Continuous max-flow and global minimization for high dimensional data

Xue-Cheng Tai,
University of Bergen, Norway

Collaborations with:
E. Bae, A. Bertozzi, E. Merkurjev
Industrial project leader:
Junyong You (CMR, Christian Michelson Research)

June 15, 2015

Table of Contents

Interface (classification) problems

Interface problems exists everywhere in science and technology. For imaging and vision, it is somehow classical:

- Mumford-Shal model
- GAC model
- Chan-Vese model

How to solve these interface problems?

Interface (classification) problems

Interface problems exists everywhere in science and technology. For imaging and vision, it is somehow classical:

- Mumford-Shal model
- GAC model
- Chan-Vese model

How to solve these interface problems?

- active count our
- level set
- phase-field
- ...

Interface (classification) problems

Interface problems exists everywhere in science and technology. For imaging and vision, it is somehow classical:

- Mumford-Shal model
- GAC model
- Chan-Vese model

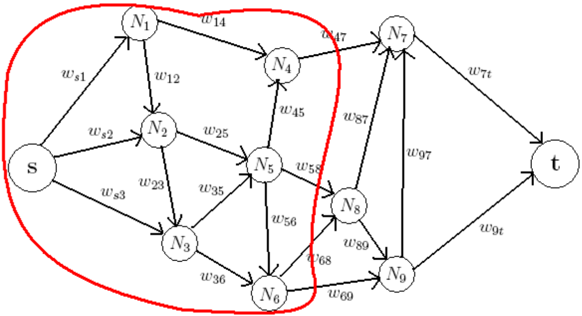
How to solve these interface problems?

- active count our
- level set
- phase-field
- ...

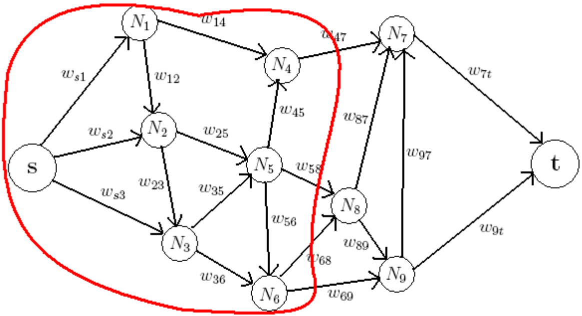
Traditional methods are:

- Nonlinear
- Non-convex
- ...

Max-Flow / Min-Cut

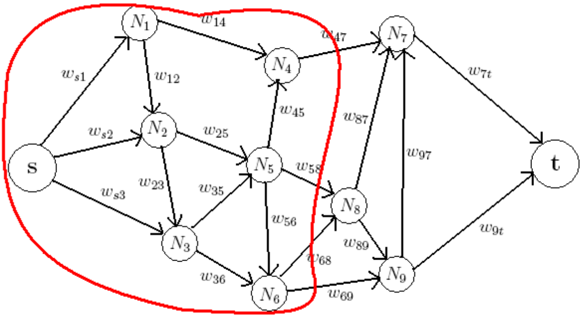


Max-Flow / Min-Cut



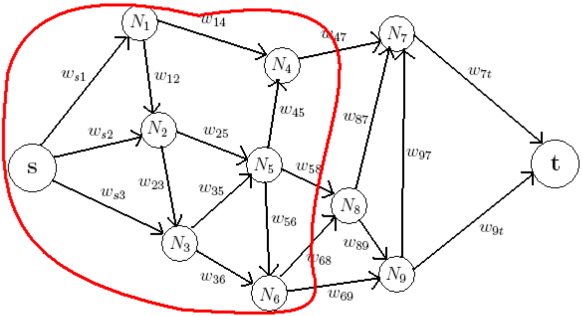
(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$

Max-Flow / Min-Cut



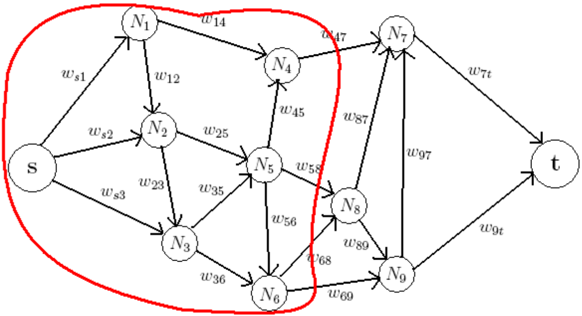
(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,

Max-Flow / Min-Cut



(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,
Max-Flow: Find the maximum amount of flow from s to t.

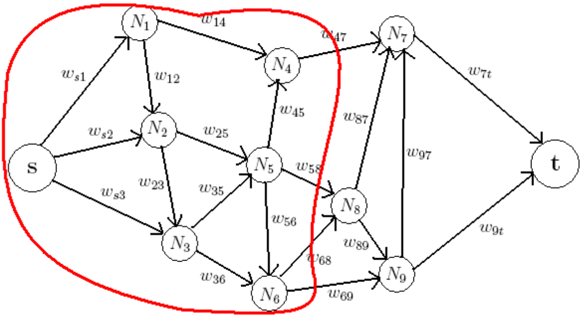
Max-Flow / Min-Cut



(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,
Max-Flow: Find the maximum amount of flow from s to t.

