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

APSIPA Conferences: APSIPA 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

• Signals on \textit{irregular} data kernels described by graphs.
  • Graph: nodes and edges.
  • Edges reveals \textit{node-to-node relationships}.

1. Data domain is naturally a graph.
  • \textbf{Ex}: ages of users on social networks.

2. Underlying data structure unknown.
  • \textbf{Ex}: images: 2D grid $\rightarrow$ structured graph.

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

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 Fourier Transform (GFT)

Graph Laplacian:

- **Adjacency Matrix** $A$: entry $A_{i,j}$ has non-negative edge weight $w_{i,j}$ connecting nodes $i$ and $j$.
- **Degree Matrix** $D$: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row $i$ of $A$.

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

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

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

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

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

PKU Visit 11/27/2017
Graph Spectrum from GFT

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

  $$Lu_i = \lambda_i u_i$$

  - eigenvalue
  - eigenvector

1. Edge weights affect shapes of eigenvectors.
2. Eigenvalues ($\geq 0$) as **graph frequencies**.
   - Constant eigenvector is **DC**.
   - # **zero-crossings** increases as $\lambda$ increases.
   - GFT defaults to **DCT** for un-weighted connected line.
   - GFT defaults to **DFT** for un-weighted connected circle.

PKU Visit 11/27/2017
Variants of Graph Laplacians

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

\[ Lu_i = \lambda_i u_i \]

eigenvalue

eigenvector

• Other definitions of graph Laplacians:

  • **Normalized** graph Laplacian:

\[ L_n = D^{-1/2} LD^{-1/2} = I - D^{-1/2} AD^{-1/2} \]

  • **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.

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 $f(x_i)$.

**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.

---


Graph-Signal Smoothness Prior

- **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

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

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

shifted version of signal

- Prior minimizes sums of sample values despite negative edges!

- **Counter example:**
  - \( x = [\rho, \rho + 100, \rho] \), for large \( \rho \)
  - Agrees w/ negative edges, Large penalty.

Graph-Signal Smoothness Prior: Candidate 2

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

\[
|Lx| = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ x_1 - x_3 \\ x_3 - x_2 \end{bmatrix}
\]

degree 0 at node 2

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

  - Disagrees w/ negative edge,
  - Zero penalty.

Graph-Signal Smoothness Prior: Candidate 3

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

\[
D_{i,i}^s = \sum_j |w_{i,j}|
\]

\[
L^s = D^s - W
\]

- Prior minimizes sum of first two samples!

- **Counter example:**
  - \(x = [\rho, -\rho, -\rho]\), for small \(\rho\)
  - Disagrees w/ negative edge,
  - Zero penalty.

Graph-Signal Smoothness Prior: Candidate 4

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

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

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

\[
x^T L x = -1(x_1 - x_2)^2 + (x_2 - x_3)^2
\]

- Sensible, but numerically unstable.

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 \( x \):

\[
Pr(y|x) = p^k (1 - p)^{K-k} \\
k = \|y - Hx\|_0
\]

- MAP formulation:

\[
\min_{x} \|y - Hx\|_0 + \mu x^T (L + \Delta) x
\]

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**:

\[
    w_{i,j} = \exp\left(-\frac{(h_i - h_j)^T \Xi (h_i - h_j)}{\sigma^2 h_i}\right)
\]

• Optimization of feature weights in [1].

Graph Construction: add negative edges

**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

- **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].

Finding Perturbation Matrix: min norm

• **Minimum norm criteria:** smallest $\Delta$ to ensure PSD:

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

\[
L = V \Lambda V^T
\]

• Sol’n is special case of Thm 5.1 in [1]:

\[
\Delta = V \text{ diag}(\tau) \ V^T
\]

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

• **Observations:**

1. $L + \Delta$ is PSD (**good**).
2. $L + \Delta$ preserves same eigen-vectors (**good**).
3. Eigenvalue 0 has $p+1$ eigen-vectors (**bad**).

Finding Perturbation Matrix: eigen-structure preservation

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

- One sol’n is $\Delta = \lambda_{\text{min}} I$, i.e. shift all eigenvalues up by $\eta = \lambda_{\text{min}}$.

- **Intuition**: signal variations + signal energies

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

- **Q**: computed lower-bound for $\lambda_{\text{min}}$ 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


Lower Bound $\lambda_{\text{min}}$

- **Matrix Inertia:**
  \[
  \text{In}(A) = (i^+(A), i^-(A), i^0(A))
  \]

- **Haysworth Inertia Additivity:**
  \[
  \text{In}(L) = \text{In}(L_{1,1}) + \text{In}(L/L_{1,1})
  \]

- **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 $\kappa^t_{\text{min}}$.

