Gene Cheung
National Institute of Informatics
27th November, 2017

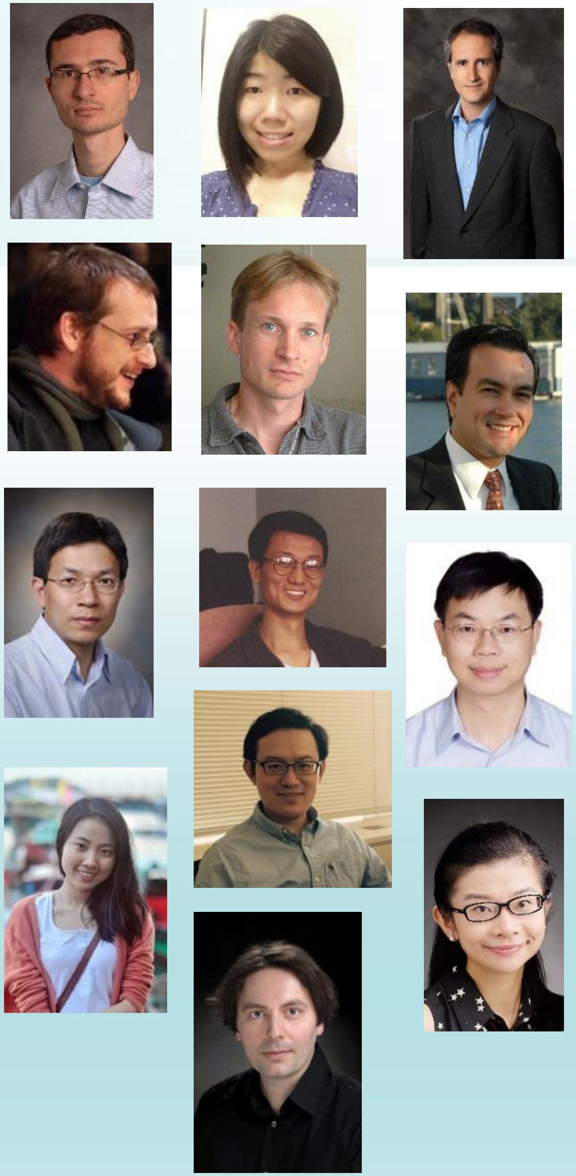
Semi-Supervised Graph Classifier Learning with Negative Edge Weights

.

Acknowledgement

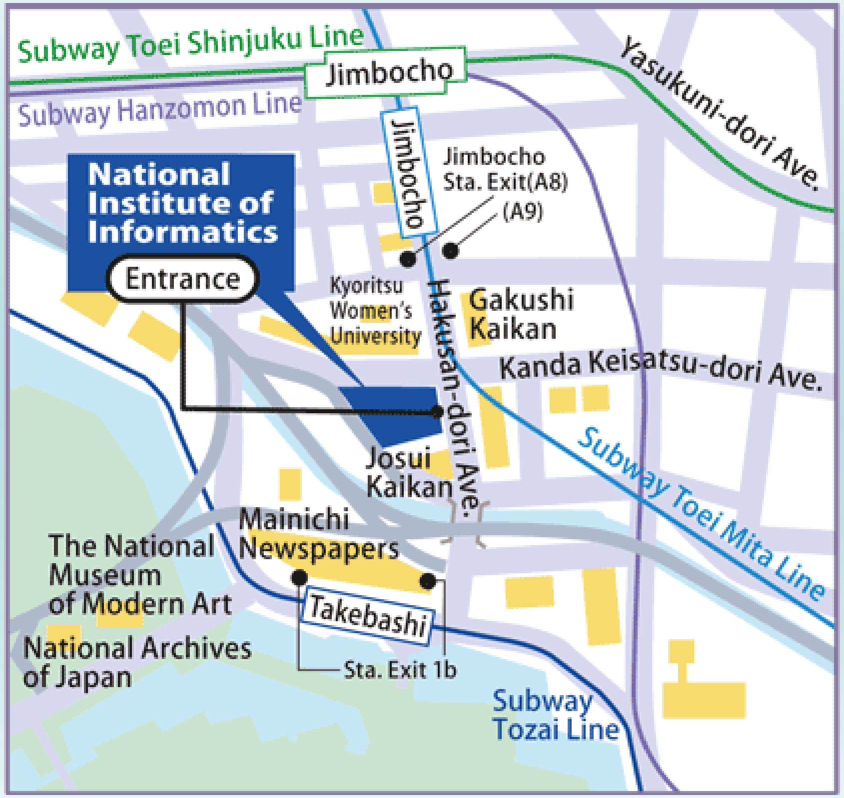
Collaborators:

- M. Kaneko (NII, Japan)
- A. Ortega (USC, USA)
- D. Florencio (MSR, USA)
- P. Frossard (EPFL, Switzerland)
- J. Liang, I. Bajic (SFU, Canada)
- V. Stankovic (U of Strathclyde, UK)
- X. Wu (McMaster U, Canada)
- P. Le Callet (U of Nantes, France)
- X. Liu (HIT, China)
- W. Hu, J. Liu, Z. Guo (Peking U., China)
- L. Fang (Tsinghua, China)
- C.-W. Lin (National Tsing Hua University, Taiwan)



NII Overview

- **National Institute of Informatics**
- Chiyoda-ku, Tokyo, Japan.
- Government-funded research lab.



- Offers graduate courses & degrees through **The Graduate University for Advanced Studies** (Sokendai).
- 60+ faculty in "**informatics**": quantum computing, discrete algorithms, database, machine learning, computer vision, speech & audio, image & video processing.

- **Get involved!**
    - 2-6 month Internships.
    - Short-term visits via MOU grant.
    - Lecture series, Sabbatical.

Introduction to APSIPA and APSIPA DL

APSIPA Mission: To promote broad spectrum of research and education activities in signal and information processing in Asia Pacific

APSIPA Conferences: ASPIPA Annual Summit and Conference

APSIPA Publications: Transactions on Signal and Information Processing in partnership with Cambridge Journals since 2012; APSIPA Newsletters

APSIPA Social Network: To link members together and to disseminate valuable information more effectively

APSIPA Distinguished Lectures: An APSIPA educational initiative to reach out to the community

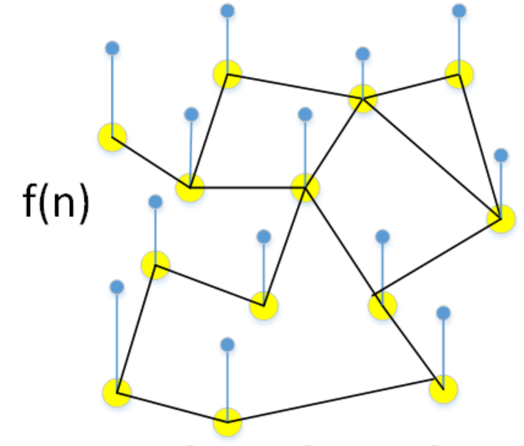
Outline

- Graph Signal Processing
    - Graph spectrum
- Semi-supervised Graph Classifier
    - Smoothness prior & MAP formulation
    - Graph construction
    - Graph Laplacian perturbation
    - Lower bound min eigenvalue computation
    - IRLS algorithm
- Experimental Results
- Conclusion

Outline

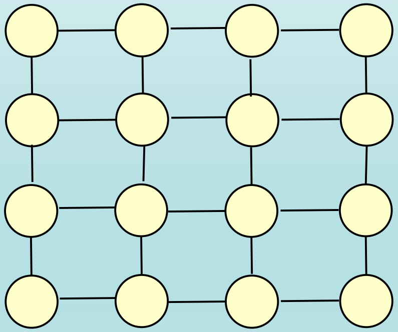
- Graph Signal Processing
  - Graph spectrum
- Semi-supervised Graph Classifier
  - Smoothness prior & MAP formulation
  - Graph construction
  - Graph Laplacian perturbation
  - Lower bound min eigenvalue computation
  - IRLS algorithm
- Experimental Results
- Conclusion

Graph Signal Processing



example graph-signal

- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.



1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.
2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.

