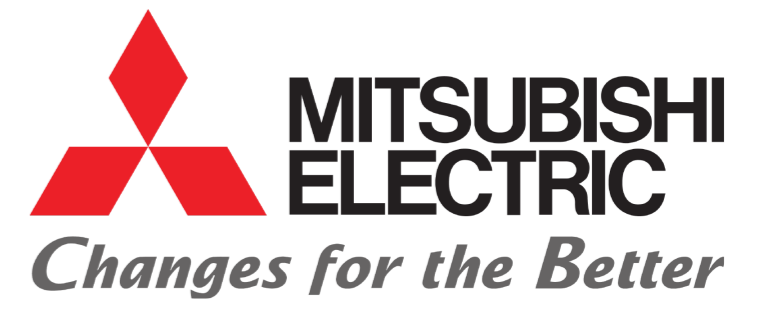


Disc-GLasso: Discriminative Graph Learning with Sparsity Regularization



USC University of Southern California



Jiun-Yu Kao¹, Dong Tian², Hassan Mansour², Antonio Ortega¹ and Anthony Vetro²

¹ University of Southern California, Los Angeles, CA

² Mitsubishi Electric Research Labs (MERL), Cambridge, MA

Motivation

Graph structures are natural to use

- Construct optimal graph not trivial
 - learn from data
- Prior art learn graphs w.r.t.
 - Representability: benefits energy compaction
 - Sparsity: benefits interpretation

Prior Art Approach and Issue Apply GLasso for data in each class independently

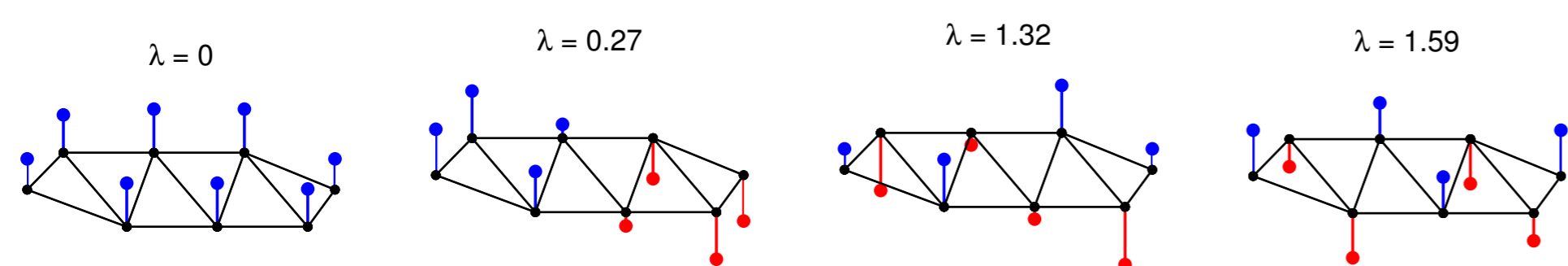
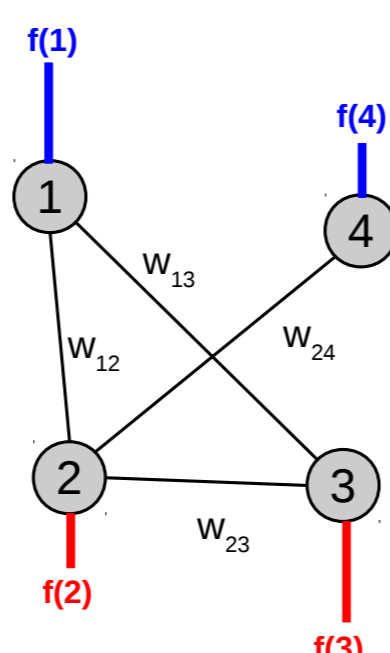
- Only consider representability and sparsity within each class
- Discrimination between classes not guaranteed

