COMPARISONS OF DIFFERENT METHODS FOR BALANCED DATA CLASSIFICATION UNDER THE DISCRETE NON-LOCAL TOTAL VARIATIONAL FRAMEWORK

SHIXIU ZHENG 1,2, ZHILEI XU 1
HUAN YANG 1, JINTAO SONG 1 AND ZHENKUAN PAN 1 *

1 College of Computer Science and Technology, Qingdao University
Qingdao 266071, China
2 College of Business, Qingdao University
Qingdao 266071, China

(Communicated by Yiming Ying)

Abstract. Because balanced constraints can overcome the problems of trivial solutions of data classification via minimum cut method, many techniques with different balanced strategies have been proposed to improve data classification accuracy. However, their performances have not been compared comprehensively so far. In this paper, we investigate seven balanced classification methods under the discrete non-local total variational framework and compare their accuracy performances on graph. The two-class classification problem with equality constraints, inequality constraints and Ratio Cut, Normalized Cut, Cheeger Cut models are investigated. For cases of equality constraint, we firstly compare the Penalty Function Method (PFM) and the Augmented Lagrangian Method (ALM), which can transform the constrained problems into unconstrained ones, to show the advantages of ALM. The other cases are all solved using the ALM also. In order to make the comparison fairly, we solve all models using ALM method and using the same proportion of fidelity points and the same neighborhood size on graph. Experimental results demonstrate ALM with the equality balanced constraint has the best classification accuracy compared with other six constraints.

1. Introduction. Data classification is one of fundamental problems in the fields of data mining, machine learning, pattern recognition, computer vision and so on. Its task is to divide a specific dataset into different parts via labels without overlapping and vacuum problem. The minimum cut method on a graph to realize data classification has been accepted as an efficient tool recently, but the intrinsic trivial solution problems must be overcome, which exists also in data clustering problems [15]. The trivial solution problem is that in many cases, the solution of mincut simply separates one individual vertex from the rest of the graph, this is obviously not what we want to achieve.

Data classification by minimum cut [8] on a discrete graph usually starts with an undirected weighted graph, which can be denoted as a triple $G(V, E, W)$. Where $V$ is the collection of $n$ vertexes, and $E$ is the collection of all edges describing the
similarities of adjacent vertexes, $W$ is the collection of weights defined on edges as $w(x, y) \in W$, where $x, y \in V$, $w(x, y) \geq 0$ and $w(x, y) = w(y, x)$. Additionally, the degree of a vertex, degree of a graph and number of vertexes are defined as $d(x) = \sum_{y \in V} w(x, y)$, $\text{vol}(V) = \sum_{x \in V} d(x)$, and $|V| = n$ respectively.

Based on the above mentioned descriptions, the data classification problem of two-class can be stated as the following minimum cut problem.

$$\min \text{Cut}(V_1, V_2) = \text{Cut}(V_1, V_2),$$

where $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$, $\text{Cut}(V_1, V_2) = \sum_{x \in V_1, y \in V_2} w(x, y)$.

In order to avoid the problem of trivial solution of (1), some researchers have investigated many balanced cut approaches, such as Ratio Cut (RC), Cheeger Cut (CC) [3, 13, 24, 21, 5, 17] and Normalized Cut (NC) [20] as follows

$$\min \text{RC}(V_1, V_2) = \text{Cut}(V_1, V_2) \left( \frac{1}{|V_1|} + \frac{1}{|V_2|} \right),$$

$$\min \text{NC}(V_1, V_2) = \text{Cut}(V_1, V_2) \left( \frac{1}{\text{vol}(V_1)} + \frac{1}{\text{vol}(V_2)} \right),$$

$$\min \text{CC}(V_1, V_2) = \min_{V_1 \subset V} \min_{V_2 \subset V} \frac{\text{Cut}(V_1, V_2)}{|\text{vol}(V_1)|, |\text{vol}(V_2)|}.$$  

where, $\text{vol}(V_1) = \sum_{x \in V_1} d(x)$ is the degree of sub-set $V_1$. All are NP hard problems [23]. In order to solve them properly, a binary label function $u(x) \in \{0, 1\}$, $x \in V$, $u(x) = \begin{cases} 1 & \text{if } x \in V_1 \\ 0 & \text{otherwise} \end{cases}$.

Additionally, there are some balanced methods with equality and inequality constraints to deal with the trivial problem. The first one is mainly used to the cases that the number of each class is known exactly and the second one can be used to the cases that only the range of each class is known approximately. In semi-supervised classification problems, one usually assumes that the regions should have approximately equal sizes. The simplest way to achieve this goal is to give size for each class, and the corresponding equality constraint [23, 14, 2] can be stated as

$$|V_i| = \alpha_i, \ (i = 1, 2).$$

In this paper, we use two different methods to implement constraints in (5). One is equality constraint Penalty Function Method (EC-PFM), the other is equality constraint Augmented Lagrangian Method (EC-ALM). The inequality constraint [3] can be stated as

$$\alpha_i^u \leq |V_i| \leq \alpha_i^l, \ (i = 1, 2),$$

where $\alpha_i^u$ is the upper bound of the size of class $i$ and $\alpha_i^l$ is the lower bound. Note that (6) becomes (5) if $\alpha_i^l = \alpha_i^u = \alpha_i$. When low bound does not exist, we can use a single direction inequality constraint. Therefore, the method (6) can represent single direction inequality constraint (SDIC) and double directions inequality constraint (DDIC). If piecewise constant label functions are used to partition different data classes, the Cut is just the same as non-local total variation (NL-TV) whose local version was proposed originally in image processing [19]. This equivalent relation has already been successfully applied to semi-supervised data classification [24, 20] with clearer boundary than Tikhonov model [22]. The systematic definitions
of non-local operators associated with NL-TV were given in [9, 10] in continuous domain, which were extended to discrete non-local operators on graphs [25, 7]. These works have laid the foundation for the application of NL-TV in multi-class data classification and clustering [3], which is beyond the research scope of this paper. The research focus of this paper is to investigate the dependences of two-class data classification accuracy on different balanced constraints to guide their applications to multi-class classification. To our knowledge, there is no comprehensive studies on this topic from theoretic or experimental perspectives. Due to the mathematical difficulties for theoretic investigations, we resort to numerical experiments to draw some empirical conclusions. We start this work from two-class classification problems on undirected weighted graph using discrete NL-TV models, and then solve them using convex relaxification and thresholding strategies. For the models with explicit equality constraints, we design their Penalty Function Method and Augmented Lagrangian Method, and for the other cases, we design their Augmented Lagrangian Method. The paper starts with introduction of the non-local operator on graph in section 2. Section 3 provides seven schemes of balanced classification with or without explicit constraints, along with their algorithms. In section 4, numerous experiments are given to illustrate the effect of different constraints on accuracy. Some conclusions are drawn in section 5 finally.

