\mathbf{Y}\in Ind, s} 2sTr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - s^2\|\mathbf{Y}\|_e
\end{aligned} \tag{16}$$

which completes the proof. □

According to the above theorem, we know that problem (13) is equivalent to the following problem

$$\max_{\mathbf{Y}\in Ind} Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - \frac{1}{2}\|\mathbf{Y}\|_e \tag{17}$$

According to the analysis in Sections 4.1 and 4.3, we know that problem (17) can be considered as a balanced min-cut problem, in which the exclusive lasso is used as a balance regularizer. However, the regularization parameter in problem (17) is fixed as $\frac{1}{2}$ and can not be adjusted according to the data. In Section 5.4, we will discuss how problem (15) automatically adjusts the balance parameter s.

Problem (15) can be solved with an alternative optimization approach. In the next two subsections, we show how to update $\mathbf{Y}$ and s.

5.1. Solving $\mathbf{Y}$ with s fixed

If $s > 0$ is fixed, we can obtain $\mathbf{Y}$ by solving the following problem

$$\min_{\mathbf{Y}\in Ind, \mathbf{G}=\mathbf{Y}} Tr(\mathbf{Y}^T\Theta\mathbf{G}) \tag{18}$$

where $\Theta = \frac{s}{2}\mathbf{1}\mathbf{1}^T - \mathbf{A}$.

The above problem is non-smooth and difficult to optimize. In this paper, we use the ALM algorithm described in Algorithm 1 to solve problem (18). According to Algorithm 1, we need to solve the following problem

$$\min_{\mathbf{Y}\in Ind, \mathbf{G}} Tr(\mathbf{Y}^T\Theta\mathbf{G}) + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{G} + \frac{1}{\mu}\Lambda\|_F^2 \tag{19}$$

An accurate, joint minimization with respect to both $\mathbf{Y}$ and $\mathbf{G}$ is difficult and costly. In this paper, we use the alternating direction method of multipliers (ADMM) to solve this problem. Specifically, we optimize problem (19) with respect to one variable when fixing the other variable, which result in the following two subproblems.

When $\mathbf{Y}$ is fixed, the Lagrangian function of problem (19) is

$$\mathcal{L}(\mathbf{G}) = Tr(\mathbf{Y}^T\Theta\mathbf{G}) + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{G} + \frac{1}{\mu}\Lambda\|_F^2 \tag{20}$$

Taking the derivative of $\mathcal{L}(\mathbf{G})$ with respect to $\mathbf{G}$ and setting it to zero, we have

$$\Theta^T\mathbf{Y} - (\mu(\mathbf{Y} - \mathbf{G}) + \Lambda) = 0 \tag{21}$$

which leads to

$$\mathbf{G} = \mathbf{Y} - \frac{1}{\mu}(\Theta^T\mathbf{Y} - \Lambda) \tag{22}$$

When $\mathbf{G}$ is fixed, problem (19) becomes

$$\min_{\mathbf{Y}\in Ind} Tr(\mathbf{Y}^T\Theta\mathbf{G}) + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{G} + \frac{1}{\mu}\Lambda\|_F^2 \tag{23}$$

Note that problem (23) is independent between different i, so we can solve the following problem individually for each $\mathbf{y}^i$ by solving the following problem

$$\min_{\mathbf{y}^i\in Ind} \mathbf{y}^i(\omega^i)^T + \frac{\mu}{2}\|\mathbf{y}^i - \mathbf{g}^i + \frac{1}{\mu}\lambda^i\|_2^2 \tag{24}$$

where ω^i is the i-th row of $\Omega = \Theta\mathbf{G}$. Problem (24) can be further rewritten as

$$\min_{\mathbf{Y}\in Ind} \|\mathbf{y}^i - (\mathbf{g}^i - \frac{\omega^i + \lambda^i}{\mu})\|_2^2 \tag{25}$$

Then the optimal solution of $\mathbf{y}^i$ is

$$y_{ij} =< j = \arg\max_{j'\in[1,c]} t_{ij'} > \tag{26}$$

where $< . >$ is 1 if the argument is true or 0 otherwise, and t_{ij} is defined as

$$t_{ij} = \mathbf{G}_{ij} - \frac{\Omega_{ij} + \Lambda_{ij}}{\mu} \tag{27}$$

5.2. Solving s with $\mathbf{Y}$ fixed

If $\mathbf{Y}$ is fixed, we can obtain s by solving the following problem

$$\min_{s>0} \left(s - \frac{Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_e}\right)^2 \tag{28}$$

