Predicting Switching Graph Labelings with Cluster Specialists

MoN18: Eighteenth Mathematics of Networks Meeting

James Robinson (joint work with Mark Herbster)

8 April 2019

Department of Computer Science
University College London

Outline

Introduction

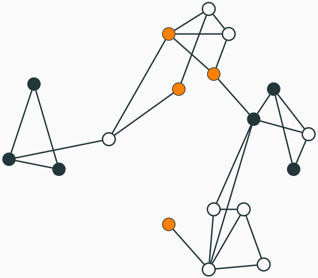
Introduction

- Graph prediction is a foundational problem in machine learning
- Many flavours/settings (node classification, edge classification, clustering)
- Today: Node classification in the *online learning* setting (sequential prediction)
- Want to develop algorithms with *performance guarantees*

Predicting Switching Graph Labelings

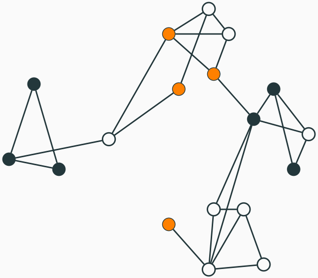
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$



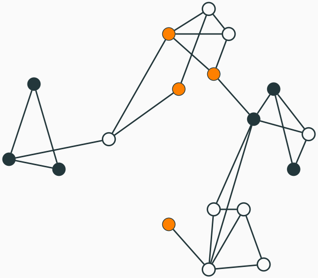
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$



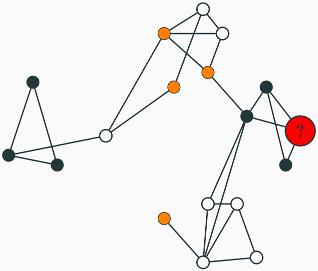
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:



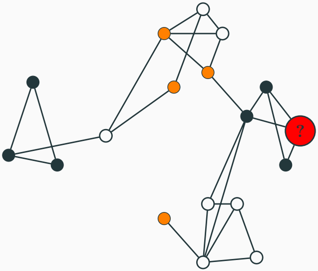
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
  1. Nature selects a vertex $i_t \in V$



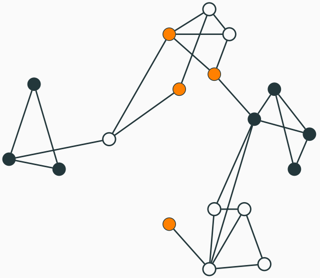
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
  1. Nature selects a vertex $i_t \in V$
  2. Learner predicts $\hat{y}_t \in \{-1, 1\}$



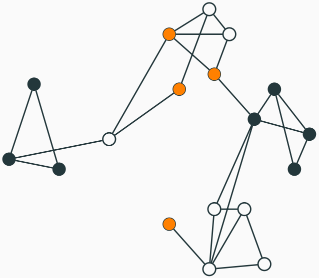
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
  1. Nature selects a vertex $i_t \in V$
  2. Learner predicts $\hat{y}_t \in \{-1, 1\}$
  3. Nature reveals label $\boldsymbol{u}_t(i_t) \in \{-1, 1\}$



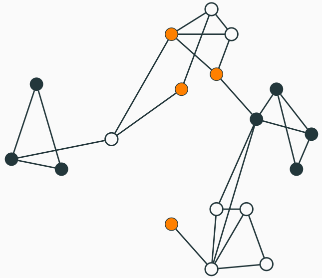
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
  1. Nature selects a vertex $i_t \in V$
  2. Learner predicts $\hat{y}_t \in \{-1, 1\}$
  3. Nature reveals label $\boldsymbol{u}_t(i_t) \in \{-1, 1\}$
  4. Learner incurs loss $m_t = [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$



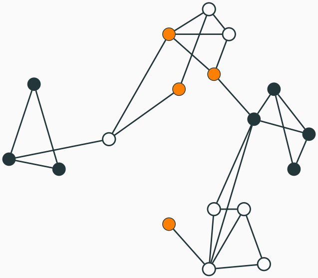
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
  1. Nature selects a vertex $i_t \in V$
  2. Learner predicts $\hat{y}_t \in \{-1, 1\}$
  3. Nature reveals label $\boldsymbol{u}_t(i_t) \in \{-1, 1\}$
  4. Learner incurs loss $m_t = [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$
- No statistical assumptions are made!
  Nature could be adversarial



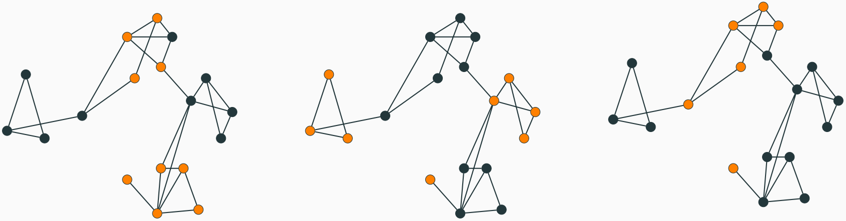
Predicting Graph Labelings Online

- n-vertex Graph $\mathcal{G} = (V, E)$,
  $V = \{1, \ldots, n\}$
- A labeling is a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$
- Online learning protocol:
  For $t = 1, \ldots, T$ **do**:
    1. Nature selects a vertex $i_t \in V$
    2. Learner predicts $\hat{y}_t \in \{-1, 1\}$
    3. Nature reveals label $\boldsymbol{u}_t(i_t) \in \{-1, 1\}$
    4. Learner incurs loss $m_t = [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$
- No statistical assumptions are made!
  Nature could be adversarial
- Performance guarantees hold in the worst case