2. Non-local total variation on graph. In this section, we introduce some basic definitions on discrete non-local operator [25, 7] for two-class data classification on an undirected weighted graph. To deal with this problem, let dataset $V$ be divided into $V_1$ and $V_2$, and introduce a binary label function $u(x)$ to partition them, i.e.,

$$u(x) \in \{0, 1\}, x \in V, u(x) = \begin{cases} 1 & \text{if } x \in V_1 \\ 0 & \text{otherwise} \end{cases}.$$  (7)

The definition of discrete non-local derivative on an edge starting from $x$ is defined as

$$\partial_x u(x) = (u(y) - u(x)) \sqrt{w(x,y)}, y \in N(x),$$  (8)

where, $N(x)$ means the neighbourhood of $x$. Based on it, the discrete non-local gradient, discrete non-local divergence of non-local vector $v(x, y)$, inner product of two non-local vector $v_1$ and $v_2$, the norm of the non-local gradient, and the non-local Laplacian are expressed as

$$\nabla_w u(x, y) = (u(y) - u(x)) \sqrt{w(x,y)}, y \in N(x),$$  (9)

$$(\nabla_w v) (x) = \sum_{y \in N(x)} (v(x, y) - v(y, x)) \sqrt{w(x,y)},$$  (10)

$$|\nabla_w u| (x) = \sqrt{\sum_{y \in N(x)} (u(y) - u(x))^2 w(x,y)},$$  (11)

$$\nabla_w \cdot (\nabla_w u) (x) = 2 \sum_{y \in N(x)} (u(y) - u(x)) w(x,y),$$  (12)

respectively. $\nabla_w u(x, y)$ is a non-local vector. Considering the equivalence of Cut and discrete NL-TV, [24, 11] rewrite the minimum Cut (1) as

$$\min \text{Cut} (V_1, V_2) = \min \sum_{x \in V_1, y \in V_2} w(x,y) = \min_{u \in \{0, 1\}} \sum_{x \in V} |\nabla_w u| (x,y),$$  (13)
\[ \min_{u \in [0,1]} \left\{ E(u) = \sum_{x \in V} |\nabla_w u(x)| \right\}, \quad (14) \]

which can be solved via convex relaxification and thresholding techniques. In order to overcome the problem of trivial solutions, various balanced constraint strategies have been proposed, well rewrite the corresponding discrete variational models combining them with (14) in the next section.

3. Different models of balanced classification and their algorithms. In this part, we consider the cases of explicit equality constraints, inequality constraints and Ratio Cut, Normalized Cut, Cheeger Cut without explicit constraints. The inequality constraints include single direction and double direction constraints. We use PFM or ALM to solve the corresponding minimization problems.

3.1. Equality constraint with penalty function method (EC-PFM). Using binary function (7), the equality constraint (5) can be written as

\[ \sum_{x \in V_1} u(x) = V - \sum_{x \in V_2} u(x) = \alpha. \quad (15) \]

Augmenting this constraint to (14), and using PFM leads to the convex model

\[ \min_{u \in [0,1]} \left\{ E(u) = \sum_{x \in V} |\nabla_w u(x)| + \frac{\mu_0}{2} \left( \sum_{x \in V_1} u(x) - \alpha \right)^2 \right\}, \quad (16) \]

where, \( \mu_0 \) is a penalty parameter. To solve it easily, we design its alternating optimization scheme by introducing a non-local auxiliary variable \( v = \nabla_w u \) as

\[ (u^{k+1}, v^{k+1}) = \arg \min_{u, v \in [0,1]} E(u, v) = \begin{cases} \sum_{x \in V} |v| + \frac{\mu_0}{2} \left( \sum_{x \in V} u(x) - \alpha \right)^2 \\
+ \sum_{x \in V} \lambda_1^k \cdot (v - \nabla_w u) \\
+ \mu_1 \sum_{x \in V} (v - \nabla_w u)^2 \end{cases}, \quad (17a) \]

\[ \lambda_1^{k+1} = \lambda_1^k + \mu_1 (v^{k+1} - \nabla_w u^{k+1}). \quad (17b) \]

where \( \lambda_1 \) is a Lagrange multiplier and \( \mu_1 \) is another penalty parameter. The alternating optimization is implemented by solving a series of sub-problems alternatively in each step, i.e., minimize only one sub-problem while fix other variables temporally. So (17a) is divided into two sub-problems as

\[ u^{k+1} = \arg \min_{u \in [0,1]} E_1(u) = E(u, v^k), \quad (18a) \]

\[ v^{k+1} = \arg \min_{v} E_2(v) = E(u^{k+1}, v), \quad (18b) \]