The optimal solution of s is

$$s = \frac{Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{\|\mathbf{Y}\|_e} \tag{29}$$

5.3. The optimization algorithm

The detailed algorithm to solve problem (15), named the Self-Balanced Min-Cut (SBMC), is summarized in Algorithm 2. The balance parameter s and cluster indicator matrix $\mathbf{Y}$ are iteratively updated until convergence. In the new algorithm, we need $O(r_1(nc^2 + r_2nc^2))$ time to iteratively solve s and $\mathbf{Y}$ where r_1 is the number of iterations to update s and r_2 is the average number of iterations to update $\mathbf{Y}$. Here, the discrete solution $\mathbf{Y}$ converges very fast due to its limited solution space so r_2 is usually very small. Therefore, the SBMC has a time complexity of $O(n)$, which is same as the computational complexity of k-means. Compared to the conventional spectral clustering methods which have a time complexity of $O(n^3)$, the new algorithm has a significant reduction in computation.

Algorithm 2 Self-Balanced Min-Cut Algorithm to solve problem (15)

Input: An affinity matrix $\mathbf{A}$, parameter $\rho \in (1, 2)$.
Randomly initialize $\mathbf{Y}$.
repeat
  Update $s = \frac{Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})}{Tr(\mathbf{Y}^T\mathbf{1}\mathbf{1}^T\mathbf{Y})}$.
  Update $\Theta = \frac{s}{2}\mathbf{1}\mathbf{1}^T - \mathbf{A}$.
  Initialize $\mu > 0$ and $\Lambda \in \mathcal{R}^{n\times c}$.
  Randomly initialize $\mathbf{Y}$.
  repeat
    Update $\mathbf{G} = \mathbf{Y} - \frac{1}{\mu}(\Theta^T\mathbf{Y} - \Lambda)$.
    Update $\mathbf{Y}$ according to Eq. (26).
    Update $\Lambda = \Lambda + \mu(\mathbf{Y} - \mathbf{G})$ and $\mu = \rho\mu$.
  until problem (19) converges
until problem (15) converges
Output: The clustering result $\mathbf{Y}$.

5.4. Connection to conventional min-cut algorithms

Problem (15) can be rewritten as

$$\max_{\mathbf{Y}\in Ind, s} s(Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - \frac{s}{2}\|\mathbf{Y}\|_e) \tag{30}$$

According to the analysis in Sections 4.1 and 4.3, we know that $\max_{\mathbf{Y}\in Ind}(Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - \frac{s}{2}\|\mathbf{Y}\|_e)$ can be considered as balanced min-cut model, which tends to produce more balanced clustering results with bigger balance parameter s. According to Eq. (29), s is inversely proportional to $\|\mathbf{Y}\|_e$. The more balanced the cluster structure that the data contains, the smaller the $\|\mathbf{Y}\|_e$ is and the bigger the s is. Then solving $\max_{\mathbf{Y}\in Ind}(Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) - \frac{s}{2}\|\mathbf{Y}\|_e)$ results in more balanced clusters. Therefore, the new model can automatically adjust the balance parameter according to the learnt cluster indicator matrix $\mathbf{Y}$. On the other hand, maximizing s will maximize $Tr(\mathbf{Y}^T\mathbf{A}\mathbf{Y})$ which is the objective

function of min-cut. Therefore, the new model can simultaneously minimize the graph cut and balance the partition across all clusters.

According to Eq. (29), it can be verified that $s \in (0, 1]$. If $\mathbf{X}$ consists of balanced and significant cluster structure (with big $Tr(\mathbf{Y}^T\mathbf{AY})$), problem (15) will learn a big s. If a data set $\mathbf{X}$ consists of imbalanced and insignificant cluster structure (with small $Tr(\mathbf{Y}^T\mathbf{AY})$), problem (15) will learn a small s. In such case, $s \to 0$ and solving problem (15) will be approximately equivalent to solving the following problem

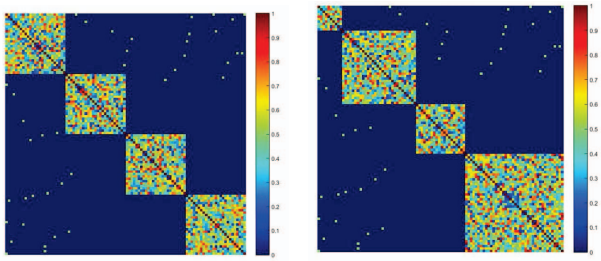
$$\max_{\mathbf{Y} \in Ind} Tr(\mathbf{Y}^T\mathbf{AY}) \tag{31}$$

which is exactly the min-cut problem in Eq. (12).