Max-flow = min-cut.

Max-Flow / Min-Cut



(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,
Max-Flow: Find the maximum amount of flow from s to t.

Max-flow = min-cut.

Higher dimensional problems

A graph for 2D images:

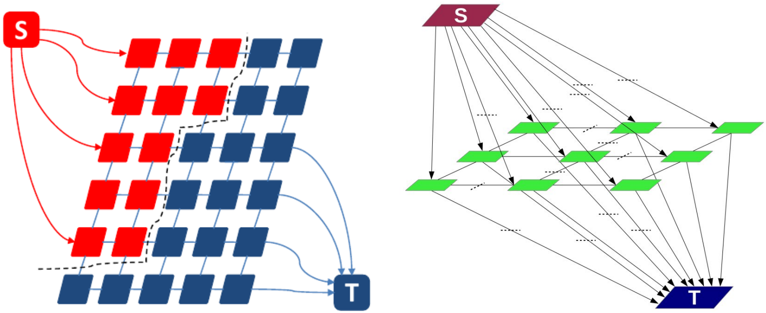


Figure: Graph used for discrete 2D binary labeling

Two-phase Min-cut – Discretized setting

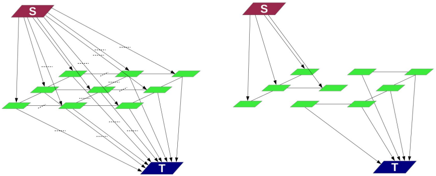


Figure: Graph and cut for discrete binary labeling

It is easy to see the cost of a cut ($u(p) = 0$ or 1). A minimum cut is to find u for:

$$\min_{u \in \{0,1\}} \sum_{p \in \mathcal{P}} f_1(p)(1-u(p)) + f_2(p)u(p) + \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p^k} g(p,q)|u(p)-u(q)|.$$

Capacity:

$$w_{s,p} = f_1(p),\ w_{t,p} = f_2(p),\ w_{p,q} = g(p,q).$$

Ref: $\mathcal{N}_p^k$ is the k-neighborhood of $p \in \mathcal{P}$.

Two-phase Min-cut – corresponding continuous setting

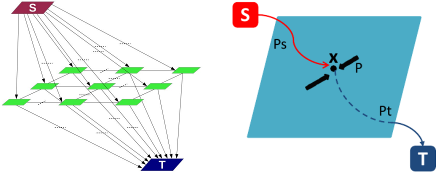


Figure: Graph used for discrete and continuous binary labeling

A "continuous" minimum cut is to solve:

$$\min_{u \in \{0,1\}} \int_{\Omega} f_1(x)(1-u(x)) + f_2(x)u(x) + g_1(x)|D_1 u(x)| + g_2(x)|D_2 u(x)|.$$

Capacity:

$$w_s(x) = f_1(x),\ \ w_t(x) = f_2(x),\ \ w_1(x) = g_1(x), \quad w_2(x) = g_2(x).$$

Max-Flow over a graph

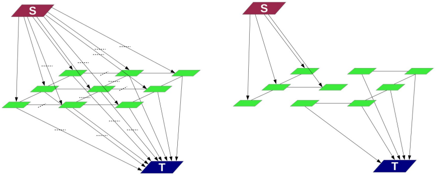


Figure: Graph used for discrete binary labeling

Max-flow formulation

$$\max_{p_s, p_t, q} \sum_{v \in \mathcal{V} \setminus \{s,t\}} p_s(v)$$

subject to

$$\begin{aligned} |q(v,u)| &\leq g(v,u), \quad \forall (v,u) \in \mathcal{V} \times \mathcal{V} \\ 0 \leq p_s(v) &\leq f_1(v), \quad \forall v \in \mathcal{V} \setminus \{s,t\}; \\ 0 \leq p_t(v) &\leq f_2(v), \quad \forall v \in \mathcal{V} \setminus \{s,t\}; \\ \Big(\sum_{u \in N(v)} q(v,u) \Big) - p_s(v) + p_t(v) &= 0, \quad \forall v \in \mathcal{V} \setminus \{s,t\}; . \end{aligned}$$

Continuous Max-Flow

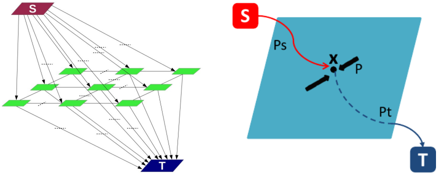


Figure: Discrete (left) vs. Continuous (right)

Continuous max-flow formulation

$$\sup_{p_s, p_t, q} \int_\Omega p_s(x)\, dx$$

subject to
$$\begin{aligned}
|q_1(x)| \leq g_1(x);\ |q_2(x)| &\leq g_2(x), & \forall x \in \Omega; \\
0 \leq p_s(x) &\leq f_1(x), & \forall x \in \Omega; \\
0 \leq p_t(x) &\leq f_2(x), & \forall x \in \Omega; \\
\operatorname{div} q(x) - p_s(x) + p_t(x) &= 0, & \text{a.e. } x \in \Omega.
\end{aligned}$$

Related: (G. Strang (1983)).