Employing the standard variational method to (18a), we obtain the Euler-Lagrangian equation on \( u \) as

\[ \begin{cases} \mu_0 \left( \sum_{x \in V_1} u(x) - \alpha \right) + \nabla_w \cdot \lambda_1^k + \mu_1 \nabla_w \cdot (v^k - \nabla_w u) = 0. \quad (19) \end{cases} \]
Its discrete iterative formulation is
\[
\hat{u}_{k+1}^2(x) = \frac{2\mu_1}{\sum_{y \in N(x)} w(x,y)} \sum_{y \in N(x)} w(x,y) \nabla_w \cdot \left( \alpha - \sum_{x \in V_1} u^k(x) \right) - \lambda_k - \mu_0 \left( \nabla_w \cdot \left( \nabla_w u_k^2(x) \right) \right) + \mu_1 \nabla_w \cdot \left( \nabla_w u_k^2(x) \right) + \mu_0 \left( \nabla_w \cdot \left( \nabla_w u_k^2(x) \right) \right)
\].
\]
\[
\hat{v}_{k+1}^2(x) = \max \left( \min \left( 1, \hat{u}_{k+1}^2(x) \right) \right)
\]
Using the same method to (19b), we get the generalized soft thresholding formula
\[
v_{k+1}^2(x) = \max \left( \left| \nabla_w \left( \nabla_w u_{k+1}^2(x) \right) \right| - \frac{\lambda_k}{\mu_1} \frac{\nabla_w \cdot \left( \nabla_w u_{k+1}^2(x) \right)}{0} \right)
\]
After the energy approaches minimum, \( u \in [0,1] \) should be recovered via thresholding to \( u \in \{0,1\} \) as
\[
u(x) = \begin{cases} 
1 & \text{if } u(x) > 0.5 \\
0 & \text{otherwise}
\end{cases}
\]
To summarize the algorithm introduced in this sub-section, we write down the pseudo-code of algorithm EC-PFM here.

**Algorithm 1 EC-PFM**

**Initialization**
\( u_0 \) given by random initialization
\( \lambda_0^2 = 0 \)

**while** energy not converged **do**
\( u_{k+1} \) given by (20)
\( v_{k+1} \) given by (21)
\( \lambda_{k+1} \) given by (17b)

**end while**

3.2. Equality constraint with augmented Lagrange method (EC-ALM).

in order to circumvent the difficulty of the strongly dependence of the optimization problem on penalty parameter \( \mu_0 \) in the previous sub-section, we employ ALM method to enforce the equality constraint (15). The modified energy functional is
\[
\min_{u \in [0,1]} E(u,v) = \left\{ \sum_{x \in V} |\nabla_w u(x)|^2 + \lambda_0 \left( \sum_{x \in V} u(x) - \alpha \right) \right\} + \frac{\mu_0}{2} \left( \sum_{x \in V} u(x) - \alpha \right)^2
\]
where, \( \mu_0 \) s a penalty parameter and \( \lambda_0 \) is Lagrange multiplier. To solve it easily, we design its alternating optimization scheme by introducing a non-local auxiliary variable \( v = \nabla_w u \) as
\[
(u_{k+1}, v_{k+1}) = \arg \min_{\forall, u \in [0,1]} E(u,v) = \left\{ \sum_{x \in V} |v|^2 + \lambda_0 \left( \sum_{x \in V} u(x) - \alpha \right) \right\} + \frac{\mu_0}{2} \left( \sum_{x \in V} u(x) - \alpha \right)^2 + \sum_{x \in V} \lambda_1 \cdot (v - \nabla_w u)
\]
\[
\lambda_{1}^{k+1} = \lambda_{1}^{k} + \mu(v^{k+1} - \nabla w u^{k+1}), \quad (24b)
\]
\[
\lambda_{0}^{k+1} = \lambda_{0}^{k} + \mu_{0} \left( \sum_{x \in V} u^{k+1}(x) - \alpha \right), \quad (24c)
\]

where \( \lambda_{1} \) is a Lagrange multiplier and \( \mu_{1} \) is another penalty parameter. The sub-problems of optimization on \( u \) and \( v \) have the same forms as (18a) and (18b) respectively. Using the similar process as the previous sub-section, we get the Euler-Lagrange equation on \( u \)

\[
\begin{cases}
\lambda_{0}^{k} + \mu_{0} \left( \sum_{x \in V_{i}} u(x) - \alpha \right) + \nabla w \cdot \lambda_{1}^{k} + \mu_{1} \nabla w \cdot (v^{k} - \nabla w u) = 0,
\end{cases}
\]

with its discrete iterative formulation

\[
\begin{align*}
\bar{u}^{k+1}(x) &= \frac{2\mu_{1} \sum_{y \in \mathcal{N}(x)} u^{k}(y) w(x,y) - \nabla w \cdot \lambda_{1}^{k} - \mu_{1} \nabla w \cdot (v^{k} - \nabla w u)}{2 \mu_{1} \sum_{y \in \mathcal{N}(x)} w(x,y)} \\
u^{k+1}(x) &= \max \left( \min \left( 1, \bar{u}^{k+1}(x) \right) \right)
\end{align*}
\]

(26)

\( v^{k+1} \) is same as eq.(21).

The algorithm of EC-ALM for solving (24) is stated as follows.

**Algorithm 2 EC-ALM**

\begin{itemize}
\item \( u^{0} \) given by random initialization
\item \( \lambda_{0}^{0}, \lambda_{1}^{0} \leftarrow 0 \)
\item **while** energy not converged **do**
\item \( u^{k+1} \) given by (26)
\item \( v^{k+1} \) given by (21)
\item \( \lambda_{0}^{k+1} \) given by (24c)
\item \( \lambda_{1}^{k+1} \) given by (24b)
\item **end while**
\end{itemize}

3.3. Single direction inequality constraint (SDIC). for the single direction inequality constraint \( \sum_{x \in V} u(x) \leq \alpha \), using the KKT method for optimization problem with inequality, (14) can be transformed into the following equivalent form

\[
\begin{cases}
\min_{u \in [0,1]} \{ E(u) = \sum_{x \in V} |\nabla w u| (x) + \lambda_{0} \left( \sum_{x \in V} u(x) - \alpha \right) \}.
\end{cases}
\]

(27)

After introducing non-local auxiliary variable and some Lagrangian multipliers, penalty parameters, we get it’s alternating direction optimization formulation

\[
(u^{k+1}, v^{k+1}) = \arg \min_{u, v \in [0,1]} E(u, v) = \left\{ \sum_{x \in V} |v| + \lambda_{0}^{k} \left( \sum_{x \in V} u(x) - \alpha \right) \right\} + \sum_{x \in V} \lambda_{1}^{k} \cdot (v - \nabla w u) + \mu_{1} \sum_{x \in V} (v - \nabla w u)^{2},
\]