Switching Graph Labelings

Sequence of labelings $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_T$ s.t. $|\{t : \boldsymbol{u}_t \neq \boldsymbol{u}_{t+1}\}| = K$



$t \rightarrow$ $\quad t = 1 \ldots$ $\quad \ldots t = 7 \ldots$ $\quad \ldots t = 20 \ldots$

The learner **doesn't** know when switches occur

Assume K is ‘small’

Objectives

- Minimize the number of mistakes

$$M = \sum_{t=1}^{T} m_t = \sum_{t=1}^{T} [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$$

Objectives

- Minimize the number of mistakes

$$M = \sum_{t=1}^{T} m_t = \sum_{t=1}^{T} [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$$

- Provide *good* mistake bound guarantees for *switching*:

$$M \leq f\left(\textit{complexity}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T), K, \textit{structure}(\mathcal{G})\right)$$

Objectives

- Minimize the number of mistakes

$$M = \sum_{t=1}^{T} m_t = \sum_{t=1}^{T} [\boldsymbol{u}_t(i_t) \neq \hat{y}_t]$$

- Provide *good* mistake bound guarantees for *switching*:

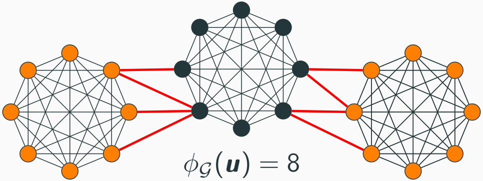
$$M \leq f\left(\mathit{complexity}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T), K, \mathit{structure}(\mathcal{G})\right)$$

- Algorithms should be *fast* (online predictions)

$complexity(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - **Cut-size** ϕ

We assume that a graph $\mathcal{G}$ consists of tightly-connected clusters, with loose inter-cluster connections. Nodes in a cluster (mostly) share the same label.

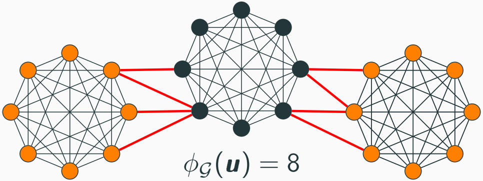
A labeling $\boldsymbol{u} : V \mapsto \{-1, 1\}$ induces a *cut* $\phi_{\mathcal{G}}(\boldsymbol{u}) = \sum_{(i,j)\in E} [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)]$



complexity$(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Cut-size ϕ

We assume that a graph $\mathcal{G}$ consists of tightly-connected clusters, with loose inter-cluster connections. Nodes in a cluster (mostly) share the same label.

A labeling $\boldsymbol{u} : V \mapsto \{-1, 1\}$ induces a *cut* $\phi_{\mathcal{G}}(\boldsymbol{u}) = \sum_{(i,j) \in E} [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)]$

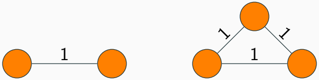


Static mistake bounds typically scale *linearly* with $\phi_{\mathcal{G}}(\boldsymbol{u})$ - **sensitive!**

- [HLP08] - $\mathcal{O}\left(\phi_{\mathcal{G}}(\boldsymbol{u}) \log \frac{n}{\phi_{\mathcal{G}}(\boldsymbol{u})} + \phi_{\mathcal{G}}(\boldsymbol{u})\right)$
- [HP06] - $\mathcal{O}\left(\phi_{\mathcal{G}}(\boldsymbol{u}) R_{\mathcal{G}}\right)$, $\quad R_{\mathcal{G}} = f(\mathit{structure}(\mathcal{G}))$

$\textit{complexity}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Effective Resistance $r_{i,j}$

Define $r_{i,j}$ to be the *effective resistance* between nodes i and j when $\mathcal{G}$ is a network of *unit* resistors (edges)



Define $r_{i,j}$ to be the *effective resistance* between nodes i and j when $\mathcal{G}$ is a network of *unit* resistors (edges)

$r_{a,b} = 1$ $\qquad$ $r_{c,d} = \frac{1}{1+\frac{1}{2}} = \frac{2}{3}$

(Kirchoff's laws for resistors in series and parallel)

$r_{i,j}$ is a measure of connectivity - effective resistance between two nodes *decreases* with increased connectivity

$\mathit{complexity}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Effective Resistance $r_{i,j}$

Define $r_{i,j}$ to be the *effective resistance* between nodes i and j when $\mathcal{G}$ is a network of *unit* resistors (edges)