Key Ideas

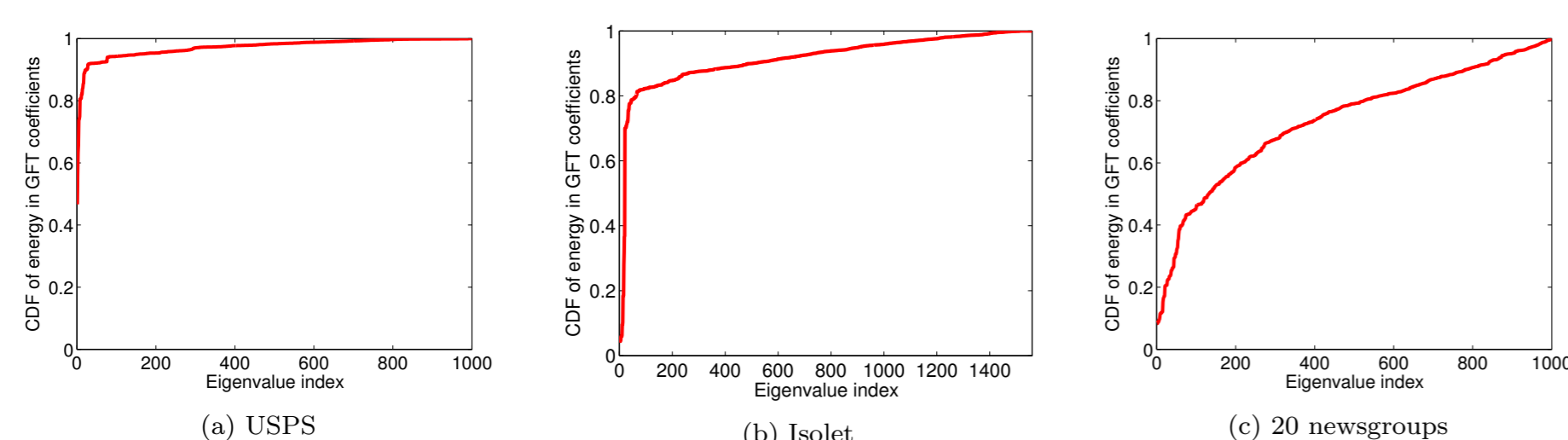
- Class labels \implies *Smooth or bandlimited graph signals*.
- Choosing nodes to label \implies *Best sampling set selection*.
- Predicting unknown labels \implies *Signal reconstruction from samples*.

Background: Graph Signal Processing

- **Graph** $G = (\mathcal{V}, \mathcal{E})$ with n nodes
 - nodes \implies data points.
 - w_{ij} : similarity between points i and j .
 - Adjacency matrix $\mathbf{W} = [w_{ij}]_{n \times n}$.
 - Degree matrix $\mathbf{D} = \text{diag}\{\sum_j w_{ij}\}$.
 - Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
 - Normalized Laplacian $\mathcal{L} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$.
- **Graph signal** $f : \mathcal{V} \rightarrow \mathbb{R}$, denoted as $\mathbf{f} \in \mathbb{R}^n$.
 - Membership function \mathbf{f}_i : $\mathbf{f}_i(j) = 1 \implies$ node j belongs to class i .
- Spectrum of $\mathcal{L} \implies$ spectral representation for graph signals.
 - Frequencies: $\{\lambda_k\} \in [0, 2]$; Fourier basis: $\{\mathbf{u}_k\}$.

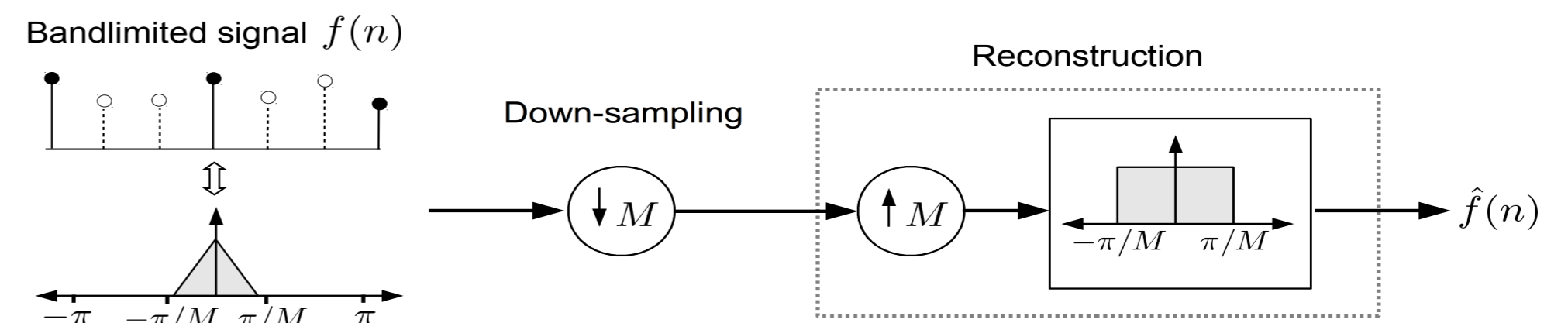


- **Graph Fourier Transform (GFT)**: $\tilde{\mathbf{f}}(\lambda_k) = \langle \mathbf{f}, \mathbf{u}_k \rangle$ or $\tilde{\mathbf{f}} = \mathbf{U}^\top \mathbf{f}$.
- **$\mathbf{PW}_\omega(\mathbf{G})$** : space of ω -bandlimited graph signals
 - Support of GFT = $[0, \omega]$, i.e., $\tilde{\mathbf{f}}(\lambda) = 0 \quad \forall \lambda > \omega$
- **Class membership functions are smooth graph signals.**