(28a)

\[
\lambda_{1}^{k+1} = \lambda_{1}^{k} + \mu(v^{k+1} - \nabla w u^{k+1}),
\]

(28b)
\[ \lambda^{k+1}_0 = \max \left( \lambda^k_0 + \mu_0 \left( \sum_{x \in V} u^{k+1}(x) - \alpha \right), 0 \right). \]  

(28c)

Solving (28a) via the same procedure as previous sub-sections, we get the Euler-Lagrange equation on \( u \)
\[
\begin{align*}
\lambda^k_0 + \nabla_w \lambda^k_1 + \mu_1 \nabla_w \cdot (v^k - \nabla_w u) &= 0, \\
u \in [0, 1],
\end{align*}
\]

(29)

and its discrete iterative formulation
\[
\begin{align*}
\tilde{u}^{k+1}(x) &= \frac{2\mu_1}{2\mu_1} \sum_{y \in N(x)} u^k(y) w(x,y) \nabla_w \cdot (v^k - \nabla_w u) \\
u^{k+1}(x) &= \max \left( \min \left( 1, \tilde{u}^{k+1}(x) \right) \right).
\end{align*}
\]

(30)

\( v^k \) is the same as (21) also. The corresponding algorithm is as follows.

**Algorithm 3 SDIC**

\[
\begin{align*}
u^0 \text{given by random initialization} \\
\lambda^0_0, \lambda^0_1 &\leftarrow 0 \\
\text{while energy not converged do} \\
\quad u^{k+1} \text{given by (30)} \\
\quad v^{k+1} \text{given by (21)} \\
\quad \lambda^{k+1}_0 \text{given by (28b)} \\
\quad \lambda^{k+1}_1 \text{given by (28c)}
\end{align*}
\]

end while

3.4. **Double direction inequality constraint (DDIC).** For the problem with double directional inequality constraint \( \alpha - \varepsilon \leq \sum u(x) \leq \alpha + \varepsilon \), we use KKT method to reformulate (14) to

\[
\begin{align*}
\min_{u \in [0, 1]} E(u) &= \left\{ \sum_{x \in V} \left| \nabla_w u \right|(x) + \lambda_0 \left( \sum_{x \in V_1} u(x) - \alpha - \varepsilon \right) \left( \sum_{x \in V_1} u(x) - \alpha + \varepsilon \right) \right\} , \\
\text{s.t.} \quad \alpha - \varepsilon \leq \sum_{x \in V} u(x) \leq \alpha + \varepsilon
\end{align*}
\]

(31)

Using the same procedure as the last sub-section, we get

\[
(u^{k+1}, v^{k+1}) = \arg \min_{u, v \in [0, 1]} E(u, v) = \left\{ \sum_{x \in V} \left| v \right| \right. \\
+ \lambda_0 \left( \sum_{x \in V_1} u(x) - \alpha - \varepsilon \right) \left( \sum_{x \in V_1} u(x) - \alpha + \varepsilon \right) \\
+ \lambda^k_1 \cdot (v - \nabla_w u) \\
+ \mu_1 \left( \sum_{x \in V} (v - \nabla_w u)^2 \right) \left\}
\]

(32a)

\[ \lambda^{k+1}_1 = \lambda^k_1 + \mu (v^{k+1} - \nabla_w u^{k+1}) , \]

(32b)

\[ \lambda^{k+1}_0 = \max \left( \lambda^k_0 + \mu_0 \left( \sum_{x \in V_1} u(x) - \alpha - \varepsilon \right) \left( \sum_{x \in V_1} u(x) - \alpha + \varepsilon \right), 0 \right). \]

(32c)
Solving (32a) via the variational method, we get \( u \) fulfilling

\[
\begin{cases}
\lambda_0^k \left( \sum_{x \in V_1} u(x) - \alpha - \varepsilon \right) \left( \sum_{x \in V_1} u(x) - \alpha + \varepsilon \right) + \nabla_w \lambda_1^k + \mu_1 \nabla_w \cdot (v^k - \nabla_w u) = 0, \\
u \in [0, 1]
\end{cases}
\]

(33)

with its discrete iterative formulation

\[
\begin{cases}
\tilde{u}_{k+1}(x) = \frac{2\mu_1 \sum_{y \in N(x)} u^k(y)w(x,y) - \nabla_w \lambda_1^k - \mu_1 \nabla_w \cdot (v^k) + \lambda_0^k T}{2\mu_1 \sum_{x \in N(x)} w(x,y)}, \\
u_{k+1}(x) = \max \left( \min \left( 1, \tilde{u}_{k+1}(x) \right) \right)
\end{cases}
\]

(34)

Where \( T = \left( \sum_{x \in V_1} u(x) - \alpha - \varepsilon \right) \left( \sum_{x \in V_1} u(x) - \alpha + \varepsilon \right). \) \( v^k \) is the same as (21) also.

The algorithm is stated as

**Algorithm 4 DDIC**

\[ u^0 \text{given by random initialization} \]

\[ \alpha^0, \lambda^0 \leftarrow 0 \]

**while** energy not converged **do**

\[ u_{k+1} \text{given by (34)} \]

\[ v_{k+1} \text{given by (21)} \]

\[ \lambda_{1,1}^{k+1} \text{given by (32b)} \]

\[ \lambda_{0,1}^{k+1} \text{given by (32c)} \]

**end while**

3.5. **Ratio Cut (RC).** For the Ratio Cut model (2) without explicit constraint, the minimum problem is

\[
\min_{u \in [0,1]} E(u) = \left\{ \sum_{x \in V} |\nabla_w u|(x) \left( \frac{1}{\sum_{x \in V} \nabla_w u(x)} + \frac{1}{n - \sum_{x \in V} \nabla_w u(x)} \right) \right\}. 
\]

(35)