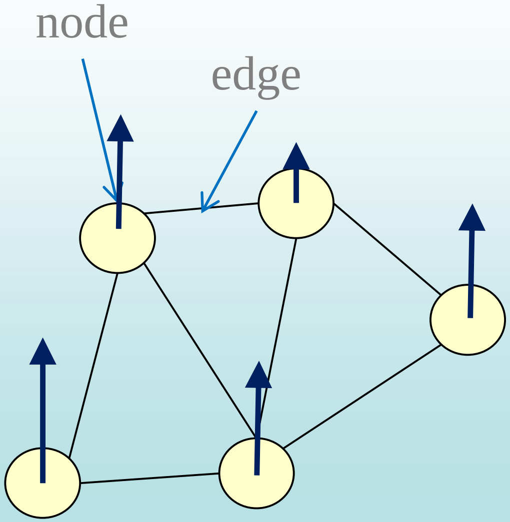
Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

[1] D. I. Shuman et al.,"**The Emerging Field of Signal Processing on Graphs: Extending High-dimensional Data Analysis to Networks and other Irregular Domains**," *IEEE Signal Processing Magazine*, vol.30, no.3, pp.83-98, 2013.

Graph Signal Processing

Research questions*:

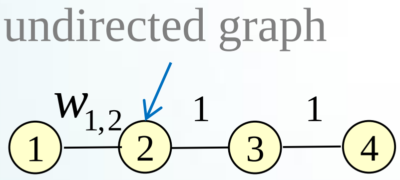
- **Sampling**: how to efficiently acquire / sense a graph-signal?
  - Graph sampling theorems.
- **Representation**: Given graph-signal, how to compactly represent it?
  - Transforms, wavelets, dictionaries.
- **Signal restoration**: Given noisy and/or partial graph-signal, how to recover it?
  - Graph-signal priors.



*__Graph Signal Processing Workshop__, Philadelphia, US, May, 2016. https://alliance.seas.upenn.edu/~gsp16/wiki/index.php?n=Main.Program

*__Graph Signal Processing Workshop__, Pittsburgh, US, May, 2017. https://gsp17.ece.cmu.edu/

Graph Fourier Transform (GFT)



Graph Laplacian:

- *Adjacency Matrix A*: entry $A_{i,j}$ has *non-negative* edge weight $w_{i,j}$ connecting nodes *i* and *j*.

$$A = \begin{bmatrix} 0 & w_{1,2} & 0 & 0 \\ w_{1,2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- *Degree Matrix D*: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row *i* of *A*.

$$D = \begin{bmatrix} w_{1,2} & 0 & 0 & 0 \\ 0 & w_{1,2}+1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D_{i,i} = \sum_j A_{i,j}$$

- *Combinatorial Graph Laplacian L*: *L = D-A*
  - ***L*** is *symmetric* (graph undirected).
  - ***L*** is a *high-pass* filter.
  - ***L*** is related to 2^{nd} *derivative*.

$$L = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$L_{3,:}x = -x_2 + 2x_3 - x_4$$

$$f''(x) = \lim_{h \to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

*https://en.wikipedia.org/wiki/Second_derivative

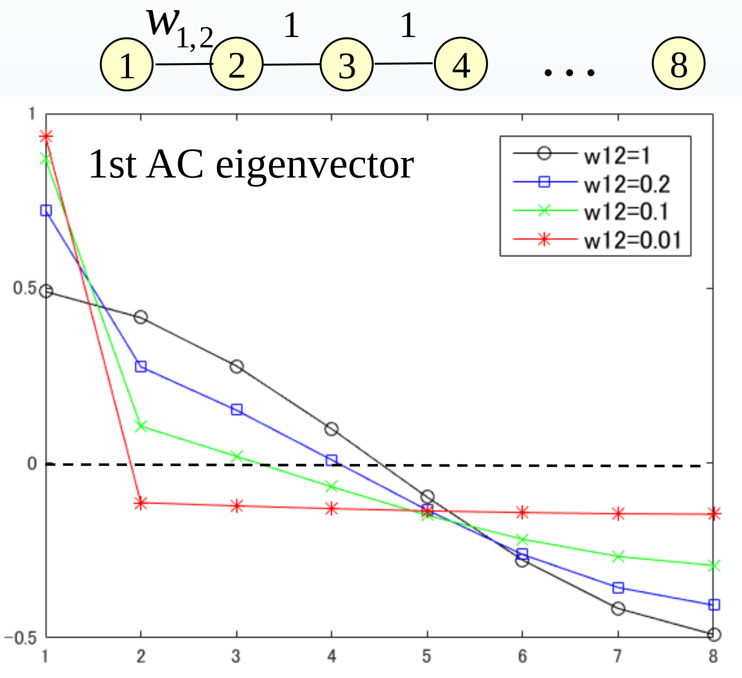
Graph Spectrum from GFT

- **Graph Fourier Transform** (GFT) is eigen-matrix of graph Laplacian *L*.

eigenvalue

$$L u_i = \lambda_i u_i$$

eigenvector



1. Edge weights affect shapes of eigenvectors.
2. Eigenvalues (≥ 0) as *graph frequencies*.
   - Constant eigenvector is DC.
   - # *zero-crossings* increases as λ increases.

- GFT defaults to *DCT* for un-weighted connected line.
- GFT defaults to *DFT* for un-weighted connected circle.

Variants of Graph Laplacians

- **Graph Fourier Transform** (GFT) is eigen-matrix of graph Laplacian *L*.

eigenvalue

$$L\,u_i = \lambda_i\,u_i$$

eigenvector

- Other definitions of graph Laplacians:
  - **Normalized** graph Laplacian:

$$L_n = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

  - **Random walk** graph Laplacian:

$$L_{rw} = D^{-1} L = I - D^{-1} A$$

  - **Generalized** graph Laplacian [1]:

$$L_g = L + D^{*}$$

Characteristics:

- Normalized.
- Symmetric.
- No DC component.

- Normalized.
- Asymmetric.
- Eigenvectors not orthog.

- Symmetric.
- **L** plus self loops.
- Defaults to DST, ADST.

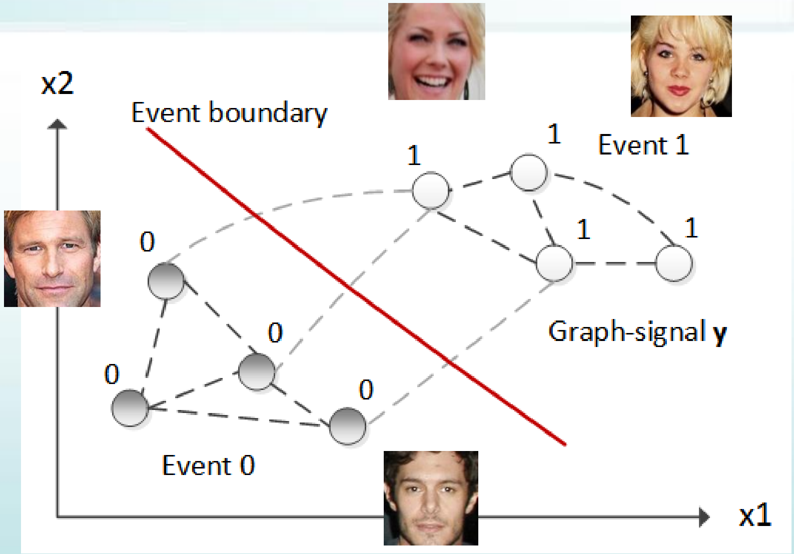
[1] Wei Hu, Gene Cheung, Antonio Ortega, "**Intra-Prediction and Generalized Graph Fourier Transform for Image Coding**," *IEEE Signal Processing Letters*, vol.22, no.11, pp. 1913-1917, November 2015.

Outline

- Graph Signal Processing
  - Graph spectrum
- Semi-supervised Graph Classifier
  - Smoothness prior & MAP formulation
  - Graph construction
  - Graph Laplacian perturbation
  - Lower bound min eigenvalue computation
  - IRLS algorithm
- Experimental Results
- Conclusion

Semi-Supervised Graph Classifier Learning

- **Binary Classifier**: given feature vector x_i of dimension K, compute $f(x_i) \in \{0,1\}$.
- **Classifier Learning**: given partial / noisy labels (x_i, y_i), train classifier $f(x_i)$.