$r_{a,b} = 1$ $\qquad$ $r_{c,d} = \frac{1}{1+\frac{1}{2}} = \frac{2}{3}$

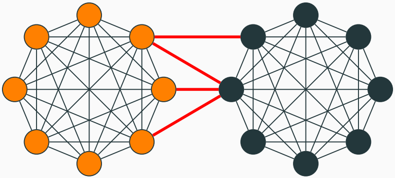
(Kirchoff's laws for resistors in series and parallel)

$r_{i,j}$ is a measure of connectivity - effective resistance between two nodes *decreases* with increased connectivity

Definition
Define the *resistance-weighted cut-size* to be:

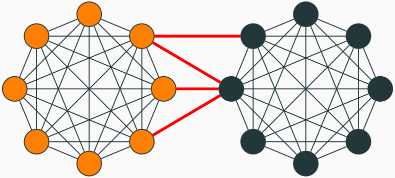
$$\phi^r(\boldsymbol{u}) = \sum_{(i,j)\in E} r_{i,j} \left[\boldsymbol{u}(i) \neq \boldsymbol{u}(j)\right]$$

$\mathit{complexity}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Resistance-weighted cut-size ϕ^r

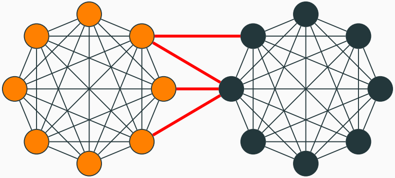


- Two m-cliques with $\ell < m$ edges between them

$complexity(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Resistance-weighted cut-size ϕ^r



- Two m-cliques with $\ell < m$ edges between them
- For all vertices $i, j \in V$, we have $r_{i,j} \leq \Theta(\frac{1}{\ell})$

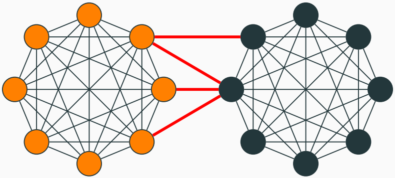


- Two m-cliques with $\ell < m$ edges between them
- For all vertices $i, j \in V$, we have $r_{i,j} \leq \Theta(\frac{1}{\ell})$
- Hence,

$$\phi(\boldsymbol{u}) = \sum_{(i,j) \in E} [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)] = \ell$$

$$\phi^r(\boldsymbol{u}) = \sum_{(i,j) \in E} r_{i,j} \, [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)] \leq \Theta(1)$$

$complexity(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)$ - Resistance-weighted cut-size ϕ^r



- Two m-cliques with $\ell < m$ edges between them
- For all vertices $i, j \in V$, we have $r_{i,j} \leq \Theta(\frac{1}{\ell})$
- Hence,

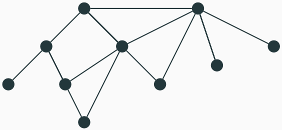
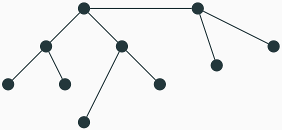
$$\phi(\boldsymbol{u}) = \sum_{(i,j)\in E} [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)] = \ell$$

$$\phi^r(\boldsymbol{u}) = \sum_{(i,j)\in E} r_{i,j} \, [\boldsymbol{u}(i) \neq \boldsymbol{u}(j)] \leq \Theta(1)$$

- $\phi^r(\boldsymbol{u})$ is robust!

Random Spanning Tree - Resistance weighted cut-size

How to exploit $\phi^r(\boldsymbol{u})$?

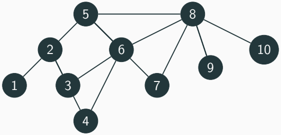
Expected cut-size of a random spanning tree **generated uniformly at random** ([CBGV09]):

$$\begin{aligned}\mathbb{E}\left[\phi_{\mathcal{T}}(\boldsymbol{u})\right] &= \sum_{(i,j)\in E} \mathbb{P}((i,j) \in E_{\mathcal{T}}) \left[\boldsymbol{u}(i) \neq \boldsymbol{u}(j)\right] \\ &= \sum_{(i,j)\in E} r_{i,j} \left[\boldsymbol{u}(i) \neq \boldsymbol{u}(j)\right] \\ &= \phi^r(\boldsymbol{u})\end{aligned}$$

Mistake bounds in terms of $\phi(\boldsymbol{u})$ become *expected* mistake bounds in terms of $\phi^r(\boldsymbol{u})$!