To solve it easily, we design its alternating optimization scheme by introducing a non-local auxiliary variable \( v = \nabla_w u \) as

\[
(u^{k+1}, v^{k+1}) = \arg \min_{u, v \in [0,1]} E(u, v) = \left\{ \sum_{x \in V} |v|(x) \left( \sum_{x \in V} \frac{1}{\nabla_w u(x)} + \frac{1}{n - \sum_{x \in V} \nabla_w u(x)} \right) + \sum_{x \in V} \lambda_1^k \cdot (v - \nabla_w u) + \frac{\mu_1}{2} \sum_{x \in V} (v - \nabla_w u)^2 \right\},
\]

(36a)

\[
\lambda_1^{k+1} = \lambda_1^k + \mu_1 (v^{k+1} - \nabla_w u^{k+1}).
\]

(36b)

Using the variational method, we get in each loop

\[
\begin{cases}
g |v^k| + \nabla_w \lambda_1^{k+1} + \mu_1 \nabla_w \cdot (v - \nabla_w u) = 0, \\
u \in [0, 1]
\end{cases}
\]

(37)
where, \( g = \frac{1}{n-\sum_{x \in V} \nabla_w u(x)} - \frac{1}{\sum_{x \in V} \nabla_w u(x)} \)

With its discrete iterative formulation

\[
\begin{align*}
\tilde{u}^{k+1}(x) &= \frac{2\mu_1 \sum_{y \in N(x)} u^k(y)w(x,y) - \nabla_w \lambda_1 u^k + \nabla_w \cdot (u^k) + g | v^k|}{2\mu_1 \sum_{x \in N(x)} w(x,y) - \lambda_1 u^k + \nabla_w \cdot (v^k)} , \\
u^{k+1}(x) &= \max \left( \min \left( 1, \tilde{u}^{k+1}(x) \right) \right)
\end{align*}
\]

(38)

and

\[v^{k+1} = \max \left( \left| \nabla_w u^{k+1} - \frac{\lambda_1^k}{\mu_1} \right| - \frac{f}{\mu_1}, 0 \right) \frac{\nabla_w u^{k+1} - \lambda_1^k}{\mu_1}, 0 \|0\| = 0.\]  

(39)

where, \( f = \frac{1}{\sum_{x \in V} \nabla_w u(x)} + \frac{1}{n-\sum_{x \in V} \nabla_w u(x)} \)

The algorithm is summarized as

**Algorithm 5 RC**

\[u^0 \text{ given by random initialization, } v^0, \lambda^0_1 \leftarrow 0\]

\[\text{while energy not converged do}\]

\[u^{k+1} \text{ given by (38)}\]

\[v^{k+1} \text{ given by (39)}\]

\[\lambda^{k+1} \text{ given by (36b)}\]

\[\text{end while}\]

\[3.6. \text{Cheeger Cut (CC).} \text{ For the Cheeger Cut problem, its variation model is}\]

\[
\min_{u \in [0,1]} \frac{\sum_{x \in V} |\nabla_w u(x)|}{\min \left( \sum_{x \in V} u(x), n - \sum_{x \in V} u(x) \right)}.
\]

(40)

It has been shown in [21] that (40) is equivalent to

\[
\min_{u \in [0,1]} E(u) = \frac{\sum_{x \in V} |\nabla_w u(x)|}{\sum_{x \in V} |u(x) - m(u)|}.
\]

(41)

Because \( u \in [0,1] \), we select that the median \( m(u) \) of \( u \) is 0.5. So (41) can be rewritten as

\[
\min_{u \in [0,1]} E(u) = \frac{\sum_{x \in V} |\nabla_w u(x)|}{\sum_{x \in V} |u(x) - 0.5|}.
\]

(42)

which is a minmax problem

\[
\min_{u \in [0,1]} \max_{\lambda_2 \in R} E(u) = \sum_{x \in V} |\nabla_w u(x)| + \lambda_2 \sum_{x \in V} |u - 0.5|.
\]

(43)
Its alternating optimization scheme can be designed as

\[
(u^{k+1}, v^{k+1}, w^{k+1}) = \arg \min_{v, w, u \in [0, 1]} E(u, v, w) = \left\{ \begin{array}{l}
\sum_{x \in V} |v| - \lambda_2 \sum_{x \in V} |w|
+ \sum_{x \in V} \lambda^k_1 \cdot (v - \nabla_w u)
+ \frac{\mu_1}{2} \sum_{x \in V} (v - \nabla_w u)^2
+ \sum_{x \in V} \lambda^k_0 \cdot (w - u + m(u))
+ \frac{\mu_0}{2} \sum_{x \in V} (w - u + m(u))^2
\end{array} \right.,
\]

\[
\lambda^k_1 = \lambda^k_1 + \mu_1 (v^{k+1} - \nabla_w u^{k+1}),
\lambda^k_0 = \lambda^k_0 + \mu_0 (w^k - u^k + m(u)),
\lambda^k_2 = \frac{\|\nabla_w u^{k+1}\|}{|u^{k+1} - m(u)|}.
\]

Using alternatin direction optimization method to (44a), we get the Euler-Lagrangian equation on \(u\)

\[
\nabla_w \lambda^k_1 + \mu_1 \nabla_w \cdot (v^k - \nabla_w u) - \lambda^k_0 - \mu_0 (w^k - u + m(u)) = 0,
\]

and its discrete iterative formulation

\[
\tilde{u}^{k+1}(x) = \frac{2\mu_1 \sum_{y \in N(x)} u^k(y)w(x,y) - \nabla_w \lambda^k_1 - \mu_1 \nabla_w \cdot (v^k - \nabla_w u^k) + \lambda^k_0 + \mu_0 (w^k - u^k + m(u))}{2\mu_1 \sum_{y \in N(x)} w(x,y)},
\]

\[
u^{k+1}(x) = \max (\min (1, \tilde{u}^{k+1}(x)))
\]