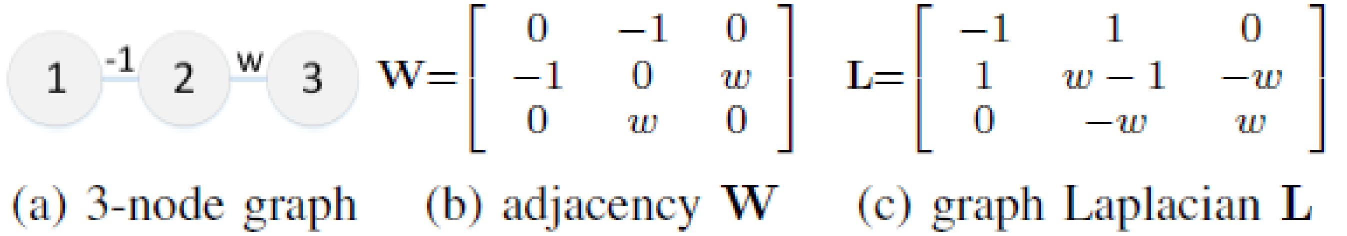
example graph-based classifier

- **GSP Approach** [1]:
  1. Construct ***similarity graph*** with +/- edges.
  2. Pose MAP graph-signal restoration problem.
  3. Perturb graph Laplacian to ensure PSD.
  4. Solve num. stable MAP as sparse lin. system.

[1] Yu Mao, Gene Cheung, Chia-Wen Lin, Yusheng Ji, "**Image Classifier Learning from Noisy Labels via Generalized Graph Smoothness Priors**," *IEEE IVMSP Workshop*, Bordeaux, France, July 2016. (**Best student paper award**)

[2] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, November 2016. (arXiv)

Graph-Signal Smoothness Prior

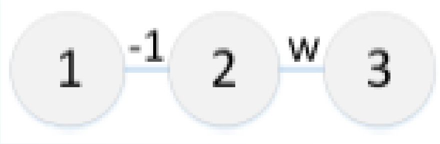


$$\mathbf{W}=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & w \\ 0 & w & 0 \end{bmatrix} \quad \mathbf{L}=\begin{bmatrix} -1 & 1 & 0 \\ 1 & w-1 & -w \\ 0 & -w & w \end{bmatrix}$$

(a) 3-node graph (b) adjacency **W** (c) graph Laplacian **L**

- **Smoothness:** signal "consistent" w/ underlying graph.
- **Q1:** how to define smoothness w.r.t. graph with +/- edges?
- **Q2:** is signal smoothness prior robust to errs?
- **Q3:** is signal smoothness prior easy to solve?

Graph-Signal Smoothness Prior: Candidate 1



w=-1

- **Shift-based Smoothness Prior [1]:**

$$\|\mathbf{x} - \mathbf{W}\mathbf{x}\|_2^2 = \|(\mathbf{I} - \mathbf{W})\mathbf{x}\|_2^2 = \left\| \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right\|_2^2$$

$$= (x_1 + x_2)^2 + (x_1 + x_2 + x_3)^2 + (x_2 + x_3)^2$$

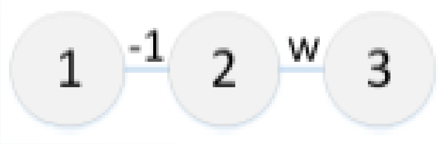
shifted version of signal

- Prior minimizes sums of sample values despite negative edges!
- **Counter example**:
  - x = [ρ, ρ+100, ρ], for large ρ

- Agrees w/ negative edges,
- Large penalty.

[1] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, “**Signal recovery on graphs: Variation minimization**,” *IEEE Transactions on Signal Processing*, vol. 63, no.17, September 2015, pp. 4609–4624.

Graph-Signal Smoothness Prior: Candidate 2



w=1

- **Total Variation (TV) [1] on graph:**

$$|\mathbf{Lx}| = \left| \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right| = \left| \begin{matrix} x_2 - x_1 \\ x_1 - x_3 \\ x_3 - x_2 \end{matrix} \right|$$

degree 0 at node 2

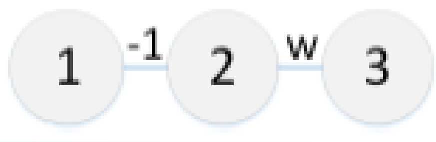
- Prior minimizes diffs in every pair!
- **Counter example**:
  - x = [ρ, ρ, ρ], for ρ > 0

- Disagrees w/ negative edge,
- Zero penalty.

---

[1] L. Rudin, S. Osher, and E. Fatemi, “**Nonlinear total variation based noise removal algorithms**,” *Physica D*, vol. 60, no.1-4, November 1992, pp. 259–268.

Graph-Signal Smoothness Prior: Candidate 3



w=1

- **Signed graph Laplacian [1]:**

$$D^s_{i,i} = \sum_j |w_{i,j}|$$

$$\mathbf{x}^T \mathbf{L}^s \mathbf{x} = (x_1 + x_2)^2 + (x_2 - x_3)^2$$

$$\mathrm{L}^s = \mathrm{D}^s - \mathrm{W}$$

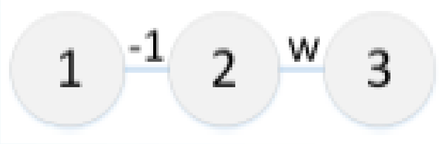
- Prior minimizes sum of first two samples!
- **Counter example**:
  - x = [ρ, -ρ, -ρ], for small ρ

- Disagrees w/ negative edge,
- Zero penalty.

---

[1] J. Kunegis et al., "**Spectral analysis of signed graphs for clustering, prediction and visualization**," in *SIAM International Conference on Data Mining*, Columbus, Ohio, May 2010.

Graph-Signal Smoothness Prior: Candidate 4



w=1

- **Graph Laplacian Regularizer [1]:**

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{(i,j)\in\mathcal{E}} w_{i,j}\left(x_i - x_j\right)^2 = \sum_k \lambda_k\, \alpha_k^2$$

GFT coeff

eigenvalues / graph freqs

- Promote large / small inter-node differences depending on edge signs.

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = -1(x_1 - x_2)^2 + (x_2 - x_3)^2$$

Promote large difference

Promote small difference

- Sensible, but numerically unstable.

[1] J. Pang and G. Cheung, "**Graph Laplacian regularization for image denoising: Analysis inn the continuous domain**," in *IEEE Transactions on Image Processing*, vol. 26, no.4, April 2017, pp. 1770–1785.

MAP Problem Formulation

- **Label Noise Model:** uniform noise model [1]

$$Pr(y_i|x_i) = \begin{cases} 1-p & \text{if } \; y_i = x_i \\ p & \text{o.w.} \end{cases}$$

- Probability of observing noisy **y** given ground truth **x**:

$$Pr(\mathbf{y}|\mathbf{x}) = p^k (1-p)^{K-k}$$
$$k = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0$$

- MAP formulation:

perturbation matrix
to ensure PSD!

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 + \mu \, \mathbf{x}^T \left(\mathbf{L} + \boldsymbol{\Delta}\right) \mathbf{x}$$

fidelity term

graph-signal smoothness prior

---

[1] A. Brew, D. Greene, and P. Cunningham, "**The interaction between supervised learning and crowdsourcing**," *Computational Social Science and the Wisdom of Crowds Workshop at NIPS*, Whistler, Canada, December 2010.