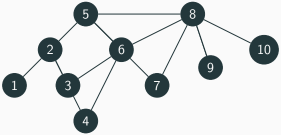
Two Transformations - Trees and Linear Embeddings

Original graph $\mathcal{G}$:

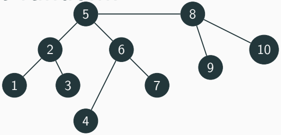


Two Transformations - Trees and Linear Embeddings

Original graph $\mathcal{G}$:

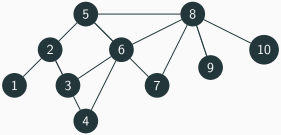


Sample $\mathcal{T}$ uniformly at random:

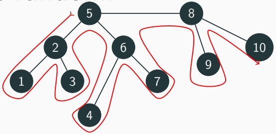


Two Transformations - Trees and Linear Embeddings

Original graph $\mathcal{G}$:



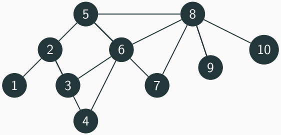
Sample $\mathcal{T}$ uniformly at random:



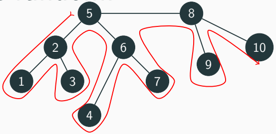
Compute *spine* $\mathcal{S}$ from $\mathcal{T}$ (depth-first search):

Two Transformations - Trees and Linear Embeddings

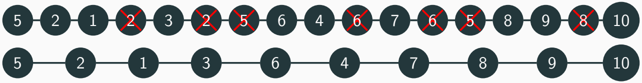
Original graph $\mathcal{G}$:



Sample $\mathcal{T}$ uniformly at random:

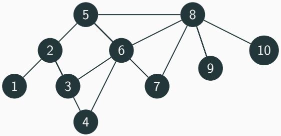


Compute *spine* $\mathcal{S}$ from $\mathcal{T}$ (depth-first search):

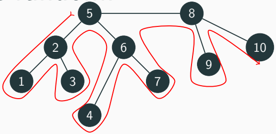


Two Transformations - Trees and Linear Embeddings

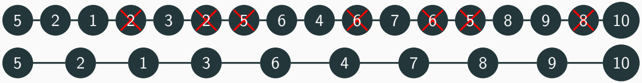
Original graph $\mathcal{G}$:



Sample $\mathcal{T}$ uniformly at random:



Compute *spine* $\mathcal{S}$ from $\mathcal{T}$ (depth-first search):

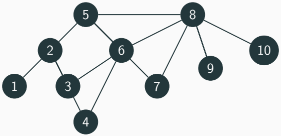


Satisfies:

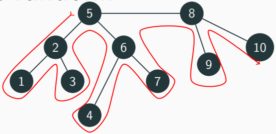
$$\phi_{\mathcal{S}}(\boldsymbol{u}) \leq 2\phi_{\mathcal{T}}(\boldsymbol{u}) \leq 2\phi_{\mathcal{G}}(\boldsymbol{u})$$

Two Transformations - Trees and Linear Embeddings

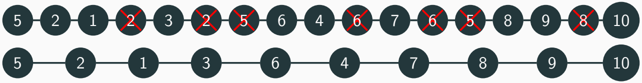
Original graph $\mathcal{G}$:



Sample $\mathcal{T}$ uniformly at random:



Compute *spine* $\mathcal{S}$ from $\mathcal{T}$ (depth-first search):



Properties: ([HLP08])

$$\phi_{\mathcal{S}}(\boldsymbol{u}) \leq 2\phi_{\mathcal{T}}(\boldsymbol{u}) \leq 2\phi_{\mathcal{G}}(\boldsymbol{u})$$

$$\mathbb{E}\left[\phi_{\mathcal{S}}(\boldsymbol{u})\right] \leq 2\mathbb{E}\left[\phi_{\mathcal{T}}(\boldsymbol{u})\right] = 2\phi^{r}(\boldsymbol{u})$$

Cluster Specialists

Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \Box\}$

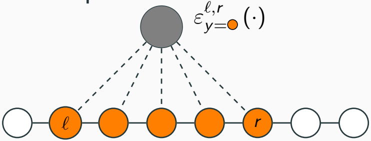
Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction

Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction
- Given $\mathcal{S}$ denote the vertices $\{1, \ldots, n\}$ in linear order
- For a vertex $v \in V$ a cluster specialist predicts:

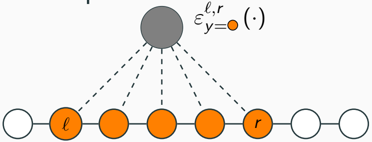
$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$



Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction
- Given $\mathcal{S}$ denote the vertices $\{1, \ldots, n\}$ in linear order
- For a vertex $v \in V$ a cluster specialist predicts:

$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

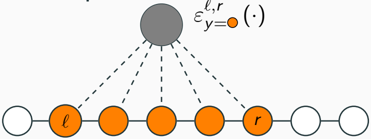


- How to construct a specialist set?

Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction
- Given $\mathcal{S}$ denote the vertices $\{1, \ldots, n\}$ in linear order
- For a vertex $v \in V$ a cluster specialist predicts:

$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

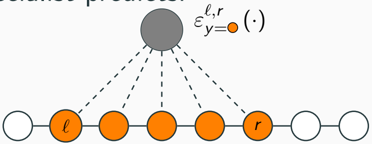


- How to construct a specialist set?
  - Needs to be *complete* (any labeling $\boldsymbol{u} \in \{-1, 1\}^{|V|}$ is *covered*)

Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction
- Given $\mathcal{S}$ denote the vertices $\{1, \ldots, n\}$ in linear order
- For a vertex $v \in V$ a cluster specialist predicts:

$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

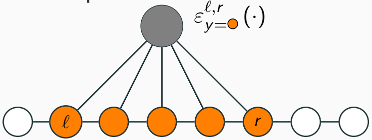


- How to construct a specialist set?
  - Needs to be *complete* (any labeling $\boldsymbol{u} \in \{-1, 1\}^{|V|}$ is *covered*)
  - The 'covering set' of a labeling should **not** be *too large*

Cluster Specialists

- A specialist is a *basis function* $\varepsilon : V \to \{-1, 1, \square\}$
- "$\square$" - on some inputs a specialist can offer no prediction
- Given $\mathcal{S}$ denote the vertices $\{1, \ldots, n\}$ in linear order
- For a vertex $v \in V$ a cluster specialist predicts:

$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \le v \le r \\ \square & \text{otherwise} \end{cases}$$

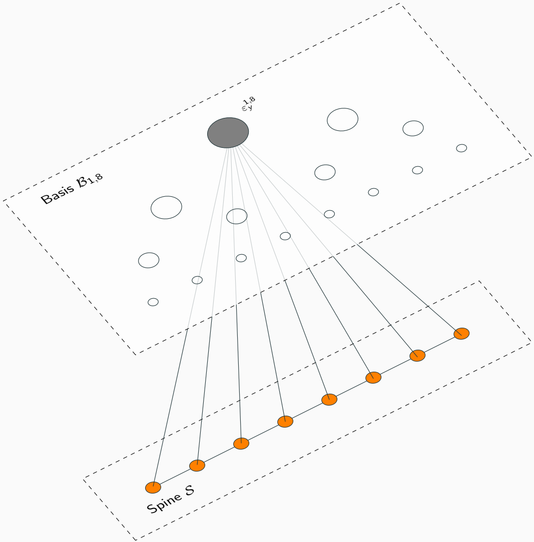


- Two specialist sets:

$$\mathcal{F}_n := \{\varepsilon_y^{\ell,r} : \ell, r \in [n], \ell \le r; y \in \{-1,1\}\}, \qquad |\mathcal{F}_n| = \mathcal{O}(n^2)$$

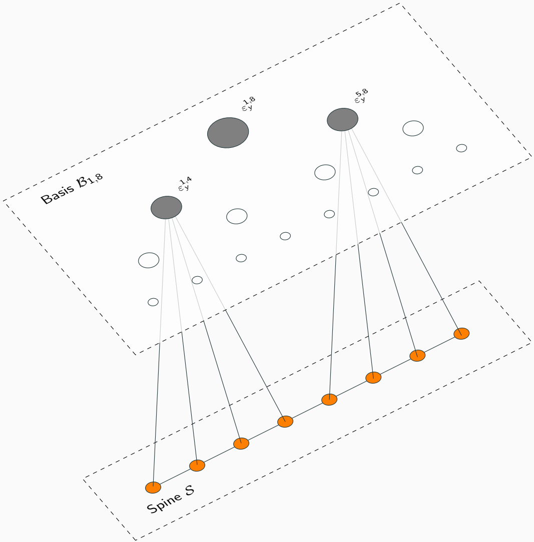
$$\mathcal{B}_{m,n} := \begin{cases} \{\varepsilon_{-1}^{m,n}, \varepsilon_1^{m,n}\} & m = n \\ \{\varepsilon_{-1}^{m,n}, \varepsilon_1^{m,n}\} \cup \mathcal{B}_{m,\lfloor \frac{m+n}{2} \rfloor} \cup \mathcal{B}_{\lceil \frac{m+n}{2} \rceil, n} & m \ne n \end{cases}, \qquad |\mathcal{B}_{1,n}| = \mathcal{O}(n)$$

Basis Set $\mathcal{B}_{1,n}$

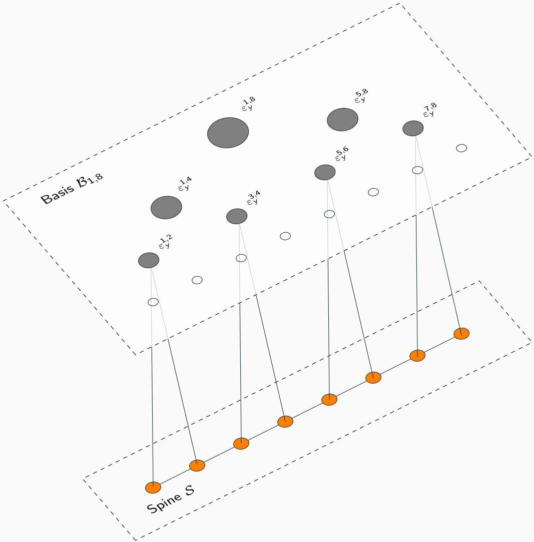


$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

Basis Set $\mathcal{B}_{1,n}$

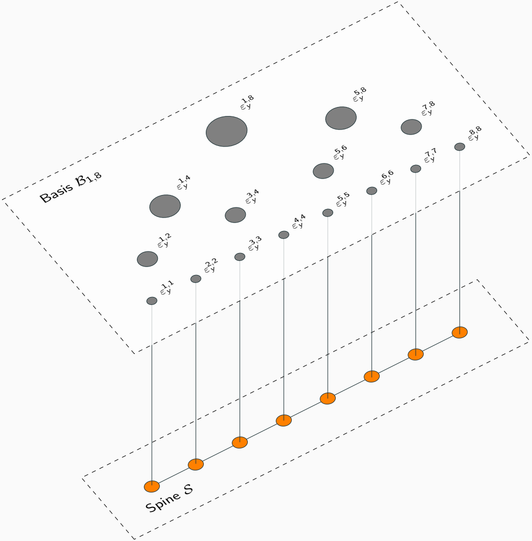


$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \le v \le r \\ \square & \text{otherwise} \end{cases}$$