\(v^{k+1}\) is the same as (21) also. For \(w^{k+1}\), it can be written as the following formula analytically

\[
\tilde{w}^{k+1} = f_w + \frac{\lambda_2}{\mu_0} f_w,
\]

where, \(f_w = u^{k+1} - m(u) - \frac{\lambda_0}{\mu_0}\). To avoid the trivial solution as stated in introduction, we apply the renormalization step:

\[
\left\{ \begin{array}{l}
\tilde{w}^{k+1} = \frac{w^{k+1} - \min(\tilde{w}^{k+1})}{\max(\tilde{w}^{k+1}) - \min(\tilde{w}^{k+1})}
\end{array} \right.,
\]

This algorithm is summarized as

**Algorithm 6 CC**

\[ u^0 \text{given by random initialization} \]

\[ \text{While out loop not converged do} \]

\[ \lambda^0_1, \lambda^0_0 \leftarrow 0 \]

\[ \text{while energy not converged do} \]

\[ u^{k+1} \text{given by (46)} \]

\[ v^{k+1} \text{given by (21)} \]

\[ w^{k+1} \text{given by (48)} \]

\[ \lambda^k_1 \text{given by (44b)} \]

\[ \lambda^k_0 \text{given by (44c)} \]

\[ \text{end while} \]

\[ \lambda^k_2 \text{given by (44d)} \]

\[ \text{end while} \]
3.7. **Normalized Cut (NC).** Using binary function (7), Normalized Cut (3) can be written as

\[
\min_{u \in [0,1]} E(u) = \left\{ \sum_{x \in V} |\nabla_w u| (x) \left( \frac{1}{\text{vol}(V_1)} + \frac{1}{\text{vol}(V_2)} \right) \right\}.
\]

(49)

Its ALM scheme is

\[
(u^{k+1}, v^{k+1}) = \arg \min_{v,u \in [0,1]} E(u,v) = \left\{ \sum_{x \in V} |v| \left( \frac{1}{\text{vol}(V_1)} + \frac{1}{\text{vol}(V_2)} \right) \right\} + \sum_{x \in V} \lambda_1^k \cdot (v - \nabla_w u) + \frac{\mu_1}{2} \sum_{x \in V} (v - \nabla_w u)^2.
\]

(50)

\[
\lambda_1^{k+1} = \lambda_1^k + \mu (v^{k+1} - \nabla_w u^{k+1}).
\]

(50b)

From (50a), we get the Euler-Lagrangian equation on \(u\) as

\[
\nabla_w \lambda_1^{k+1} + \mu_1 \nabla_w \cdot (v - \nabla_w u) = 0,
\]

(51)

and its discrete iterative formulation

\[
\begin{aligned}
\tilde{u}^{k+1}(x) &= \frac{2\mu_1}{\sum_{y \in N(x)} u^k(y)w(x,y) - \nabla_w \lambda_1^k - \mu_1 \nabla_w (v^k)} \\
u^{k+1}(x) &= \max \left( \min \left( 1, \tilde{u}^{k+1}(x) \right) \right)
\end{aligned}
\]

(52)

along with the soft threshold formula of \(v\)

\[
v = \max \left( \left| \nabla_w u^{k+1} - \frac{\lambda_1^k}{\mu_1} \right| - \frac{g}{\mu_1}, 0 \right),
\]

\[
\frac{\nabla_w u^{k+1} - \frac{\lambda_1^k}{\mu_1}}{\left| \nabla_w u^{k+1} - \frac{\lambda_1^k}{\mu_1} \right|}, 0 \right| = 0,
\]

(53)

where \(g = \left( \frac{1}{\text{vol}(V_1)} + \frac{1}{\text{vol}(V_2)} \right)\). The algorithm is described as

| Algorithm 7 Normalized Cut |
|-----------------------------|
| \( u^0 \) given by random initialization |
| \( \lambda_1^0 \leftarrow 0 \) |
| **while** energy not converged **do** |
| \( u^{k+1} \) given by (52) |
| \( \lambda_1^{k+1} \) given by (50b) |
| **end while** |

4. **Experimental results.** In this section, we use three different data sets to analyze the performance of the seven different algorithms in Sec. 3 until

\[
\left| \frac{E^{k+1} - E^k}{E^k} \right| \leq \eta,
\]

(54)

where \( E^{k+1} \) is the energy functional of the \((k + 1)th\) step, and is the counterpart of the \(k\)th step \(k\), \(\eta\) is a small positive number for convergence. When this condition is fulfilled, the program terminates. The error rate is defined as the ratio between the numbers of incorrect vertices and all vertices on the graph.
All experiments are performed on a 3.3 GHz Inter Core i5 Quad computer. We use 3 datasets: two-moons, handwritten digit set 3&8 and handwritten digit set 4&9. The handwritten digits are generated by the Courant Institute of New York University, and consists of 70000 images of handwritten digits (0-9). The two-moons is a synthetic data set, to test different methods. We construct the unidirectional graph using 9 nearest neighbors and the scaling is based on the 9th closest neighbors, i.e., the k-nearest neighbor method. For the fidelity term, we choose 50 points per class in two-moons and 300 points per class in handwritten digits 3&8 and handwritten digits 4&9. The corresponding ratios are 5%, 4.3% and 4.35% respectively. The penalty parameters used in different algorithm are listed in Table 0.

Table 0 Different algorithm’s parameters

|       | EC-PFM | EC-ALM | SDIC | DDIC | RC  | CC  | NC  |
|-------|--------|--------|------|------|-----|-----|-----|
| $\mu_0$ | 0.01   | 0.005  | $\frac{25}{n}$ | $\frac{25}{n} \cdot n$ | 0.01 | 15  | 15  |
| $\mu_1$ | 15     | 25     | 20   | 15   | 15  | 12  | 15  |

where $n$ is the number of data points and $k$ is the number of classes.

In Fig. 1 (a) and (c), we use red dots to denote the pre-labeled points, but in Fig.1 (b), we use two different colors to label the randomly initialized data. This expression is helpful to get a more intuitive understanding of the classification on an undirected graph. In Fig. 2, we give the reference basis of three data sets for comparisons with the results of experiments.