Graph Construction: add positive edges

- Given feature vector per sample in high dim. space.
- First to construct (dis)similarity graph with +/- edges from features.
- Positive edge weights reflect inter-node ***similarity***:

$$w_{i,j} = \exp\left(-\frac{(\mathbf{h}_i - \mathbf{h}_j)^T \Xi (\mathbf{h}_i - \mathbf{h}_j)}{\sigma_h^2}\right)$$

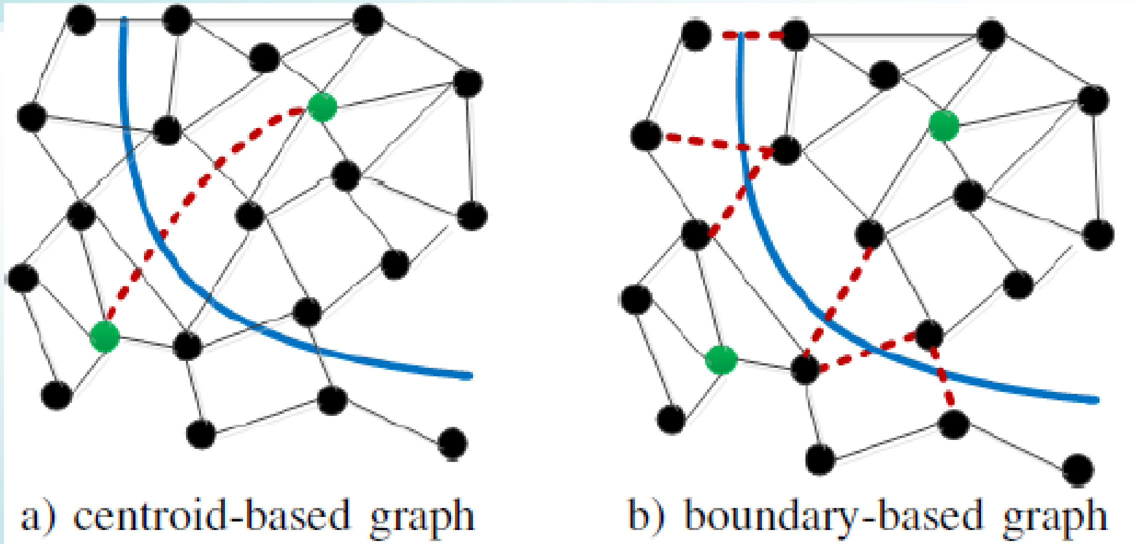
inter-node
feature distance

- Optimization of feature weights in [1].

---

[1] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "**Automated variable weighting in k-means type clustering**," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no.5, May 2005, pp. 657–668.

Graph Construction: add negative edges


a) centroid-based graph b) boundary-based graph

Centroid-based: add negative edge connecting cluster centroids.

- Connect dissimilar nodes.
- Robust, not precise.

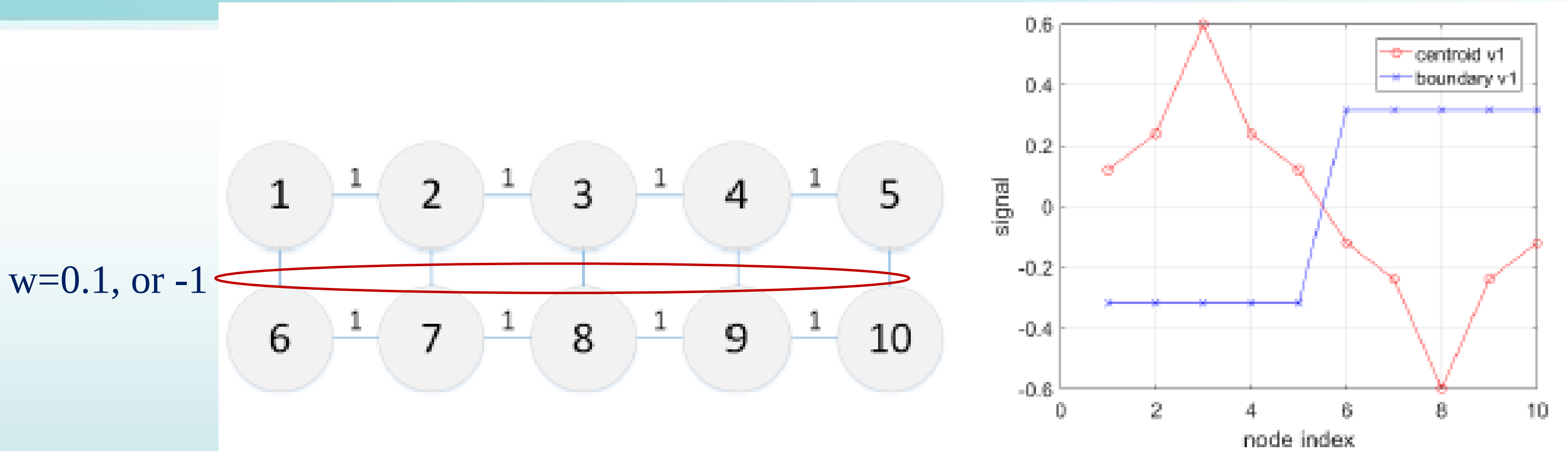
Boundary-based: add negative edges connecting boundary nodes of two clusters.

- Precise, not robust.

Idea: use convex combination as we iterate:

$$L^* = \beta(L_1 + \Delta_1) + (1 - \beta)(L_2 + \Delta_2)$$

Example: 10-node graph



a) 10-node graph b) first eigenvectors

- **Centroid-based 1st e-vector:** peaks at neg. edge endpoints.
- **Boundary-based 1st e-vector:** same level @ boundary nodes.

- *Low graph frequencies of indefinite L are useful in restoration* [1].

---

[1] A. Knyazev, “Signed Laplacian for spectral clustering revisited,” January 2017, https://arxiv.org/abs/1701.01394.

Finding Perturbation Matrix:
min norm

- **Minimum norm criteria:** smallest △ to ensure PSD:

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\| \quad \text{s.t.} \quad \mathbf{x}^T(\mathbf{L}+\boldsymbol{\Delta})\,\mathbf{x} \geq 0, \ \forall \mathbf{x}$$

$$\mathbf{L} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$$

assume has *p* negative eigenvalues

- Sol'n is special case of Thm 5.1 in [1]:

$$\boldsymbol{\Delta} = \mathbf{V}\ \mathrm{diag}(\boldsymbol{\tau})\ \mathbf{V}^T$$

$$\tau_i = \begin{cases} -\lambda_i & \text{if } 1 \leq i \leq p \\ 0 & \text{o.w.} \end{cases}$$

- **Observations**:
  1. L+△ is PSD (good).
  2. L+△ preserves same eigen-vectors (good).
  3. Eigenvalue 0 has p+1 eigen-vectors (bad).

[1] A. N. Higham and S. H. Cheng, "**Modifying the inertia of matrices arising in optimization**," *ELSEVIER Linear Algebra and its Applications*, vol. 275-279, May 1998, pp. 261–279.

Finding Perturbation Matrix:
eigen-structure preservation

- Perturb to ensure PSD while preserving *frequency components* (eigenvectors) and *frequency preferences*.

- One sol’n is $\triangle = \lambda_{\min} \mathbf{I}$, *i.e.* shift all eigenvalues up by $\eta = \lambda_{\min}$.
- **Intuition**: signal variations + signal energies

$$\mathbf{x}^T(\mathbf{L}+\boldsymbol{\Delta})\mathbf{x} = \mathbf{x}^T\mathbf{L}\mathbf{x} + \eta\,\mathbf{x}^T\mathbf{I}\mathbf{x}$$
$$= \sum_{i,j} w_{i,j}(x_i - x_j)^2 + \eta \sum_i x_i^2$$

signal energies

signal variations

- **Q:** computed lower-bound for $\lambda_{\min}$ w/o eigen-decomposition?

Fast Eigenvalue Methods

- Power iteration method [1]:
  - finds largest eigenvalue in magnitude.
- Lanczos method and variants [2]:
  - Prior knowledge about range of target eigenvalue.
- Jacobi-Davidson [3], Chebyshev-Davidson [4]:
  - Extremal eigenvalues / eigenvectors.