Continuous Max-Flow: different internal flow capacity

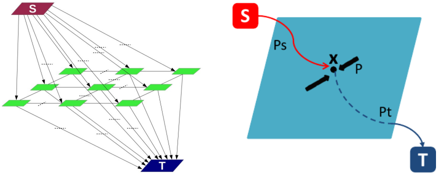


Figure: Discrete (left) vs. Continuous (right)

Continuous max-flow formulation

$$\sup_{p_s, p_t, q} \int_\Omega p_s(x)\, dx$$

subject to

$$|q(x)| = \sqrt{q_1^2(x) + q_2^2(x)} \leq g(x), \quad \forall x \in \Omega;$$

$$0 \leq p_s(x) \leq f_1(x), \quad \forall x \in \Omega;$$

$$0 \leq p_t(x) \leq f_2(x), \quad \forall x \in \Omega;$$

$$\operatorname{div} q(x) - p_s(x) + p_t(x) = 0, \quad \text{a.e. } x \in \Omega.$$

Connection: Continuous Max-Flow and Min-Cut

Lagrange multiplier u for flow conservation condition

$$\operatorname{div} q(x) - p_s(x) + p_t(x) = 0, \quad \text{a.e. } x \in \Omega.$$

yields primal-dual formulation

$$\sup_{p_s, p_t, q} \inf_u \int_\Omega p_s + u\big(\operatorname{div} q - p_s + p_t\big)\, dx$$

$$\text{s.t.} \quad p_s(x) \le f_1(x), \quad p_t(x) \le f_2(x), \quad |q(x)| \le g(x).$$

Optimizing for flows p_s, p_t, q results in:

$$\min_{u \in [0,1]} \int_\Omega f_1(x)(1 - u(x)) + f_2(x) u(x)\, dx + g(x)\, |\nabla u(x)|\, dx.$$

This is exactly the same model as the model in CEN (2006). [1]

[1] T. F. Chan and S. Esedoglu and M. Nikolova: Algorithms for finding global minimizers of image segmentation and denoising models, SIAM J. Appl. Math., 66, 1632–1648,(2006)

Three problems

PCLMS or Binary LM (Lie-Lysaker-T.,2005):

$$\min_{u(x)\in\{0,1\}} \int_\Omega f_1(1-u) + f_2 u + g(x)|\nabla u| dx.$$

Convex problem (CEN, (Chan-Esdoglu-Nikolova,2006))

$$\min_{u(x)\in[0,1]} \int_\Omega f_1(1-u) + f_2 u + g(x)|\nabla u| dx.$$

Graph-cut (Boykov-Kolmogorov,2001)

$$\max_{p_s, p_t, q} \int_\Omega p_s dx \text{ subject to:}$$

$$p_s(x) \le f_1(x),\ \ p_t(x) \le f_2(x),\ \ |p(x)| \le g(x),$$

$$\mathrm{div} p(x) - p_s(x) + p_t(x) = 0.$$

Continuous Max-Flow and Min-Cut

Multiplier-Based Maximal-Flow Algorithm

Augmented lagrangian functional (Glowinski & Le Tallec, 1989)

$$L_c(p_s, p_t, q, \lambda) := \int_\Omega p_s\, dx + \lambda\big(\operatorname{div} q - p_s + p_t\big) - \frac{c}{2}\left|\operatorname{div} q - p_s + p_t\right|^2 dx.$$

minmax subject to:

$p_s(x) \le f_1(x)\,, \quad p_t(x) \le f_2(x)\,, \quad |q(x)| \le g(x)$

ADMM algorithm: For k=1,... until convergence, solve

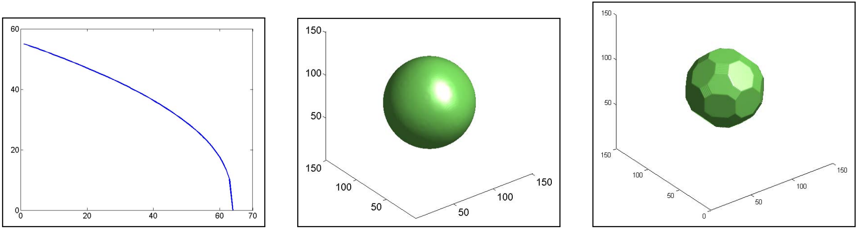
$$q^{k+1} := \arg \max_{\|q\|_\infty \le \alpha} L_c(p_s^k, p_t^k, q, \lambda^k)$$

$$p_s^{k+1} := \arg \max_{p_s(x) \le f_1(x)} L_c(p_s, p_t^k, q^{k+1}, \lambda^k)$$

$$p_t^{k+1} := \arg \max_{p_t(x) \le f_2(x)} L_c(p_s^{k+1}, p_t, q^{k+1}, \lambda^k)$$

$$\lambda^{k+1} = \lambda^k - c\left(\operatorname{div} q^{k+1} - p_s^{k+1} + p_t^{k+1}\right)$$

Metrication error, Parallel, GPU, ...