In Fig. 3, we give the initialization picture and the two-moon data classification results using different methods. But it is difficult to describe the differences of results in this figure because the differences exist mainly near interfaces of classes. So we list their error rates and ranks in Table 1 to compare the accuracies. It shows that the most accurate method is EC-ALM with error rate 1.25%. For the equality
Figure 2. Reference basis of three data sets

Figure 3. Different results for two-moon dataset classification

Table 1 Accuracy comparisons of different algorithms for two-moon dataset classification

| Method | $V_1$.class | $V_2$.class | Error  | Ranking |
|--------|-------------|-------------|--------|---------|
| Solution | 1000       | 1000       |        |         |
| EC-PFM  | 1014       | 986        | 1.7%   | 5       |
| EC-ALM  | 1005       | 995        | 1.25%  | 1       |
| SDIC    | 1016       | 984        | 1.50%  | 2       |
| DDIC    | 1015       | 985        | 1.55%  | 3       |
| RC      | 1030       | 970        | 1.70%  | 5       |
| CC      | 1021       | 979        | 1.65%  | 4       |
| NC      | 986        | 1014       | 1.90%  | 7       |

constraints problems, EC-ALM method is better than EC-PFM method; For the inequality problems, SDIC method is better than DDIC method; For the problems without explicit constraints, CC method is better than RC and NC methods.

In Fig. 4 and Table 2, we present the classification results of handwritten digits 3&8 dataset using different algorithms and their accuracies respectively. It shows that the most accurate method is EC-ALM with error rate 1.1385%. For the equality constraints problems, EC-ALM method is also better than EC-PFM method;
Figure 4. Different results for handwritten digits classification of 3&8 dataset

Table 2 Accuracy comparisons of different algorithms for handwritten digit classification of 3&8 dataset

| Method | $V_1$.class | $V_2$.class | Error  | Ranking |
|--------|-------------|-------------|--------|---------|
| Solution | 7141 | 6825 |            |         |
| EC-PFM | 7165 | 6801 | 1.2745% | 6       |
| EC-ALM | 7139 | 6827 | 1.1385% | 1       |
| SDIC   | 7128 | 6838 | 1.1671% | 3       |
| DDIC   | 7161 | 6805 | 1.2244% | 4       |
| RC     | 7173 | 6793 | 1.2387% | 5       |
| CC     | 7134 | 6832 | 1.1600% | 2       |
| NC     | 7167 | 6799 | 1.5538% | 7       |

Table 3 Accuracy comparisons of different algorithms for handwritten digit classification of 4&9 dataset

| Method | $V_1$.class | $V_2$.class | Error  | Ranking |
|--------|-------------|-------------|--------|---------|
| Solution | 6824 | 6958 |            |         |
| EC-PFM | 6814 | 6968 | 1.2988% | 4       |
| EC-ALM | 6818 | 6964 | 1.2335% | 1       |
| SDIC   | 6799 | 6983 | 1.3133% | 5       |
| DDIC   | 6811 | 6971 | 1.3206% | 6       |
| RC     | 6836 | 6946 | 1.2098% | 3       |
| CC     | 6787 | 6995 | 1.2407% | 2       |
| NC     | 6807 | 6975 | 1.4729% | 7       |

For the inequality problems, SDIC method is better than DDIC method; For the problems without explicit constraints, CC method is better than RC and NC methods.
In Fig. 5 and Table 3, we present the classification results of handwritten digits 4&9 dataset using different algorithms and their accuracies respectively. It shows that the most accurate method is also EC-ALM with error rate 1.2335%. For the equality constraints problems, EC-ALM method is also better than EC-PFM method; For the inequality problems, SDIC method is better than DDIC method; For the problems without explicit constraints, CC method is better than RC and NC methods.

In order to void the dependences on initialization, each of the previous experiments has conducted ten times with different initial conditions randomly, the error rates listed in tables averages of ten experiments. In practice, we have implemented another classification of sub-dataset of handwritten digits, such as 0&1, 7&8, 1&9, the results show the same rule of error rate, we do not give them here due to repetitions.

Obviously, the fidelity set size is an important factor to the error rates of different methods. In order to show the dependency of error rates on fidelity set sizes, we set up two experiments for two-moon classification with 2000 points and 3&8 handwritten digit classification with 13966 digits using EC-ALM method, some results are listed in Table 4. It shows that the error rate decreases gradually with increase of fidelity set size, and the effect on error rate is little when the ratio of fidelity points to total points is less than 5%.

For the problems with double direction constraints, the DDIC method introduces a small positive range parameter, which maybe affect error rates. But numerous experiments show that this effect is very little although the less the the smaller the error rate. Table 5 lists some results in an experiment on handwritten digit 3&8 dataset classification.

5. Concluding remarks. The research focus of this paper is to compare experimentally the accuracies of some balanced data classification methods with explicit equality constraints, inequality constraints and without explicit constraints. In order to perform fairly, we reformulate their models under the discrete variational...
**Table 4** The dependences of error rate on fidelity set size.

| Fidelity set size (per class) | Error rate | Fidelity set size (per class) | Error rate |
|------------------------------|------------|------------------------------|------------|
| 100                          | 1.20%      | 350                          | 1.1287%    |
| 50                           | 1.25%      | 300                          | 1.1385%    |
| 40                           | 1.65%      | 250                          | 1.2888%    |
| 30                           | 1.80%      | 200                          | 1.3604%    |
| 26                           | 1.85%      | 150                          | 1.6397%    |
| 20                           | 2.45%      | 100                          | 3.8665%    |
| 16                           | 2.55%      | 90                           | 4.1386%    |
| 10                           | 3.25%      | 70                           | 9.8597%    |
| 6                            | 4.20%      | 50                           | 11.2201%   |

**Table 5** Dependence of error rate on DDIC range parameter

| ε value | \(V_1\) class | \(V_2\) class | Error rate |
|---------|----------------|----------------|------------|
| 5       | 7153           | 6813           | 1.2316%    |
| 10      | 7160           | 6806           | 1.2387%    |
| 15      | 7121           | 6845           | 1.2459%    |
| 20      | 7159           | 6807           | 1.2602%    |
| 30      | 7160           | 6806           | 1.2576%    |
| 50      | 7161           | 6805           | 1.2724%    |

framework and design corresponding ALM algorithms: EC-PFM (Equality constraint with penalty function method), EC-ALM (Equality constraints with augmented Lagrangian method), SDIC (Single direction inequality constraint), DDIC (Double direction inequality constraint), RC (Ratio Cut), CC (Cheeger Cut), NC (Normalized Cut). Numerous experiments show that it is hard to draw a determined conclusion for their numerical accuracies globally from experimental viewpoints, because their performances may depend on dataset sometimes. But we can draw some intuitive conclusions for further investigations For the problems with explicit equality constraints, EC-ALM method is better than EC-PFM. Additionally, EC-ALM method circumvent the problem of penalty parameter selection. These methods work for the cases of known exact point numbers of sub-dataset.