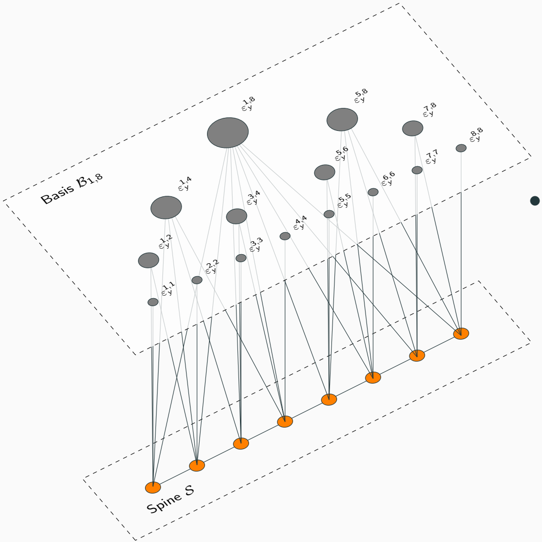
$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

Basis Set $\mathcal{B}_{1,n}$



$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

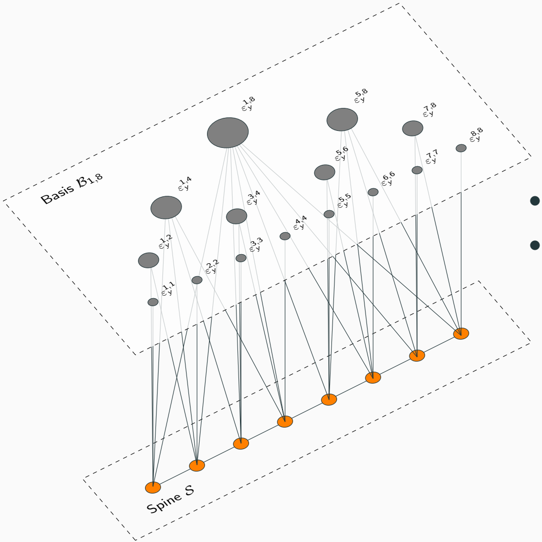
Basis Set $\mathcal{B}_{1,n}$



$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

- $\mathcal{B}_{1,n}$ is **complete**

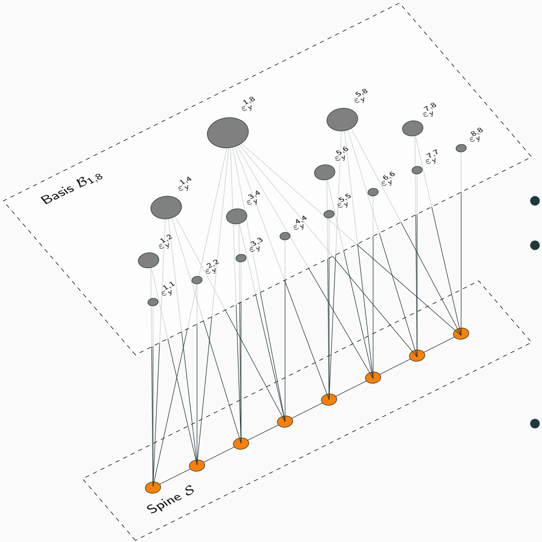
Basis Set $\mathcal{B}_{1,n}$



$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

- $\mathcal{B}_{1,n}$ is **complete**
- Maximum number of specialists required to *cover* a labeling $\boldsymbol{u} \in \{-1,1\}^{|V|}$ is bounded above by $2(\phi_{\mathcal{S}}(\boldsymbol{u})+1)\lceil \log_2 \frac{n}{2} \rceil$

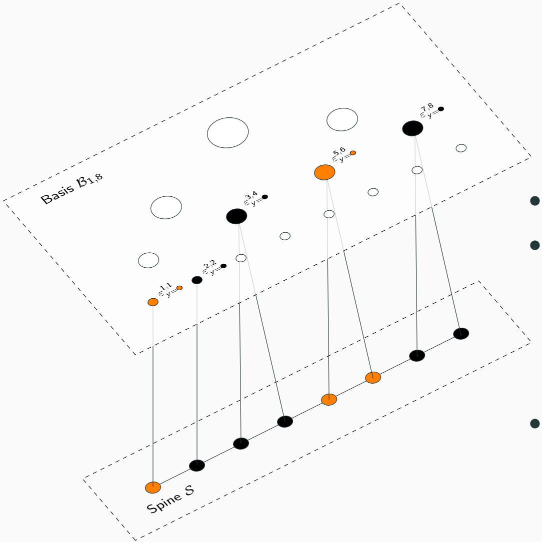
Basis Set $\mathcal{B}_{1,n}$



$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \leq v \leq r \\ \square & \text{otherwise} \end{cases}$$