Experiment of mean-curvature driven 3D surface evolution (volume size: 150X150X150 voxels). (a) The radius plot of the 3D ball evolution driven by its mean-curvature flow, which is computed by the proposed continuous max-flow algorithm; its function is theoretically $r(t) = \sqrt{C - 2t}$. (b) The computed 3D ball at one discrete time frame, which fits a perfect 3D ball shape. This is in contrast to (c), the computation result by graph cut [15] with a 3D 26-connected graph. The computation time of the continuous max-flow algorithm for each discrete time evolution is around 1 sec., which is faster than the graph cut method (120 sec.)

Ref: Y. Yuan, E. Ukwatta, X. Tai, A. Fenster, and C. Schnorr. A fast global optimization-based approach to evolving contours with generic shape prior. Technical report, also UCLA Tech. Report CAM 12-38, 2012.

Metrication error, Parallel, GPU, ...

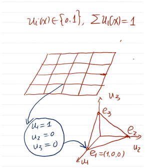
- Fully parallel, easy GPU implementation.
- linear grow of computational cost (per iteration): 2D, 3D, ...

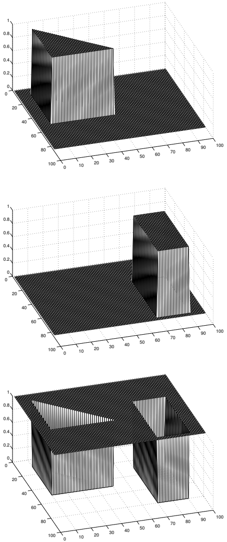
Multiphase Approaches

Multiphase problems – Approach I

We need to identify n characteristic functions $\psi_i(x),\ i = 1, 2 \cdots n$:

$$\psi_i(x) \in \{0,1\}, \quad \sum_{i=1}^{n} \psi_i(x) = 1.$$





Ref: Zach-et-al (2008), Lelmman-et-al(2009,2010, 2013), Bae-Yuan-T. (IJCV 2009), Bae-et-al (SSVM2009, 2012), Yuan-et-al (ECCV2010, CVPR2010)

Multiphase problems – Approach II

Each point $x \in \Omega$ is labelled by a vector function:

$$u(x) = (u_1(x), u_2(x), \cdots u_d(x)).$$

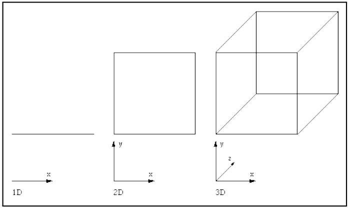
Ref: Vese-Chan (2002), Bae-T. (JMIV2013), Lie-et-al (IEEE TIP 2006), Bae-et-al(2012,2013), Liu-T.-Leung(EMMCVPR2013), Nieuwenhuis-Toppe-Cremers (IJCV2013).

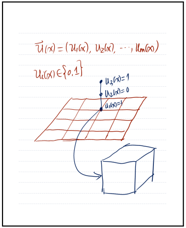
Multiphase problems – Approach II

Each point $x \in \Omega$ is labelled by a vector function:

$$u(x) = (u_1(x), u_2(x), \cdots u_d(x)).$$

- Multiphase: Total number of phases $n = 2^d$. (Chan-Vese)





$u_i(x) \in \{0, 1\}$. Total phase $n = 2^m$.

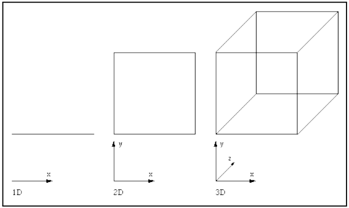
Ref: Vese-Chan (2002), Bae-T. (JMIV2013), Lie-et-al (IEEE TIP 2006), Bae-et-al(2012,2013), Liu-T.-Leung(EMMCVPR2013), Nieuwenhuis-Toppe-Cremers (IJCV2013).

Multiphase problems – Approach II

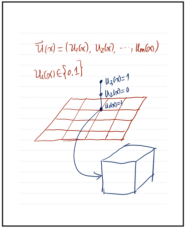
Each point $x \in \Omega$ is labelled by a vector function:

$$u(x) = (u_1(x), u_2(x), \cdots u_d(x)).$$

- Multiphase: Total number of phases $n = 2^d$. (Chan-Vese)



$u_i(x) \in \{0, 1\}$.



Total phase $n = 2^m$.

- More than binary labels: Total number of phases $n = B^d$.

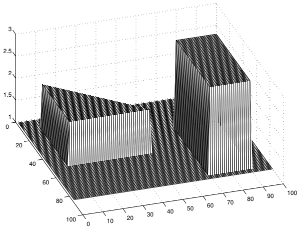
$$u_i(x) \in \{0, 1, 2, \cdots B\}.$$

Ref: Vese-Chan (2002), Bae-T. (JMIV2013), Lie-et-al (IEEE TIP 2006), Bae-et-al(2012,2013), Liu-T.-Leung(EMMCVPR2013), Nieuwenhuis-Toppe-Cremers (IJCV2013).