For the problems with explicit inequality constraints, SDIC shows better accuracy than DDIC. But DDIC method is superior to SDIC when only an approximate number range of points in a specific sub-dataset is known. The CC method is better than RC and NC methods in all experiments. Although these methods do not need explicit constraints, but the algorithms are more difficult than the ones with explicit constraints.

It is worth mentioning that in all experiments the EC-ALM method is the most accurate and the NC method is the most inaccurate one. NC is another version of RC by replacing vertex numbers of sub-datasets with their degrees, but the comparisons are based on the vertex numbers of sub-datasets, this is why RC’s accuracy is better than NC.

Although the conclusions are not drawn theoretically, they are of importance in extensions of the investigated methods to multi-class data classification.
Acknowledgments. This work was supported in part by the National Natural Science Foundation of China Grants 61772294.

REFERENCES

[1] E. Bae and E. Merkurjev, Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds, *Journal of Mathematical Imaging & Vision*, 58 (2017), 468–493.

[2] A. Ben-Dor, G. Lancia and J. Perone et al., Banishing bias from consensus sequences, *Symposium on Combinatorial Pattern Matching*, Springer-Verlag, 1264 (1997), 247–261.

[3] X. Bresson, X.-C. Tai, T. F. Chan and A. Szlam, Multi-class transductive learning based on $L_1$ relaxations of Cheeger cut and Mumford-Shah-Potts model, *Journal of Mathematical Imaging and Vision*, 49 (2014), 191–201.

[4] T. Chan and L. Vese, An active contour model without edges, *International Conference on Scale-Space Theories in Computer Vision*, Springer-Verlag, (1999), 141–151.

[5] J. Cheeger, A lower bound for the smallest eigenvalue of the laplacian, *Problems in Analysis*, (1970), 195–199.

[6] D. L. Donoho, De-noising by soft-thresholding, *IEEE Transactions on Information Theory*, 41 (1995), 613–627.

[7] A. Elmoataz, O. Lezoray and S. Bougleux, Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing, *IEEE Trans. Image Processing*, 17 (2008), 1047–1060.

[8] D. R. Fulkerson, Flows in networks, *Recent Advances in Mathematical Programming*, (1963), 319–331.

[9] G. Gilboa and S. Osher, Nonlocal operators with applications to image processing, *Multiscale Modeling & Simulation*, 7 (2008), 1005–1028.

[10] G. Gilboa and S. Osher, Nonlocal linear image regularization and supervised segmentation, *Multiscale Modeling & Simulation*, 6 (2007), 595–630.

[11] R. Glowinski, T. W. Pan and X. C. Tai, Some facts about operator splitting and alternating direction methods, *Splitting Methods in Communication, Imaging, Science, and Engineering*, 19–94. Sci. Comput., Springer, Cham, 2016.

[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 (1992), 1074–1085.

[13] M. Hein and S. Setzer, Beyond spectral clustering -tight relaxations of balanced graph cuts, *Advances in Neural Information Processing Systems (NIPS)*, (2011).

[14] M. Hein and S. Setzer, Beyond spectral clustering- tight relaxations of balanced graph cuts, In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira and K.Q.Weinberger, Editors, *Advances in Neural Information Processing Systems*, 24 (2011), 2366–2374.

[15] U. V. Luxburg, A tutorial on spectral clustering, *Statistics and Computing*, 17 (2007), 395–416.

[16] M. Meila & J. Shi. Learning segmentation by random walks. *Advances in Neural Information Processing Systems*, 13 (2000), 873- 879.

[17] E. Merkurjev, A. L. Bertozzi, X. Yan and K. Lerman, Modified cheeger and ratio cut methods using the ginzburg-landau functional for classification of high-dimensional data, *Inverse Problems*, 33 (2017), 074003, 24pp.

[18] E. Merkurjev, E. Bae, A. L. Bertozzi and X. C. Tai, Global binary optimization on graphs for classification of high-dimensional data, *Journal of Mathematical Imaging and Vision*, 52 (2015), 414–435.

[19] L. I. Rudin, S. Osher and W. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D Nonlinear Phenomena*, 60 (1992), 259–268.

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

[21] A. Szlam and X. Bresson, A total variation-based graph clustering algorithm for Cheeger ratio cuts, *Conference on Machine Learning*, (2010), 1039–1046.

[22] A. N. Tikhonov, Regularization of incorrectly posed problems, *Sov Math.*, 4 (1963), 1624–1627.

[23] D. Wagner and F. Wagner, Between min cut and graph bisection, *Mathematical Foundations of Computer Science 1993* (Gdask, 1993), 711 (1993), 744–750.
[24] Y. Z. Zhang, Y. Jiang and Z. F. Pang, Cheeger cut model for the balanced data classification problem, *Advanced Materials Research*, (2013), 765–767, 730–734.

[25] D. Zhou and B. Schlkopf, Regularization on discrete spaces, *Joint Pattern Recognition Symposium*, 3663 (2005), 361–368.

[26] X. Zhu and A. B. Goldberg, Introduction to semi-supervised learning, *Morgan & Claypool Publishers*, (2009), 130pp.

E-mail address: 13697577910@163.com
E-mail address: 1211368707@qq.com
E-mail address: hyang38e.ntu.edu.sg
E-mail address: 673167772@qq.com
E-mail address: zkpan0126.com