Goal: lower-bound of smallest negative eigenvalue

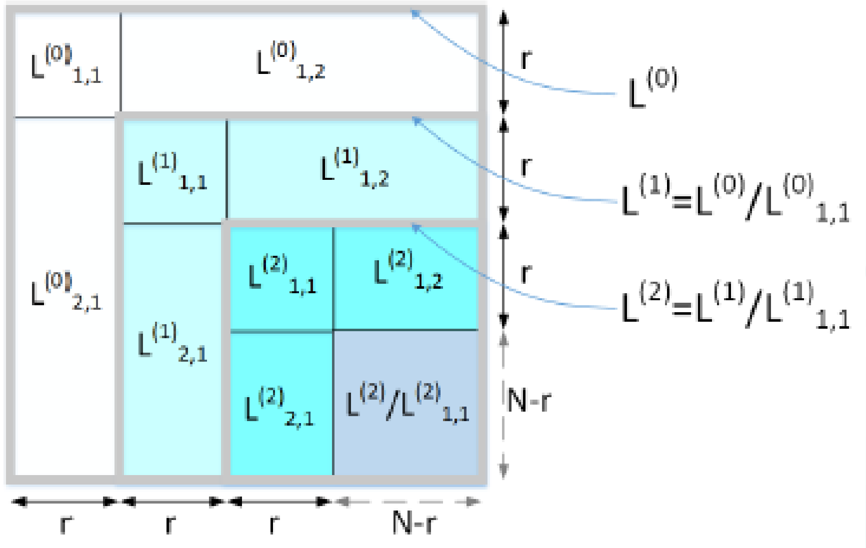
[1] A. N. Higham and S. H. Cheng, "**Modifying the inertia of matrices arising in optimization**," *ELSEVIER Linear Algebra and its Applications*, vol. 275-279, May 1998, pp. 261–279.

[2] G. Golub and C. F. V. Loan, Matrix Computations (Johns Hopkins Studies in the Mathematical Sciences). Johns Hopkins University Press, 2012.

[3] G. Sleijpen and H. V. D. Vorst, "A Jacobi-Davidson iteration method for linear eigenvalue problems," in SIAM J. Matrix Anal. and Appl., vol. 17, no.2, 1996, pp. 401–425.

[4] Y. Zhou and Y. Saad, "A Chebyshev-Davidson algorithm for large symmetric problems," in SIAM J. Matrix Anal. and Appl., vol. 29, no.3, 2007, pp. 954–971.

Lower Bound λ_{min}



- **Matrix Inertia**:

$$\mathrm{In}(\mathbf{A}) = \left(i^{+}(\mathbf{A}),\ i^{-}(\mathbf{A}),\ i^{0}(\mathbf{A})\right)$$

- **Haysworth Inertia Additivity**:

$$\mathrm{In}(\mathbf{L}) = \mathrm{In}(\mathbf{L}_{1,1}) + \mathrm{In}(\mathbf{L}/\mathbf{L}_{1,1})$$

Schur complement

$$\mathbf{L}/\mathbf{L}_{1,1} = \mathbf{L}_{2,2} - \mathbf{L}_{1,2}^{T}\mathbf{L}_{1,1}^{-1}\mathbf{L}_{1,2}$$

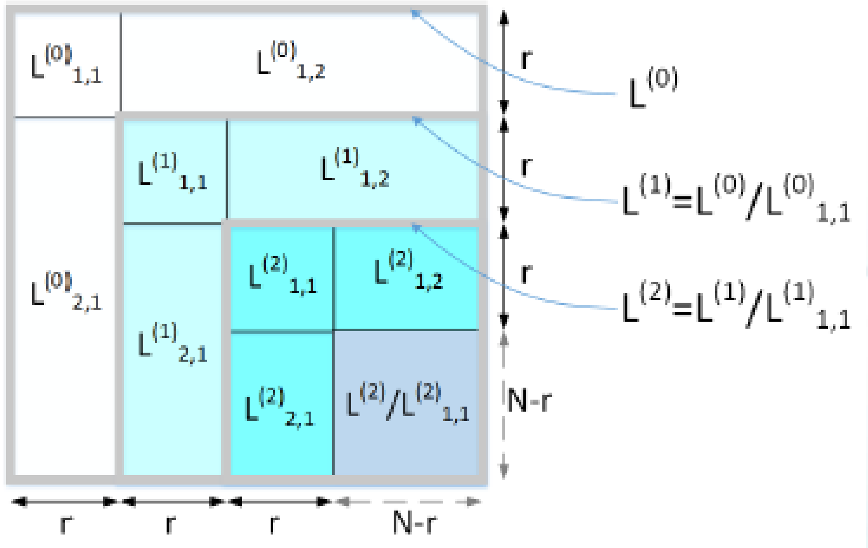
- **EvalBound (*L^t*, t)**
  - **Step 1:** divide N^t nodes in L^t into r and $N^t - r$ nodes.
    - Eigen-decompose $L^t_{1,1}$ to find smallest eigenvalue $\lambda^t_{1,1}$.
    - Perturb L^t by augmented eigenvalue κ^t_{min}

$$\kappa^t_{\mathrm{min}} = \begin{cases} \lambda^t_1 - \epsilon & \text{if} \quad \lambda^t_1 \leq 0 \\ 0 & \text{o.w.} \end{cases}$$

Ensure $L^t_{1,1}$ is PD.

[1] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, November 2016. (arXiv)

Lower Bound $\lambda_{\min}$



- **Haysworth Inertia Additivity**:

$$\mathrm{In}(\mathbf{L}) = \mathrm{In}(\mathbf{L}_{1,1}) + \mathrm{In}(\mathbf{L}/\mathbf{L}_{1,1})$$

- **Step 2:** ensure SC of $L^t_{1,1}$ is PSD: $\mathcal{L}^t/\mathcal{L}^t_{1,1} = \mathcal{L}^t_{2,2} - (\mathbf{L}^t_{1,2})^T(\mathcal{L}^t_{1,1})^{-1}\mathbf{L}^t_{1,2}$
  - if $N^t - r \leq$ r,
    - eigen-decompose L^t / $L^t_{1,1}$ to find smallest eigenvalue λ^t_2.
    - Compute lower bound: $\lambda^t_{\min} := \kappa^t_{\min} + \min\left(\lambda^t_2, 0\right)$
  - if $N^t - r >$ r,
    - Define $\mathbf{L}^{t+1} := \mathcal{L}^t/\mathcal{L}^t_{1,1}$
    - Recursively call $\eta^t_{\min} := EvalBound(\mathbf{L}^{t+1}, t+1)$
    - Return $\lambda^t_{\min} := \kappa^t_{\min} + \eta^t_{\min}$

Complexity $O(N^2\, r)$.

[1] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, November 2016. (arXiv)

IRLS Optimization

- MAP formulation:

$$\mathrm{L_g} = \mathrm{L} + \lambda^{\#}_{\min}\mathrm{I}$$

$$\min_{\mathbf{x}} \ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 \, \gamma + \sigma_0^{-2} \ \mathbf{x}^T \, \mathbf{L}_g \, \mathbf{x}$$

- ***Iterative Recursive Least Square*** (IRLS) [1]:
  - Replace L0-norm with weighted L2-norm, solve iteratively.

$$\min_{\mathbf{x}} \ (\mathbf{y} - \mathbf{H}\mathbf{x})^T \mathbf{B} (\mathbf{y} - \mathbf{H}\mathbf{x}) \gamma + \sigma_0^{-2} \ \mathbf{x}^T \, \mathbf{L}_g \, \mathbf{x}$$

diagonal matrix w/ weights *b*'s

$$b_i^{(t+1)} = \frac{1}{(y_i - \mathbf{H}_{i,:}\mathbf{x}^{(t)})^2 + \epsilon}$$

mimics L0-norm

- Sparse linear system of equations:

$$\left(\gamma \mathbf{H}^T \mathbf{B} \mathbf{H} + \sigma_0^{-2} \mathbf{L}_g\right) \mathbf{x}^* = \gamma \mathbf{H}^T \mathbf{B}^T \mathbf{y}$$

  - Matrix is sparse, symmetric, positive definite.
  - Solve via *conjugate gradient* instead of matrix inversion.

---

[1] I. Daubechies, R. Devore, M. Fornasier, and S. Gunturk, "**Iteratively re-weighted least squares minimization for sparse recovery**," *Communications on Pure and Applied Mathematics*, vol. 63, no.1, January 2010, pp. 1–38.