6. Experimental Results and Analysis

In this section, we present the experiments conducted on both synthetic and benchmark data sets to demonstrate the efficiency and effectiveness of the proposed method.

6.1. Experiments on synthetic data



(a) Original graph D_1. (b) Original graph D_2.

Figure 1. The similarity matrices of two synthetic data sets D_1, D_2.

We generated two synthetic data sets, i.e., D_1 and D_2, which are both 100×100 matrices for this experiment. D_1 contains four balanced clusters which are diagonally arranged, each of which is a 25×25 block matrix. D_2 contains four imbalanced clusters which are also diagonally arranged, with the sizes of 10×10, 30×30, 20×20 and 40×40. The data within each block are the affinities of two corresponding points in one cluster, while the data outside all blocks are noises. The affinity data within each block is randomly generated with values in $[0, 1]$, while the noise data is randomly generated with values in $[0, \psi]$ where the noise level ψ is a given parameter. Moreover, to make this clustering task more challenging, we randomly pick up 25 noise data points and set their values to be 1. Figure 1(a) and 1(b) show the similarity matrices of both D_1 and D_2 (with $\psi = 0$).

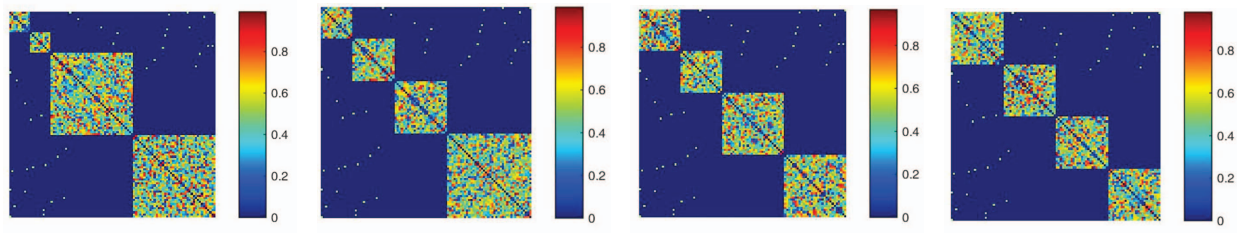
We compared SBMC with Normalized Cut (NCut) [15], Ratio Cut (RCut) [12], Multiclass Spectral Clustering (MSC) [24], Spectral Embedded Clustering (SEC) [19] and Constrained Laplacian Rank (CLR2-Constrained Laplacian Rank with ℓ_2 norm, and CLR1-Constrained Laplacian Rank with ℓ_1 norm) [18]. By setting a set of 10 parameters $\psi = \{0.5, 0.55, \ldots, 0.95\}$, we can generate 10 data sets from D_1 and D_2, respectively. For each data set, we set the neighborhood parameter $k = 10$ to construct a sparse k nearest neighbors affinity matrix with the similarity matrix construction method in[18], and used the affinity matrix to run the six methods in order to perform fair comparison. The regularization parameter in SEC was set to seven values $\{10^{-3}, \ldots, 10^3\}$. Since CLR2, CLR1 and SBMC are parameter free [1], we ran each of them on each data set 100 times and selected the best clustering result according to their objective function. For NCut and RCut, we selected the clustering result with the minimal objective function from 100 k-means clustering results on each data set. For each parameter in SEC, we selected the clustering result with the minimal objective function from 100 k-means clustering results on each data set, then the average clustering results across multiple parameters were computed. The clustering results in terms of accuracy are shown in Table 1. From this table, we can see that SBMC produced nearly similar results as CLR2 and CLR2 on both data sets, and significantly outperforms other clustering algorithms. We also observe that the clustering results of $NCut$ and $RCut$ are instable due to the affection of noise data.

Table 1. Average accuracies of six clustering algorithms on D_1 and D_2 (with different noise level).