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

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

Lower Bound $\lambda_{\text{min}}$

- **Haysworth Inertia Additivity:**

\[
\text{In}(L) = \text{In}(L_{1,1}) + \text{In}(L/L_{1,1})
\]

- **Step 2:** ensure SC of $L_{1,1}^t$ is PSD:
  - if $N^t - r \leq r$,
    - eigen-decompose $L^t / L_{1,1}^t$ to find smallest eigenvalue $\lambda_{t,2}$.
    - Compute lower bound:
      
      \[
      \lambda_{\text{min}}^t := \kappa_{\text{min}}^t + \min (\lambda_{2}^t, 0)
      \]
  - if $N^t - r > r$,
    - Define $L_{1,1}^{t+1} := L^t / L_{1,1}^t$
    - Recursively call $\eta_{\text{min}}^t := \text{EvalBound}(L_{1,1}^{t+1}, t + 1)$
    - Return $\lambda_{\text{min}}^t := \kappa_{\text{min}}^t + \eta_{\text{min}}^t$

Complexity $O(N^2 r)$.

IRLS Optimization

- **MAP formulation:**
  \[
  \min_x \|y - Hx\|_0 \gamma + \sigma_0^{-2} x^T L_g x
  \]

- **Iterative Recursive Least Square (IRLS) [1]:**
  - Replace L0-norm with weighted L2-norm, solve iteratively.
  \[
  \min_x (y - Hx)^T B(y - Hx) \gamma + \sigma_0^{-2} x^T L_g x
  \]
  \[
  b_i^{(t+1)} = \frac{1}{(y_i - H_i x^{(t)})^2 + \epsilon}
  \]

- **Sparse linear system of equations:**
  \[
  (\gamma H^T B H + \sigma_0^{-2} L_g) x^* = \gamma H^T B^T y
  \]
  - Matrix is sparse, symmetric, positive definite.
  - Solve via *conjugate gradient* instead of matrix inversion.

---
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

Experimental Results

- Comparisons w/ other classifiers:

<table>
<thead>
<tr>
<th>% label noise</th>
<th>0%</th>
<th>5%</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-Linear</td>
<td>21.83%</td>
<td>23.35%</td>
<td>24.55%</td>
<td>25.05%</td>
<td>25.64%</td>
</tr>
<tr>
<td>SVM-RBF</td>
<td>16.63%</td>
<td>16.84%</td>
<td>17.48%</td>
<td>17.72%</td>
<td>19.34%</td>
</tr>
<tr>
<td>RobustBoost [26]</td>
<td>12.81%</td>
<td>14.91%</td>
<td>17.94%</td>
<td>19.33%</td>
<td>21.50%</td>
</tr>
<tr>
<td>Graph-Pos</td>
<td>13.22%</td>
<td>14.91%</td>
<td>16.79%</td>
<td>18.17%</td>
<td>20.70%</td>
</tr>
<tr>
<td>Graph-MinNorm</td>
<td>12.90%</td>
<td>14.53%</td>
<td>16.58%</td>
<td>18.45%</td>
<td>20.56%</td>
</tr>
<tr>
<td>Graph-Bandlimited [58]</td>
<td>11.70%</td>
<td>14.06%</td>
<td>17.05%</td>
<td>18.70%</td>
<td>21.29%</td>
</tr>
<tr>
<td>Graph-AdjSmooth [9]</td>
<td>11.31%</td>
<td>13.69%</td>
<td>16.79%</td>
<td>18.65%</td>
<td>20.67%</td>
</tr>
<tr>
<td>Graph-Wavelet [6]</td>
<td>27.25%</td>
<td>28.84%</td>
<td>30.48%</td>
<td>31.95%</td>
<td>33.51%</td>
</tr>
<tr>
<td>Proposed-Centroid</td>
<td>10.81%</td>
<td>13.09%</td>
<td>16.18%</td>
<td>17.87%</td>
<td>20.47%</td>
</tr>
<tr>
<td>Proposed-Boundary</td>
<td>12.14%</td>
<td>14.44%</td>
<td>17.18%</td>
<td>19.02%</td>
<td>21.51%</td>
</tr>
<tr>
<td>Proposed-Hybrid</td>
<td><strong>10.57%</strong></td>
<td><strong>13.00%</strong></td>
<td><strong>15.44%</strong></td>
<td><strong>17.14%</strong></td>
<td><strong>19.15%</strong></td>
</tr>
<tr>
<td>Proposed-Rej</td>
<td>9.85%</td>
<td>11.53%</td>
<td>13.97%</td>
<td>14.96%</td>
<td>17.03%</td>
</tr>
</tbody>
</table>

(9.44%)  (9.69%)  (9.46%)  (9.81%)  (9.80%)
Experimental Results

- Comparisons w/ other classifiers:

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

- Comparisons w/ other classifiers:

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

• $\lambda_{\text{min}}$ versus computed lower bound:

![Comparison with Fast Computation](image-url)

- Smallest eigenvalue
- Fast algo $r=30$
- Fast algo $r=18$
- Neg graph Lap lower bound
- Gershgorim circle theorem
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].


Q&A

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