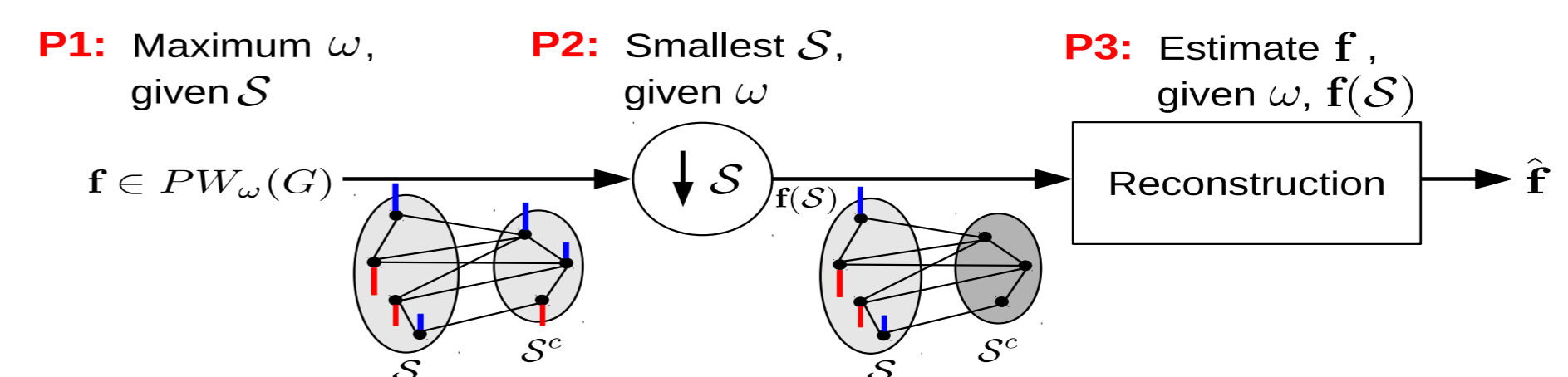


Sampling Theory for Graph Signals

- Sampling theorem: BW $\omega \Leftrightarrow$ sampling rate for unique representation

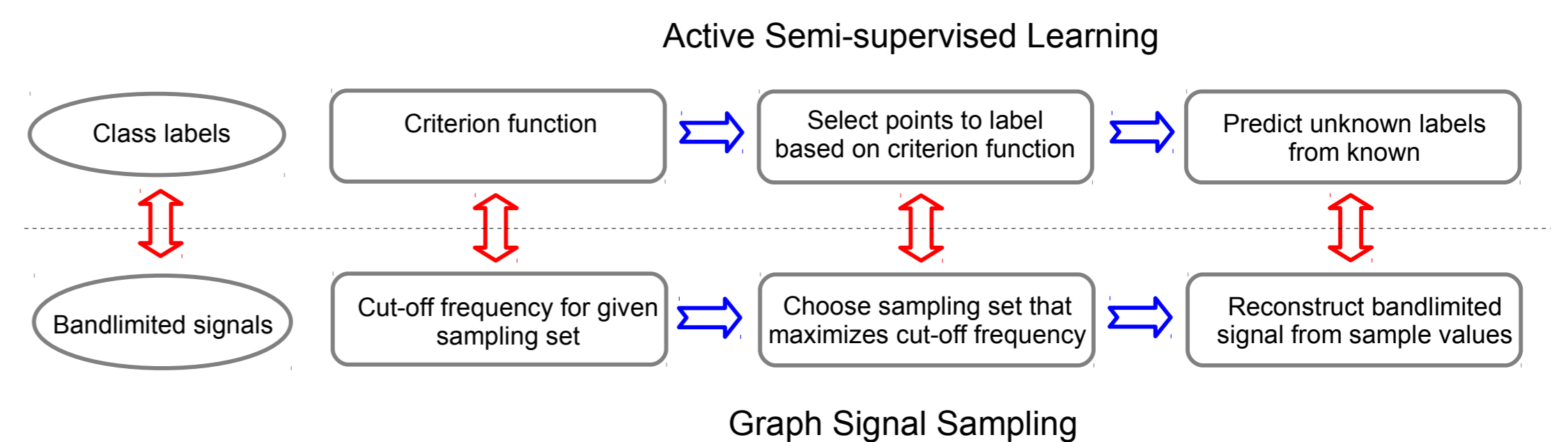


- Sampling theory for graph signals:



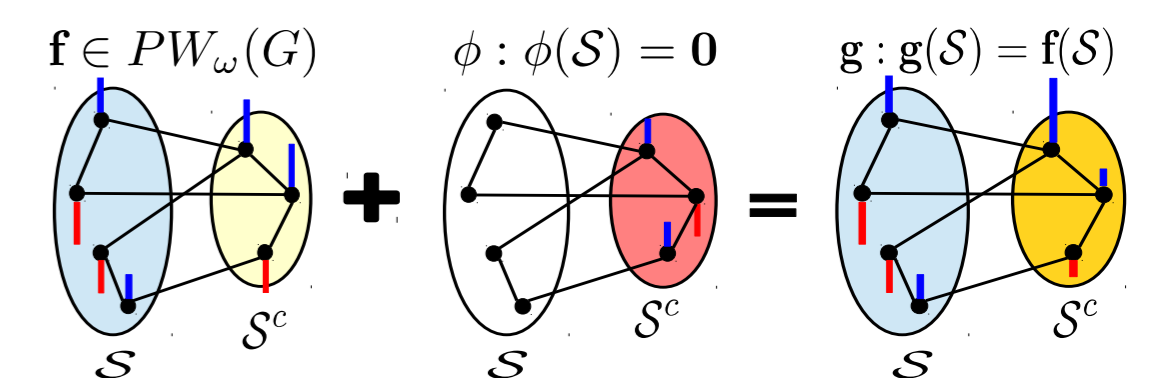
Active Semi-supervised Learning

Approach



P1: Cut-off frequency criterion

- $L_2(\mathcal{S}^c) = \{\phi : \phi(\mathcal{S}) = 0\}$.
- For unique sampling, we need $\mathbf{PW}_\omega(\mathbf{G}) \cap L_2(\mathcal{S}^c) = \{0\}$.
- Cut-off frequency = smallest BW that a $\phi \in L_2(\mathcal{S}^c)$ can have.
- Estimate by $\min_{\phi(\mathcal{S})=0} \left(\frac{\phi^\top \mathcal{L}^k \phi}{\phi^\top \phi} \right)^{1/k}$. Higher $k \implies$ better estimate.
- Let $\{\sigma_{1,k}, \psi_{1,k}\}$ be the smallest eigen-pair of $(\mathcal{L}^k)_{\mathcal{S}^c}$.
- Thus, cut-off estimate $\Omega_k(\mathcal{S}) = (\sigma_{1,k})^{1/k}$ with $\phi_{\text{opt}}(\mathcal{S}^c) = \psi_{1,k}$.



P2: Sampling set selection

- For given size, sampling set must be able to maximally capture signal information; $\mathcal{S}_{\text{opt}} = \arg \max_{|\mathcal{S}|=m} \Omega_k(\mathcal{S})$.

$$(\Omega_k(\mathcal{S}))^k = \min_{\phi(\mathcal{S})=0} \frac{\phi^\top \mathcal{L}^k \phi}{\phi^\top \phi} \approx \min_{\mathbf{x}} \left(\frac{\mathbf{x}^\top \mathcal{L}^k \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} + \alpha \frac{\mathbf{x}^\top \text{diag}(\mathbf{t}) \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \right) \Big|_{\mathbf{t}=\mathbf{1}_S} \xrightarrow{\text{binary relaxation}} \lambda_k^\alpha(\mathbf{t})|_{\mathbf{t}=\mathbf{1}_S}$$

relax the constraint

$$\mathbf{x}_{\text{opt}} \approx \phi_{\text{opt}} \implies \frac{d\lambda_k^\alpha(\mathbf{t})}{dt(i)} \approx \alpha(\phi_{\text{opt}}(i))^2.$$

- **Greedy algorithm**: $\mathcal{S} \leftarrow \mathcal{S} \cup v$, where $v = \arg \max_j (\phi_{\text{opt}}(j))^2$

References