Data sets	ψ	NCut	RCut	MSC	CLR2	CLR1	SBMC
D_1	0.5	0.25	0.5	1	1	1	1
	0.55	0.75	0.25	1	1	1	1
	0.6	1	0.5	1	1	1	1
	0.65	1	1	1	1	1	1
	0.7	1	0.5	0.99	1	1	0.99
	0.75	0.72	0.49	0.95	0.98	0.98	0.96
	0.8	0.71	0.7	0.87	0.89	0.89	0.87
	0.85	0.51	0.25	0.53	0.48	0.58	0.49
	0.9	0.25	0.37	0.4	0.36	0.36	0.41
	0.95	0.31	0.25	0.35	0.32	0.32	0.35
D_2	0.5	0.4	1	1	1	1	1
	0.55	1	0.7	1	1	1	1
	0.6	0.9	0.7	1	1	1	1
	0.65	0.9	0.9	1	0.99	0.99	1
	0.7	0.9	0.7	0.96	1	1	1
	0.75	0.96	0.99	0.9	0.97	0.97	0.99
	0.8	0.4	0.88	0.7	0.94	0.94	0.85
	0.85	0.68	0.61	0.54	0.73	0.71	0.63
	0.9	0.51	0.43	0.4	0.48	0.48	0.36
	0.95	0.39	0.38	0.35	0.38	0.38	0.39

To show the relationship between the learnt balance parameter s and the cluster structure, we generated four synthetic data sets D_3, D_4, D_5 and D_6 in the same way as we

[1]The parameter ρ in SBMC can be randomly initialized in $(1, 2)$.



(a) D_3 with learnt $s = 0.206$. (b) D_4 with learnt $s = 0.245$. (c) D_5 with learnt $s = 0.269$. (d) D_6 with learnt $s = 0.279$.

Figure 2. The heat maps of four synthetic data sets and the learnt balance parameters s by SBMC.

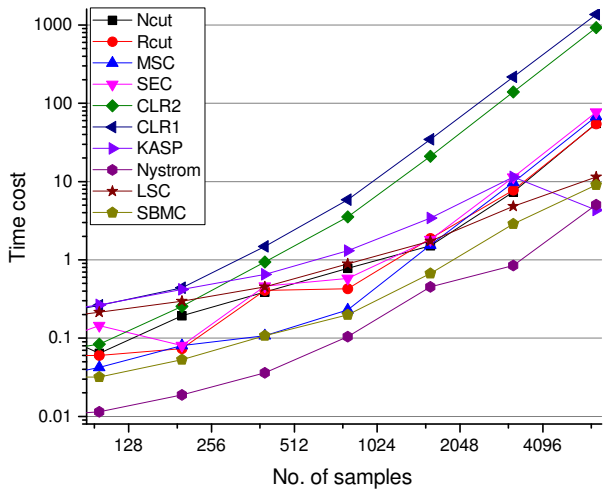


Figure 3. Time costs of 10 algorithms on 8 synthetic data sets, in which we use logarithmic scales on both the horizontal and vertical axes.

generated D_2 except that the numbers of samples in each cluster were different. Figure 2 shows the similarity matrices of the four data sets. We can see that D_6 consists of balanced clusters and the other three data sets consist of imbalanced clusters. We ran SBMC 100 times on the four data sets and selected the clustering result with the minimal objective function. The learnt s on D_3, D_4, D_5 and D_6 are 0.206, 0.245, 0.269 and 0.279 respectively. This indicates that SBMC learns bigger s for data with more balanced clusters. This experimental result is consistent with the analysis in Section 5.4.

The last experiment was conducted to study the time cost of SBMC. We generated a set of 8 synthetic data sets with the number of samples as $\{50, 100, \ldots, 6400\}$, each containing 10 features. We compared the execution time of one run of 10 clustering algorithms, i.e., Normalized Cut (NCut) [15], Ratio Cut (RCut) [12], Multiclass Spectral Clustering (MSC) [24], Spectral Embedded Clustering (SEC) [19] and Constrained Laplacian Rank (CLR2-Constrained Laplacian Rank with ℓ_2 norm, and CLR1-Constrained Laplacian Rank with ℓ_1 norm) [18], KASP (k-means-based Approximate Spectral Clustering) [22], Nyström [6], LSC (Landmark-based Spectral Clustering) [2] and our method SBMC. Here, KASP, Nyström and LSC are three approximate spectral clustering methods. The results are shown in Figure 3. From this figure, we can see that SMBC converged very fast and showed nearly linear relationship with the increase of data size. SMBC even spent less time than the approximate spectral clustering algorithms LSC. Due to multiple eigendecompositions, CLR2 and CLR1 showed the highest time costs and their time costs increased rapidly with the increase of data size. This experiment result indicates that the new method SMBC is scalable to large scale data.