- $\mathcal{B}_{1,n}$ is **complete**
- Maximum number of specialists required to *cover* a labeling $\boldsymbol{u} \in \{-1,1\}^{|V|}$ is bounded above by $2(\phi_{\mathcal{S}}(\boldsymbol{u})+1)\lceil \log_2 \frac{n}{2} \rceil$
- Only $\Theta(\log n)$ specialists are 'active' at any given time ($\mathcal{O}(n^2)$ for basis set $\mathcal{F}_n$)

Basis Set $\mathcal{B}_{1,n}$



$$\varepsilon_y^{\ell,r}(v) := \begin{cases} y & \ell \le v \le r \\ \square & \text{otherwise} \end{cases}$$

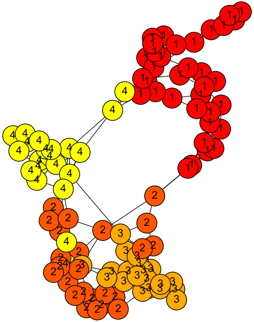
- $\mathcal{B}_{1,n}$ is **complete**
- Maximum number of specialists required to *cover* a labeling $\boldsymbol{u} \in \{-1,1\}^{|V|}$ is bounded above by $2(\phi_{\mathcal{S}}(\boldsymbol{u})+1)\lceil \log_2 \frac{n}{2} \rceil$
- Only $\Theta(\log n)$ specialists are 'active' at any given time ($\mathcal{O}(n^2)$ for basis set $\mathcal{F}_n$)

Specialists Example - USPS

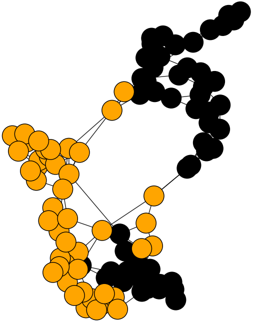
Image Source: [YHW18]

- USPS Dataset (hand-written digits)
- 16×16 pixels $\rightarrow$ points in $\mathbb{R}^{256}$
- Build graph by connecting each point with its 3 nearest neighbors

Specialists Example - USPS

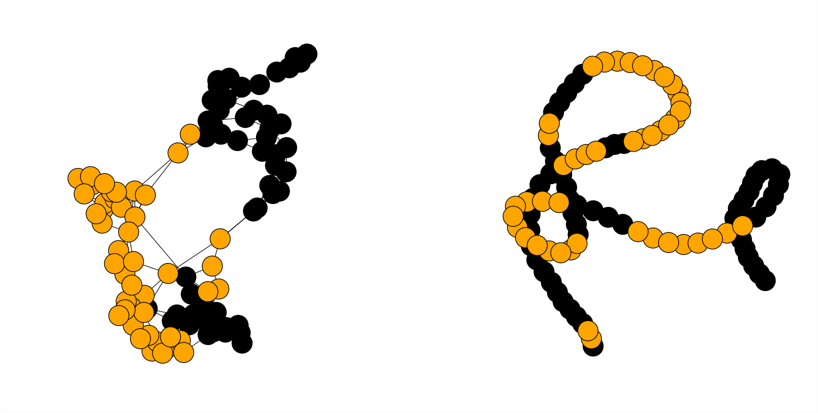


Original Graph $\mathcal{G}$



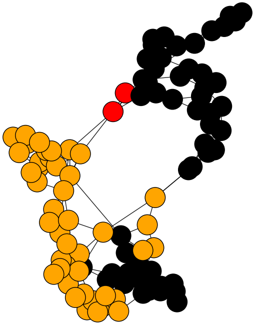
Simulated binary labeling

Specialists Example - USPS

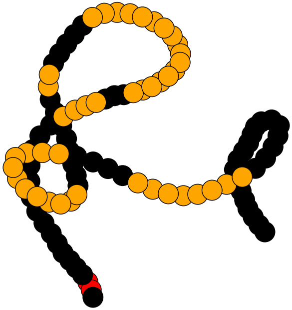


Original Graph $\mathcal{G}$ Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