Multiphase problems – Approach III

Each point $x \in \Omega$ is labelled by

$$u(x) = i, \quad i = 1, 2, \cdots n.$$

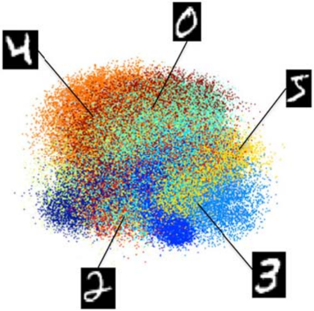


- One label function is enough for any n phases.
- More generall $u(x) = \ell_i, \; i = 1, 2, \cdots n.$

Ref: Lie-Lysaker-T. (2006), Ishikawa(2003), Darbon-Sigelle (2006), Pock-et-al(2008), Bae-T. (SSVM2009), Brown-Chan-Bresson(2011), Bae-Yuan-T.-Boykov (2013,2014).

Application to machine learning

We would like to classify a high dimensional data into multi-classes.



The model

- Graph: $G = (V, E)$, each data is one vertices in V, E are the connections.
- the weight on E:

$$w(x, y) = e^{-\frac{d(x,y)^2}{\sigma^2}}, \tag{1}$$

$d(x, y)$: distance measure.
Another choice:

$$w(x, y) = e^{-\frac{d(x,y)^2}{\sqrt{\tau(x)\tau(y)}}}, \tag{2}$$

$\tau(x) = d(x, z)^2$, and z is the M^{th} closest vertex to vertex x.

Nonlocal TV on a graph

Gradient operator $\nabla : \mathcal{V} \to \mathcal{E}$ is:

$$(\nabla\lambda)_w(x, y) = w(x, y)^{1-q}(\lambda(y) - \lambda(x)). \tag{3}$$

Divergence *div* : $\mathcal{E} \to \mathcal{V}$ is:

$$(\mathrm{div}_w\, \phi)(x) = \frac{1}{2d(x)^r} \sum_y w(x, y)^q(\phi(x, y) - \phi(y, x)), \tag{4}$$

It is true:

$$\langle \nabla u, \phi \rangle_{\mathcal{E}} = -\langle u, \mathrm{div}_w\, \phi \rangle_{\mathcal{V}}.$$

Partition problem on a graph

We are interested in solving partition problems of the form

$$\min_{S \subset V} \sum_{(x,y) \in E \,:\, x \in V,\ y \in V \setminus S} w(x, y) \tag{5}$$

Partition problem on a graph

The problem can be expressed as

$$\min_{\lambda \in \{0,1\}} E^P(\lambda) = TV_w(\lambda) + \sum_{x \in V} f(\lambda(x), x), \tag{6}$$

where

$$TV_w(\lambda) = \frac{1}{2} \sum_{x.y} w(x,y)^q |\lambda(x) - \lambda(y)|$$

$$f(\lambda(x), x) = \eta(x) |\lambda(x) - \lambda^0(x)|^2, \tag{7}$$

where λ^0 is a binary function taking value 1 or 0 at some vertices with known classification. $\eta = 0$ on unclassified vertices.

Partition problem on a graph

Define:

$$C_s(x) = f(0, x), \quad C_t(x) = f(1, x), \quad \forall x \in V,$$
$$g(\phi(x), x) = C_t(x)\phi(x) + C_s(x)(1 - \phi(x)), \ \forall x \in V. \tag{8}$$

The problem

$$\min_{\lambda \in \{0,1\}} E^P(\lambda) = TV_w(\lambda) + \sum_{x \in V} g(\lambda(x), x) \tag{9}$$

is equivalent to:

$$\min_{\lambda \in [0,1]} E^P(\lambda) = TV_w(\lambda) + \sum_{x \in V} g(\lambda(x), x), \tag{10}$$

Partition problem on a graph

Algorithm 1 Max-flow Algorithm

Initialize p_s^1, p_t^1, p^1 and λ^1. For $k = 1, \dots$ until convergence:

- Optimize p flow

$$p^{k+1} = \underset{|p(e)| \leq W(e)\ \forall e \in E}{\arg\max} -\frac{c}{2} \left\| \operatorname{div}_w p - F^k \right\|_2^2,$$

- Optimize source flow p_s

$$p_s^{k+1} = \underset{p_s(x) \leq C_s(x)\ \forall x \in V}{\arg\max} \sum_{x \in V} p_s - \frac{c}{2} \left\| p_s - G^k \right\|_2^2,$$

- Optimize sink flow p_t

$$p_t^{k+1} = \underset{p_t(x) \leq C_t(x)\ \forall x \in V}{\arg\max} -\frac{c}{2} \left\| p_t - H^k \right\|_2^2,$$

- Update λ

$$\lambda^{k+1} = \lambda^k - c \left(\operatorname{div}_w p^{k+1} - p_s^{k+1} + p_t^{k+1} \right).$$

MNIST: test

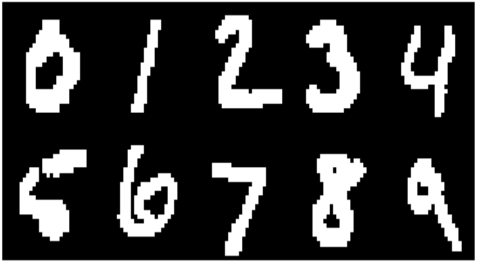


Figure: Examples of digits from the MNIST data base

Using random initialization and random fidelity, the max-flow method obtained an accuracy of around 98.48% averaged over 100 runs with different fidelity sets of 500 randomly chosen points (or only 3.62% of the set).