6.2. Experiments on benchmark image data sets

In this experiment, we compared SMBC with 9 clustering algorithms which were used in the last experiment of Section 6.1. Seven benchmark image data sets were selected for this experiment:

- **Corel-5k** image data set was downloaded from Feiping Nie's page [2]. This data set contains 5000 images from 50 classes.
- **MnistData-05** digit data set was downloaded from Feiping Nie's page. This data set contains 3495 handwritten digits sampled from the original Mnist dataset.
- **MSRA25** face data set was downloaded from Feiping Nie's page. This data set contains 1799 images from 12 individuals.
- **Yale-32x32** and **Yale-64x64** face data sets were downloaded from Deng Cai's page [3]. The two data set contain 165 grayscale images from 15 individuals.
- **YaleB-32x32** face data set was downloaded from Deng Cai's page. This data set contains 2414 images from 38 individuals.
- **NUS-WIDE** data set is used in [16]. In this data set, 12 categories about animal concept are selected from the NUS data set [9].

[2] http://www.escience.cn/people/fpnie/index.html#
[3] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

Table 2. Performance comparison of the average clustering accuracies ($Accuracy \pm StandardDeviation$).

Data	**Corel-5k**	**MnistData-05**	**MSRA25**	**Yale-32x32**	**Yale-64x64**	**YaleB-32x32**	**NUS-WIDE**
NCut	0.142 ± 0.03	0.294 ± 0.20	0.261 ± 0.15	0.215 ± 0.18	0.365 ± 0.19	0.104 ± 0.06	0.178 ± 0.06
RCut	0.100 ± 0.05	0.484 ± 0.23	0.497 ± 0.07	0.336 ± 0.05	0.294 ± 0.17	0.123 ± 0.08	0.188 ± 0.11
MSC	**0.185** ± 0.00	**0.656** ± 0.03	0.484 ± 0.22	**0.428** ± 0.02	**0.583** ± 0.04	0.196 ± 0.06	**0.221** ± 0.02
SEC	0.129 ± 0.01	0.359 ± 0.10	0.428 ± 0.08	0.351 ± 0.02	0.312 ± 0.05	0.163 ± 0.04	0.168 ± 0.02
CLR2	0.097 ± 0.02	0.529 ± 0.05	**0.554** ± 0.01	0.418 ± 0.03	0.550 ± 0.04	**0.200** ± 0.03	0.145 ± 0.02
CLR1	0.067 ± 0.00	0.422 ± 0.06	**0.561** ± 0.01	0.383 ± 0.03	0.538 ± 0.05	0.161 ± 0.05	0.126 ± 0.01
KASP	0.142 ± 0.05	0.250 ± 0.08	0.294 ± 0.07	0.248 ± 0.06	0.282 ± 0.08	0.103 ± 0.02	0.147 ± 0.03
Nyström	**0.211** ± 0.05	0.332 ± 0.08	0.268 ± 0.10	0.261 ± 0.05	0.303 ± 0.08	0.281 ± 0.09	0.087 ± 0.01
LSC	0.169 ± 0.04	0.508 ± 0.08	0.530 ± 0.07	0.407 ± 0.04	0.489 ± 0.06	0.116 ± 0.04	0.206 ± 0.02
SBNC	0.134 ± 0.01	**0.541** ± 0.11	0.506 ± 0.05	**0.435** ± 0.03	**0.559** ± 0.07	**0.214** ± 0.04	**0.240** ± 0.01

* The best 2 methods for each data set are highlighted in bold.

We used the similarity construction method in [18] to construct an affinity matrix for each data set to run seven algorithms excluding KASP, Nyström and LSC, where the neighborhood parameters were set to $\{5, 10, \ldots, 25\}$ for two small size data sets D_4 and D_5, and $\{10, 20, \ldots, 100\}$ for the other five data sets. Since CLR2, CLR1 and SBMC are parameter free [4], we ran each of them on each data set 100 times and selected the best clustering result according to their objective function. For NCut and RCut, we selected the clustering result with the minimal objective function value from 100 k-means clustering results on each data set. The regularization parameter in SEC was set to seven values $\{10^{-3}, \ldots, 10^3\}$. For each parameter in SEC, we selected the clustering result with the minimal objective function value from 100 k-means clustering results on each data set, then the average clustering results across multiple parameters were computed. We set the number of centers in KASP, the number of samples in Nyström and the number of landmarks in LSC as the same values on each data set, i.e., $\{10, 20, \ldots, 50\}$ for two small size data sets D_4 and D_5 and $\{100, 200, \ldots, 500\}$ for the other five data sets. For each parameter in the three algorithms, we selected the clustering result with the minimal objective function value from 100 k-means clustering results on each data set, then the average clustering results across multiple parameters were computed.