Outline

- Graph Signal Processing
  - Graph spectrum
- Semi-supervised Graph Classifier
  - Smoothness prior & MAP formulation
  - Graph construction
  - Graph Laplacian perturbation
  - Lower bound min eigenvalue computation
  - IRLS algorithm
- Experimental Results
- Conclusion

Experimental Setup

- KEEL database [1], face gender dataset [2].
- Features extracted for each sample; ex., local binary pattern (LBP).
- 70% / 30% are training / testing data.
- Graph construction:
  - kNN for positive edges (k=3).
  - Centroid / boundary-based negative edges.
- **Comparison schemes**:
  1. Linear SVM, SVM with RBF kernel
  2. RobustBoost
  3. Graph-Pos, Graph-MinNorm
  4. Graph-Bandlimited, Graph-AdjSmooth, Graph-Wavelet

---

[1] J. A.-F. et al., "Keel: A software tool to assess evolutionary algorithms to data mining problems," *Soft Computing*, vol. 13, no.3, February 2009, pp. 307–318.

[2] L. Spacek, "Face recognition data, university of essex, uk," http://cswww.essex.ac.uk/mv/allfaces/faces94.html, Feb. 2007.

Experimental Results

- Comparisons w/ other classifiers:

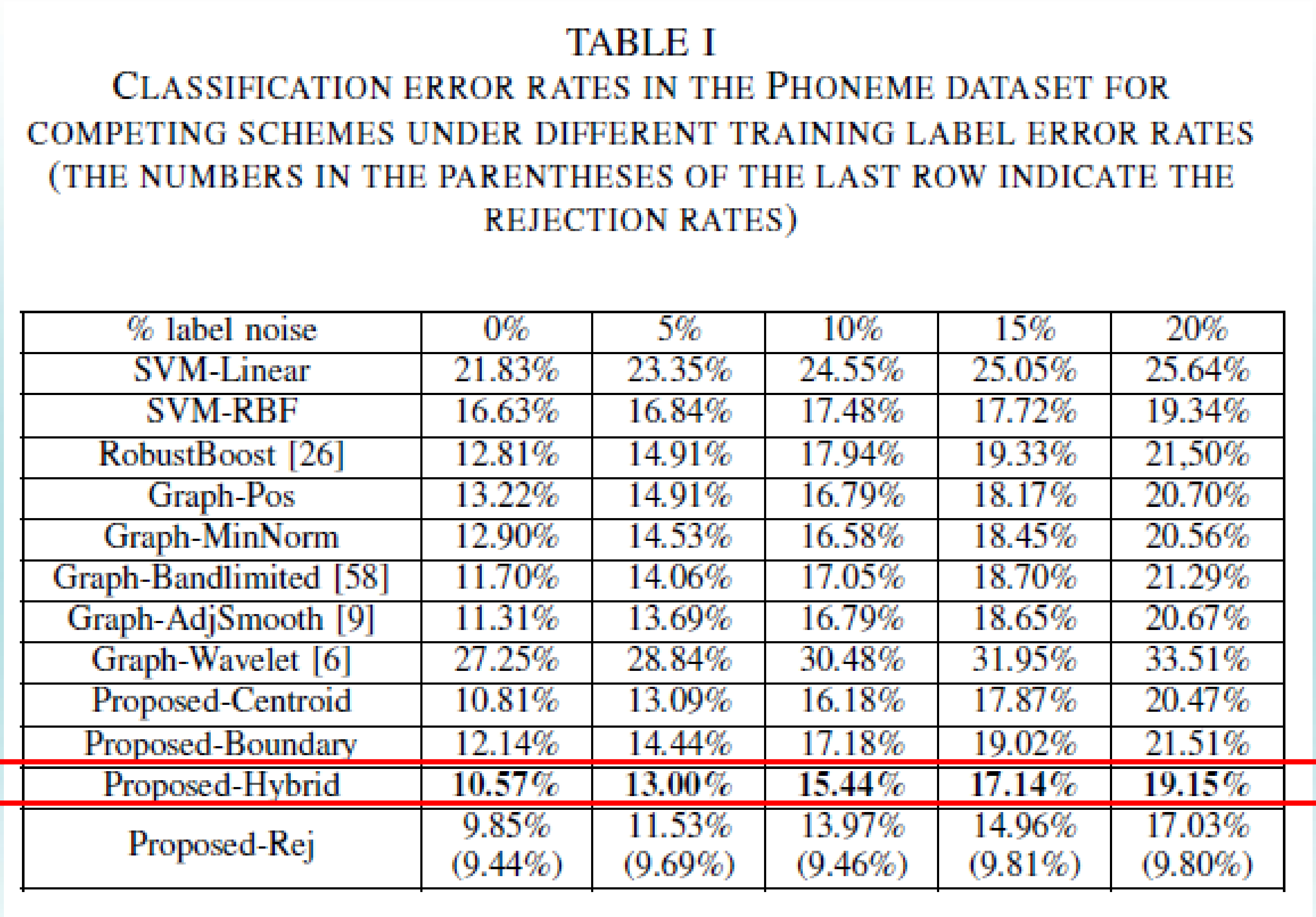


TABLE I

CLASSIFICATION ERROR RATES IN THE PHONEME DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	21.83%	23.35%	24.55%	25.05%	25.64%
SVM-RBF	16.63%	16.84%	17.48%	17.72%	19.34%
RobustBoost [26]	12.81%	14.91%	17.94%	19.33%	21,50%
Graph-Pos	13.22%	14.91%	16.79%	18.17%	20.70%
Graph-MinNorm	12.90%	14.53%	16.58%	18.45%	20.56%
Graph-Bandlimited [58]	11.70%	14.06%	17.05%	18.70%	21.29%
Graph-AdjSmooth [9]	11.31%	13.69%	16.79%	18.65%	20.67%
Graph-Wavelet [6]	27.25%	28.84%	30.48%	31.95%	33.51%
Proposed-Centroid	10.81%	13.09%	16.18%	17.87%	20.47%
Proposed-Boundary	12.14%	14.44%	17.18%	19.02%	21.51%
Proposed-Hybrid	**10.57%**	**13.00%**	**15.44%**	**17.14%**	**19.15%**
Proposed-Rej	9.85% (9.44%)	11.53% (9.69%)	13.97% (9.46%)	14.96% (9.81%)	17.03% (9.80%)

Experimental Results

- Comparisons w/ other classifiers:

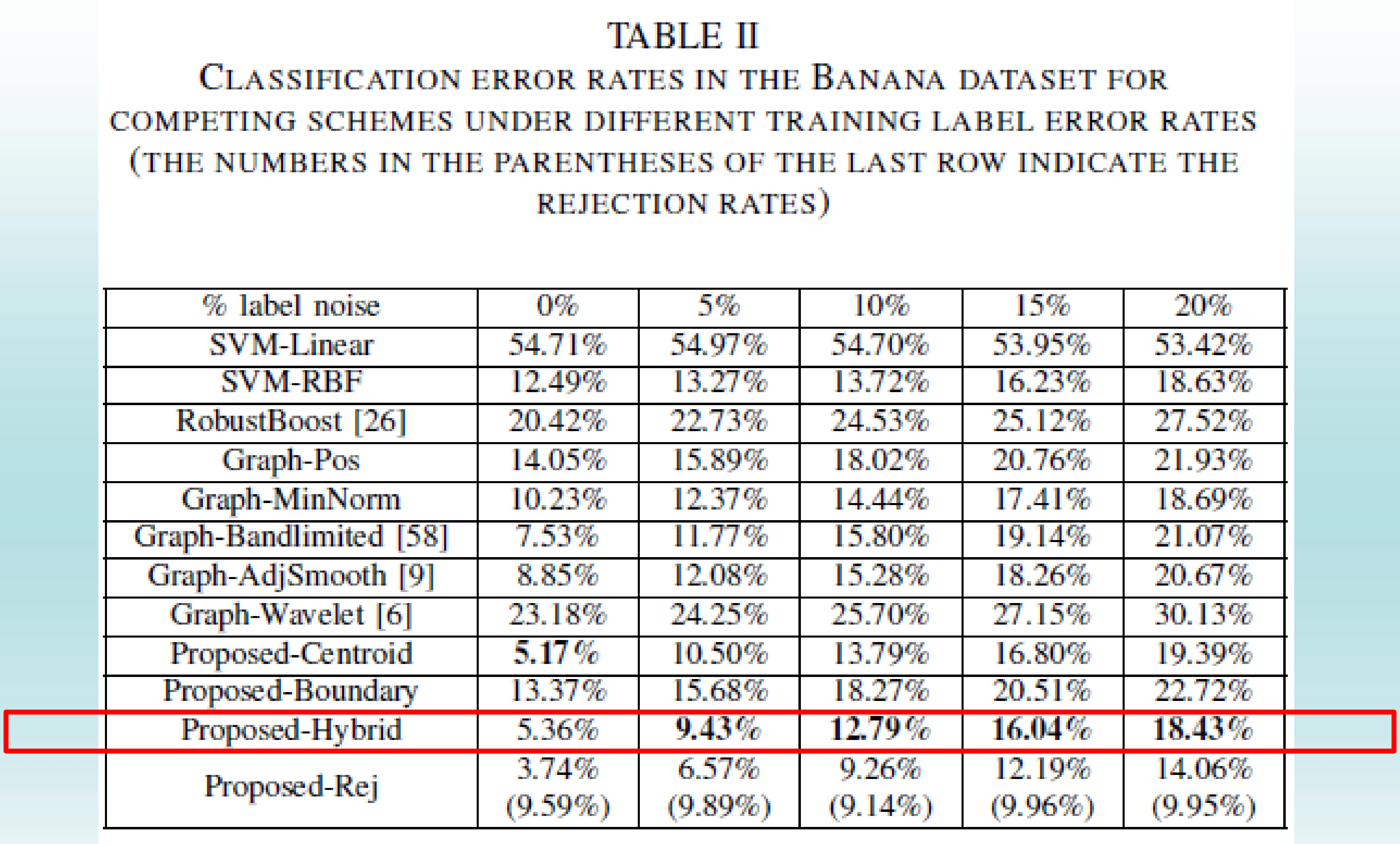


TABLE II
CLASSIFICATION ERROR RATES IN THE BANANA DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	54.71%	54.97%	54.70%	53.95%	53.42%
SVM-RBF	12.49%	13.27%	13.72%	16.23%	18.63%
RobustBoost [26]	20.42%	22.73%	24.53%	25.12%	27.52%
Graph-Pos	14.05%	15.89%	18.02%	20.76%	21.93%
Graph-MinNorm	10.23%	12.37%	14.44%	17.41%	18.69%
Graph-Bandlimited [58]	7.53%	11.77%	15.80%	19.14%	21.07%
Graph-AdjSmooth [9]	8.85%	12.08%	15.28%	18.26%	20.67%
Graph-Wavelet [6]	23.18%	24.25%	25.70%	27.15%	30.13%
Proposed-Centroid	**5.17%**	10.50%	13.79%	16.80%	19.39%
Proposed-Boundary	13.37%	15.68%	18.27%	20.51%	22.72%
Proposed-Hybrid	5.36%	**9.43%**	**12.79%**	**16.04%**	**18.43%**
Proposed-Rej	3.74% (9.59%)	6.57% (9.89%)	9.26% (9.14%)	12.19% (9.96%)	14.06% (9.95%)

Experimental Results

- Comparisons w/ other classifiers:

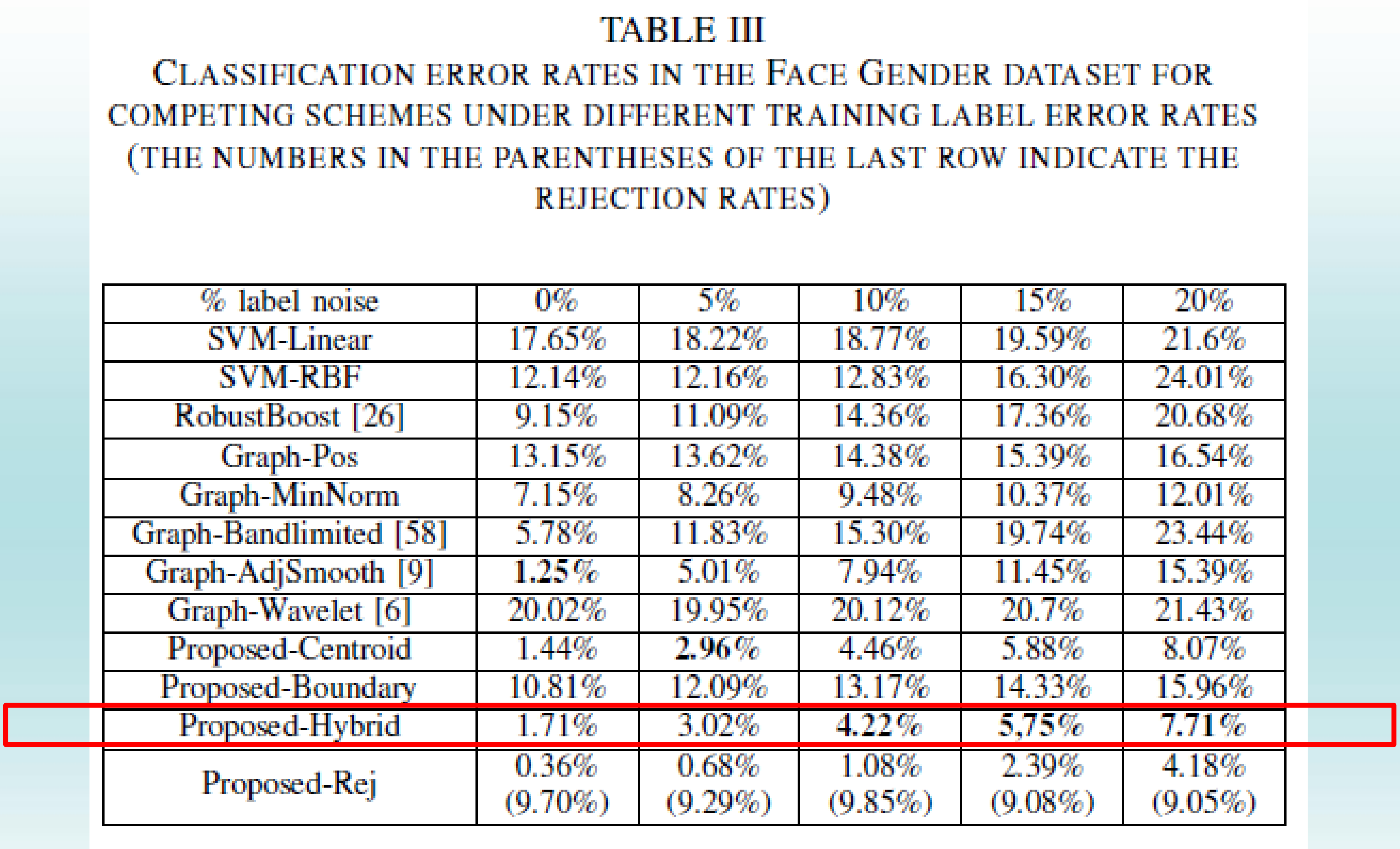


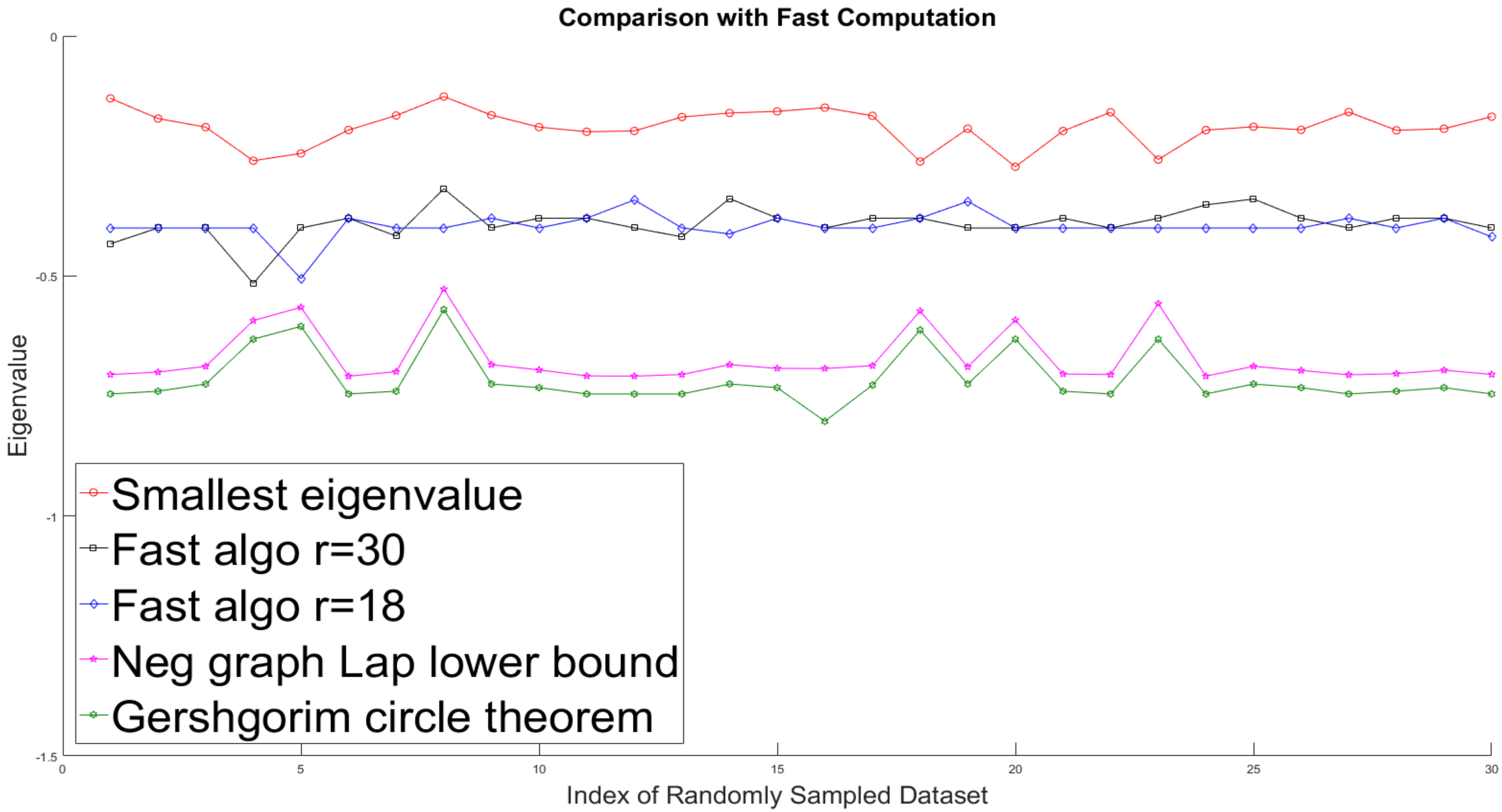
TABLE III

CLASSIFICATION ERROR RATES IN THE FACE GENDER DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	17.65%	18.22%	18.77%	19.59%	21.6%
SVM-RBF	12.14%	12.16%	12.83%	16.30%	24.01%
RobustBoost [26]	9.15%	11.09%	14.36%	17.36%	20.68%
Graph-Pos	13.15%	13.62%	14.38%	15.39%	16.54%
Graph-MinNorm	7.15%	8.26%	9.48%	10.37%	12.01%
Graph-Bandlimited [58]	5.78%	11.83%	15.30%	19.74%	23.44%
Graph-AdjSmooth [9]	**1.25%**	5.01%	7.94%	11.45%	15.39%
Graph-Wavelet [6]	20.02%	19.95%	20.12%	20.7%	21.43%
Proposed-Centroid	1.44%	**2.96%**	4.46%	5.88%	8.07%
Proposed-Boundary	10.81%	12.09%	13.17%	14.33%	15.96%
Proposed-Hybrid	1.71%	3.02%	**4.22%**	**5.75%**	**7.71%**
Proposed-Rej	0.36% (9.70%)	0.68% (9.29%)	1.08% (9.85%)	2.39% (9.08%)	4.18% (9.05%)

Experimental Results

- λ_{min} versus computed lower bound:

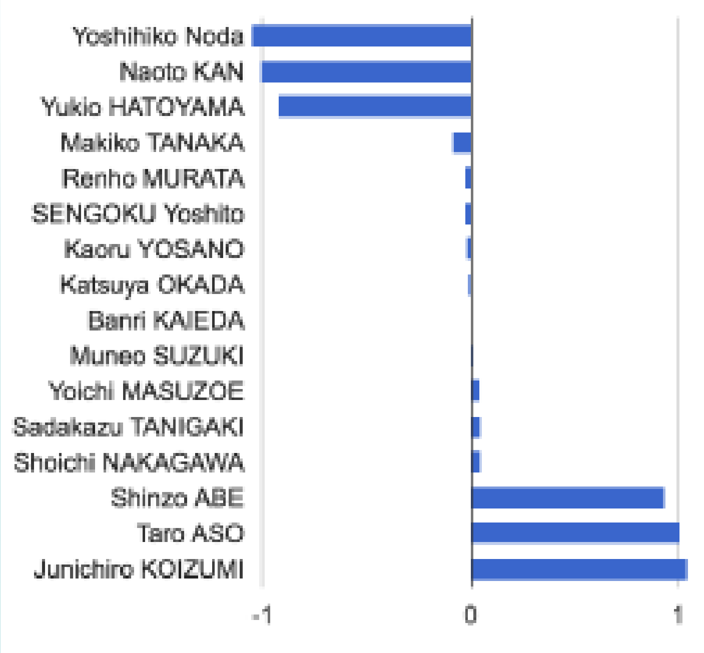


Conclusion

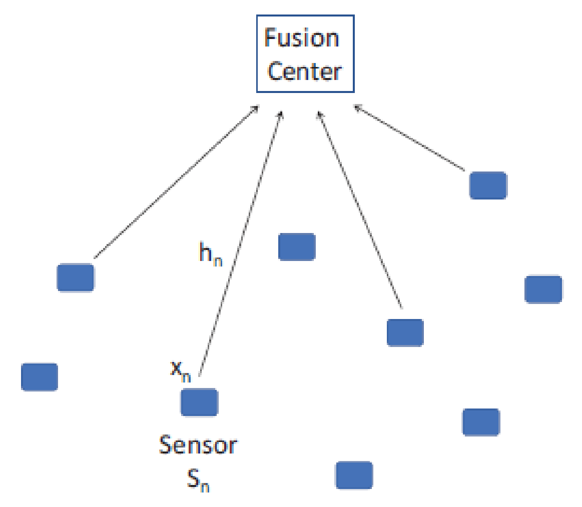
- Graph Signal Processing (GSP)
  - Tools to process signals that live on graphs.
- Graph-based binary classifier
  - Similarity graph with +/- edges, given features.
  - Perturbed graph Laplacian that is PSD.
    - Fast computation of min eigenvalue lower bound.
  - Fast MAP solver via IRLS, conjugate gradient.

Other GSP Works

- Coding of LF, spectral image [1], 3D point cloud w/ GFT.
- Coding of graph data w/ graph wavelets.
- Political leaning estimation [2].
- Wireless signal / power estimation [3].



[1] J. Zeng, G. Cheung, Y.-H. Chao, I. Blanes, J. Serra-Sagrista, A. Ortega, "**Hyperspectral Image Coding using Graph Wavelets**," *IEEE International Conference on Image Processing*, Beijing, China, September, 2017.



[2] B. Renoust et al., "**Estimation of Political Leanings via Graph-Signal Restoration**," *IEEE International Conference on Multimedia and Expo*, Hong Kong, China, July, 2017

[3] M. Kaneko, G. Cheung, W.-t. Su, C.-W. Lin, "**Graph-based Joint Signal / Power Restoration for Energy Harvesting Wireless Sensor Networks**," accepted to *IEEE Globecom*, Singapore, December, 2017.

Q&A

- Email: cheung@nii.ac.jp
- Homepage: http://research.nii.ac.jp/~cheung/