Banknote Authentication Data Set

The banknote authentication data set, from the UCI machine learning repository: http://archive.ics.uci.edu/ml/, is a data set of 1372 features extracted from images (400×400 pixels) of genuine and forged banknotes. Wavelet transform was used to extract the features from the images. The goal is to segment the banknotes into being either genuine or forged.

With the max-flow method, for a 5.1% fidelity set, we were able to obtain an average accuracy (over 100 different fidelity sets) of around 99.09%.

Two moons

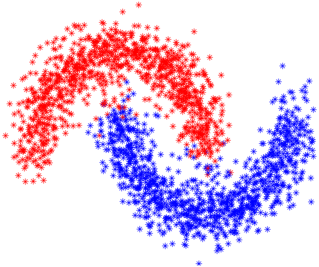


Figure: Two moons example with max-flow method

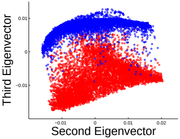
This data set is constructed from two half circles in $\mathbb{R}^2$ with a radius of one. The centers of the two half circles are at $(0, 0)$ and $(1, 0.5)$. A thousand uniformly chosen points are sampled from each circle, embedded in $\mathbb{R}^{100}$ and i.d.d. Gaussian noise with standard deviation 0.02 is added to each coordinate. Therefore, the set consists of two thousand points. Starting from some initial classification of the points, the goal is to segment the two half circles. For the max-flow method, in the case of 65 or lower number of fidelity points (3.25 %), we increased the number of edges of supervised points to others to avoid the trivial global minimizer where all points but the supervised ones are classified as one class.
Using random initialization and random fidelity, for the max-flow method, we obtained an average accuracy (over 100 different fidelity sets) of 97.10% and 97.05% in the case of 100 and 50 fidelity points, respectively.

Comparison of our convex algorithms to binary MBO and GL methods

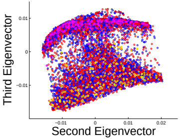
Table: Comparison of methods

	max-flow	primal augmented Lagrangian	binary MBO	binary GL
MNIST (3.6% fidelity) random initialization, random fidelity	98.48%	98.44%	98.37%	98.29%
MNIST (3.6% fidelity) 2nd eigenvector initialization, random fidelity	98.48%	98.43%	98.36%	98.25%
MNIST (3.6% fidelity) random initialization, corner fidelity	98.47%	98.40%	62.35%	64.39%
MNIST (3.6% fidelity) 2nd eigenvector initialization, corner fidelity	98.46%	98.40%	63.87%	63.19%
Banknote Data Set (5.1% fidelity)	99.09%	98.75%	95.43%	97.76%
Banknote Data Set (3.6% fidelity)	98.83%	98.29%	93.48%	96.10%
two moons (5% fidelity)	97.10%	97.07%	98.41%	98.31%
two moons (2.5% fidelity)	97.05%	96.78%	97.53%	98.15%