We show the average accuracies and the standard deviations of 10 clustering algorithms on 7 data sets in Table 2. From this table, we can see that SBMC outperformed other methods on most data sets. Specifically, SBMC produced the best results on D_4, D_6 and D_7, and the second best results on D_2 and D_5. This indicates that SBMC can produce good results on real-life data sets.

[4]The parameter ρ in SBMC can be randomly initialized in $(1, 2)$.

7. Conclusions

In this paper, we have proposed a self-balanced min-cut (SBMC) method for image clustering. The new method implicitly introduces the Exclusive Lasso as a balance regularizer in order to produce balanced partition. The regularization parameter, named the balance parameter in the new method, can be automatically learnt during the clustering process. To solve the new model, we have proposed an iterative algorithm SBMC which has a time complexity of $O(n)$ where n is the number of samples. In comparison with the conventional spectral clustering methods with high time complexities of $O(n^3)$, the new method has great computational advantage especially on large scale data sets. Above all, compared to the sampling based spectral clustering methods such as KASP, Nyström and LSC, the new method uses all data without sampling. Extensive experiments on both synthetic data sets and benchmark image data sets show the efficiency and effectiveness of our method compared to the state-of-the-art methods.

Several questions remain to be investigated in our future work:

1. It is still a challenge work to effectively optimize discrete variables in our method, especially for large scale image data. In the future, we will study more efficient optimization methods for our model.

2. In our model, the cluster indicator matrix consists of discrete values and each object only belongs to one cluster. In the future work, we will try to relax the discrete cluster indicator matrix and introduce fuzzy technique to obtain more feasible partition.

Acknowledgment

This research was supported by NSFC under Grant no.61305059, 61473194 and 61502177, Fundamental Research Funds for the Central Universities under Grant D2172500.

References

[1] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, 2 edition, 1996.

[2] D. Cai and X. Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1669–1680, 2015.

[3] X. Cai, F. Nie, W. Cai, and H. Huang. New graph structured sparsity model for multi-label image annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 801–808, 2013.

[4] X. Cai, F. Nie, H. Huang, and F. Kamangar. Heterogeneous image feature integration via multi-modal spectral clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1977–1984. IEEE, 2011.

[5] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, 13(9):1088–1096, 1994.

[6] W. Y. Chen, Y. Song, H. Bai, C. J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(3):568–586, 2011.

[7] X. Chen, X. Xu, Y. Ye, and J. Z. Huang. TW-k-means: Automated Two-level Variable Weighting Clustering Algorithm for Multi-view Data. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):932–944, 2013.

[8] X. Chen, Y. Ye, X. Xu, and J. Z. Huang. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45(1):434–446, 2012.

[9] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 48:1–48:9, New York, NY, USA, 2009. ACM.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 26(2):214–225, 2010.

[12] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[13] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, 2009.

[14] M. Li, J. T. Kwok, and B. L. Lu. Making large-scale nyström approximation possible. In *International Conference on Machine Learning*, pages 631–638, 2010.

[15] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[16] F. Nie, J. Li, X. Li, et al. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 1881–1887, 2016.

[17] F. Nie, X. Wang, and H. Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986. ACM, 2014.

[18] F. Nie, X. Wang, M. Jordan, and H. Huang. The constrained laplacian rank algorithm for graph-based clustering. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1969–1976, 2016.

[19] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang. Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, 22(11):1796–808, 2011.

[20] H. Shinnou and M. Sasaki. Spectral clustering for a large data set by reducing the similarity matrix size. In *International Conference on Language Resources and Evaluation, Lrec 2008, 26 May - 1 June 2008, Marrakech, Morocco*, pages 201–204, 2008.

[21] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[22] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–916, 2009.

[23] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10):2761–2773, 2010.

[24] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of IEEE International Conference on Computer Vision*, pages 313–319 vol.1, 2003.

[25] Y. Zhou, R. Jin, and S. C. H. Hoi. Exclusive lasso for multi-task feature selection. In *JMLR Workshop and Conference Proceedings: 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 988–995, Chia Laguna Resort, Sardinia, Italy, 2010.