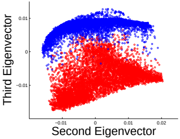
Test with random initialization: MNIST



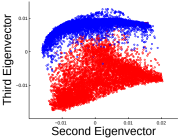
Ground Truth



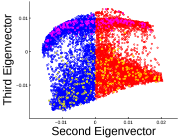
Random initialization and fidelity



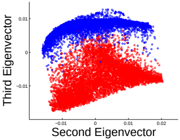
max-flow result



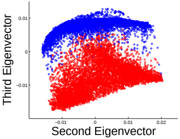
binary MBO result



2nd eigenvector initialization, random fidelity

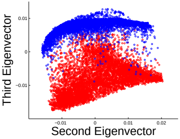


max-flow result

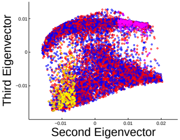


binary MBO result

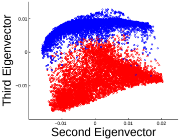
Test with random initialization: MNIST



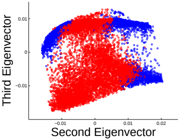
Ground Truth



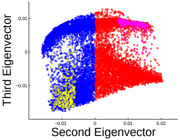
Random initial., corner fidelity



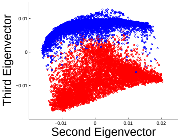
max-flow result



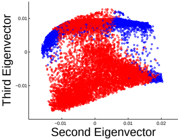
binary MBO result



2nd eigenvector initialization, corner

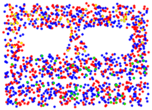


max-flow result

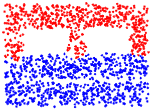


binary MBO result

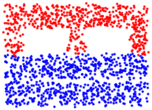
Tests: Rod



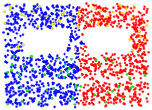
Random initialization, random fidelity



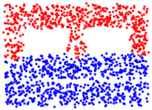
max-flow result



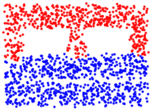
binary MBO result



2nd eigenvector initialization

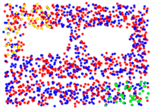


max-flow result

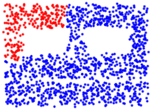


binary MBO result

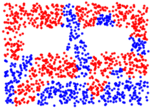
Tests: Rod



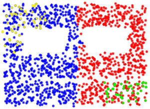
Random initial., corner fidelity



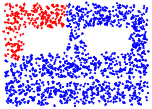
max-flow result



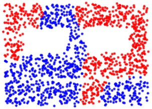
binary MBO result



2nd eigenvector initialization



max-flow result



binary MBO result

Figure: Results for Rod 1. Left: initialization, supervised points are marked in yellow and green. Middle: max-flow algorithm result. Right: binary MBO result

Number of Iterations and Timing

Table: Number of Iterations and Timing

Number of iterations	max-flow	primal augmented Lagrangian	binary MBO	binary GL
MNIST	426	2709	10	52
Banknote Authentication Data Set	314	725	7	449
two moons	1031	451	8	108
Timing (s)	**max-flow**	**primal augmented Lagrangian**	**binary MBO**	**binary GL**
MNIST [a]	2.88	43.21	0.52	0.78
Banknote Authentication Data Set	1.21	3.76	0.90	0.95
two moons	4.13	5.23	2.30	2.98

[a] This is the timing of the method using already computed weights and eigenvalues/eigenvectors of the random walk Laplacian.

Comparison of Final Energy

Table: Comparison of Final Energy

Data Set	initial energy	max-flow final energy	primal augmented Lagrangian final energy	binary MBO final energy	binary GL final energy
MNIST (random fid)	23223	789	789	798	804
MNIST (non-random fid)	23223	791	792	2167	5363
Banknote Authentication	3308	30	37	51	42
two moons	3802	533	535	538	548
rod 1 (random fid)	4159	146	148	163	159
rod 1 (non-random fid)	4159	88	89	825	391
rod 2 (random fid)	4528	171	176	186	184
rod 2 (non-random fid)	4528	101	105	709	421

Cheeger ratio cut

Two-phase:

$$\min_{\Omega \subset V} \frac{\mathrm{cut}(\Omega, \Omega^c)}{\min(\Omega, \Omega^c)}$$

Ref: Bresson, X., Tai, X.-C. C., Chan, T. F. and Szlam, A. (2013). Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. Journal of Mathematical Imaging and Vision, 49(1), 111.

Cheeger ratio cut

Two-phase:

$$\min_{\Omega \subset V} \frac{\text{cut}(\Omega, \Omega^c)}{\min(\Omega, \Omega^c)}$$

Equivalent to:

$$\min_{u \in [0,1]} \frac{TV_w(u)}{\|u\|_1} \text{ s.t. } m(u) := median(u) = 0.$$

Ref: Bresson, X., Tai, X.-C. C., Chan, T. F. and Szlam, A. (2013). Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. Journal of Mathematical Imaging and Vision, 49(1), 111.

Cheeger ratio cut

Two-phase:

$$\min_{\Omega \subset V} \frac{\text{cut}(\Omega, \Omega^c)}{\min(\Omega, \Omega^c)}$$

Equivalent to:

$$\min_{u \in [0,1]} \frac{TV_w(u)}{\|u\|_1} \text{ s.t. } m(u) := median(u) = 0.$$

Equivalent to:

$$\min_{u \in [0,1]} \max_{\lambda \in R} TV_w(u) - \lambda \|u\|_1, \qquad \text{s.t. } m(u) = 0.$$

Algorithm: primal-Dual.

Ref: Bresson, X., Tai, X.-C. C., Chan, T. F. and Szlam, A. (2013). Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. Journal of Mathematical Imaging and Vision, 49(1), 111.

Cheeger ratio cut

Two-phase:

$$\min_{\Omega \subset V} \frac{\mathrm{cut}(\Omega, \Omega^c)}{\min(\Omega, \Omega^c)}$$

Equivalent to:

$$\min_{u \in [0,1]} \frac{TV_w(u)}{\|u\|_1} \text{ s.t. } m(u) := median(u) = 0.$$

Equivalent to:

$$\min_{u \in [0,1]} \max_{\lambda \in R} TV_w(u) - \lambda \|u\|_1, \quad \text{s.t. } m(u) = 0.$$

Algorithm: primal-Dual.
Global convex optimization for u.

Ref: Bresson, X., Tai, X.-C. C., Chan, T. F. and Szlam, A. (2013). Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. Journal of Mathematical Imaging and Vision, 49(1), 111.

Multpiphase Cheeger ratio cut

Multiphasep:

$$\min_{\Omega_k \subset V, k=1,2,\cdots K} \sum_k \frac{\mathrm{cut}(\Omega_k, \Omega_k^c)}{\min(\Omega_k, \Omega_k^c)}$$

Multpiphase Cheeger ratio cut

Multiphasep:

$$\min_{\Omega_k \subset V, k=1,2,\cdots K} \sum_k \frac{\text{cut}(\Omega_k, \Omega_k^c)}{\min(\Omega_k, \Omega_k^c)}$$

Equivalent to:

$$\min_{u_k \in [0,1]} \sum_k \frac{TV_w(u_k)}{\|u_k - m(u_k)\|_1} \text{ s.t. } \sum u_k = 1.$$

Multpiphase Cheeger ratio cut

Multiphasep:

$$\min_{\Omega_k \subset V, k=1,2,\cdots K} \sum_k \frac{\mathrm{cut}(\Omega_k, \Omega_k^c)}{\min(\Omega_k, \Omega_k^c)}$$

Equivalent to:

$$\min_{u_k \in [0,1]} \sum_k \frac{TV_w(u_k)}{\|u_k - m(u_k)\|_1} \text{ s.t. } \sum u_k = 1.$$

Equivalent to:

$$\min_{u_k \in [0,1]} \max_{\lambda_k \in R} \sum_k TV_w(u_k) - \lambda_k \|u_k\|_1, \quad \text{ s.t. } m(u_k) = 0, \quad \sum_k u_k = 1.$$

Algorithm: primal-Dual.

Multpiphase Cheeger ratio cut

Multiphasep:

$$\min_{\Omega_k \subset V, k=1,2,\cdots K} \sum_k \frac{\mathrm{cut}(\Omega_k, \Omega_k^c)}{\min(\Omega_k, \Omega_k^c)}$$

Equivalent to:

$$\min_{u_k \in [0,1]} \sum_k \frac{TV_w(u_k)}{\|u_k - m(u_k)\|_1} \text{ s.t. } \sum u_k = 1.$$

Equivalent to:

$$\min_{u_k \in [0,1]} \max_{\lambda_k \in R} \sum_k TV_w(u_k) - \lambda_k \|u_k\|_1, \quad \text{s.t. } m(u_k) = 0, \quad \sum_k u_k = 1.$$

Algorithm: primal-Dual.
Global convex optimization for u_k.

Test for 4-moon

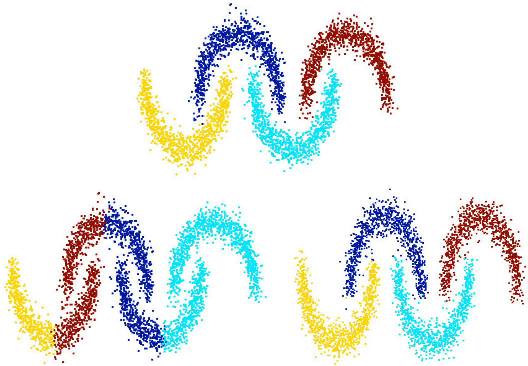
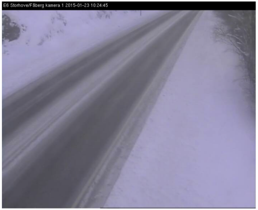


Figure: (a) True solution, (b) Shi-Malik (c) Our algorithm.

Road condition from online cameras

(f) Classification: FullSnowy
Probability: 28.11% (Dry), 47.52% (FullSnowy), 8.16% (FullSnowyIcy), 2.98% (PartialSnowyIcy), 13.23% (Wet)

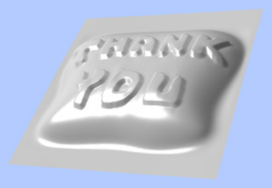


(g) Classification: FullSnowyIcy
Probability: 9.98% (Dry), 22.08% (FullSnowy), 26.87% (FullSnowyIcy), 24.57% (PartialSnowyIcy), 16.51% (Wet)

(h) Classification: Dry
Probability: 79.06% (Dry), 3.02% (FullSnowy), 9.36% (FullSnowyIcy), 4.80% (PartialSnowyIcy), 3.76% (Wet)

(i) Classification: Wet
Probability: 4.97% (Dry), 0.05% (FullSnowy), 0.04% (FullSnowyIcy), 0.76% (PartialSnowyIcy), 94.17% (Wet)

Thank you!



References:

- E. Merkurjev, E. Bae, A. L. Bertozzi, and X.-C. Tai. Global binary optimization on graphs for classification of high dimensional data. J. Math. Imag. and Vis., 2014.
- Bresson, X., Tai, X.-C. C., Chan, T. F. and Szlam, A. (2013). Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. Journal of Mathematical Imaging and Vision, 49(1), 111. http://doi.org/10.1007/s10851-013-0452-5
- Yuan, J., Bae, E. and Tai, X.-C. (2010). A study on continuous max-flow and min-cut approaches. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1), 22172224.
- Yuan, J., Bae, E., Tai, X. and Boykov, Y. (2010). A Continuous Max-Flow Approach to Potts Model, 379392, ECCV2010.
- Yuan, J., Bae, E., Tai, X.-C. C. and Boykov, Y. (2013). A spatially continuous max-flow and min-cut framework for binary labeling problems. Numerische Mathematik, 126(3), 559587. http://doi.org/10.1007/s00211-013-0569-x