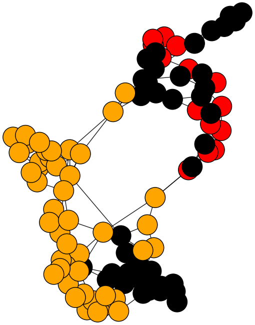


Original Graph $\mathcal{G}$

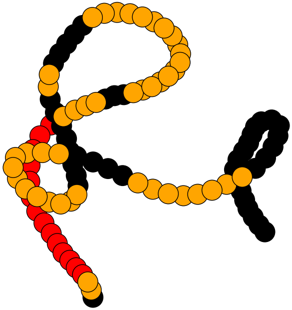


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

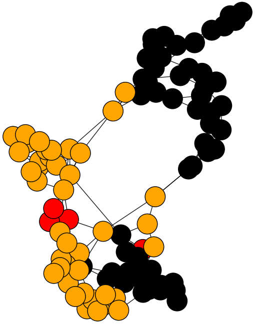


Original Graph $\mathcal{G}$

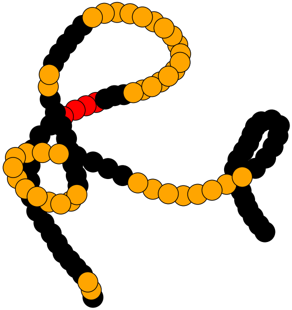


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

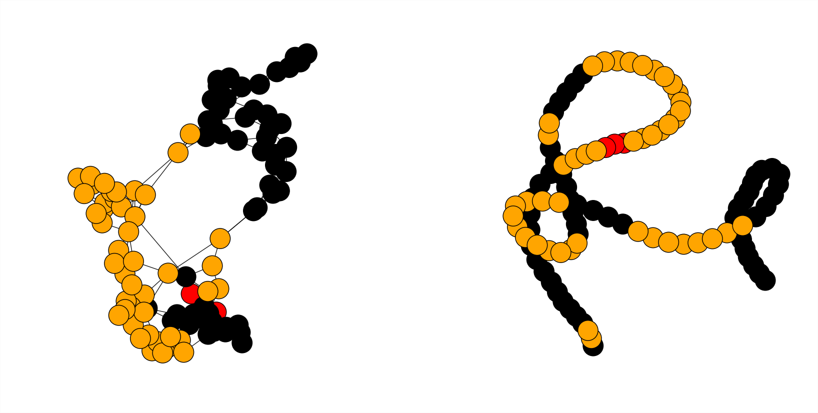


Original Graph $\mathcal{G}$



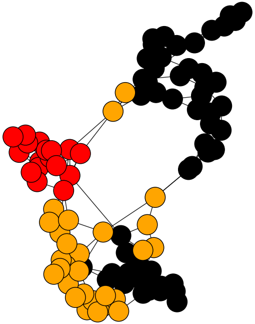
Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

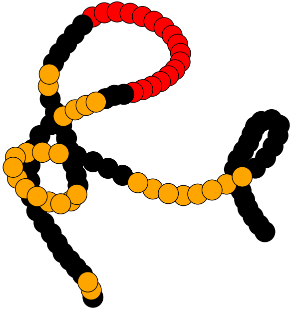


Original Graph $\mathcal{G}$ Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

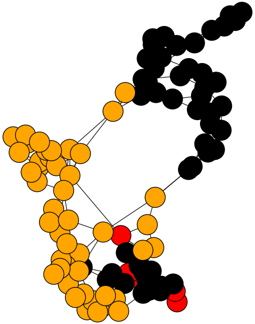


Original Graph $\mathcal{G}$

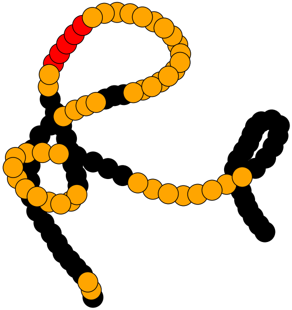


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

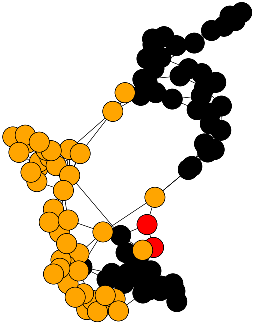


Original Graph $\mathcal{G}$

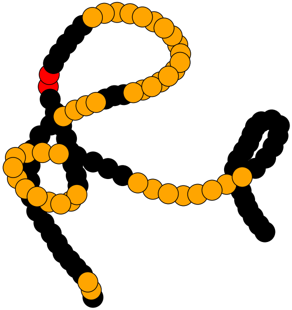


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

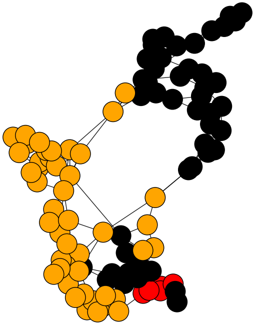


Original Graph $\mathcal{G}$

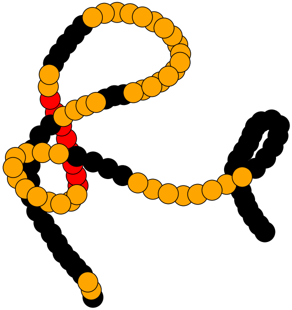


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

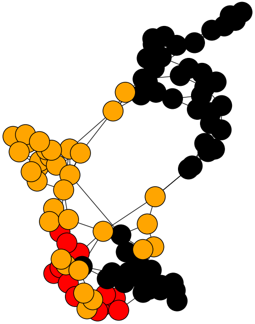


Original Graph $\mathcal{G}$

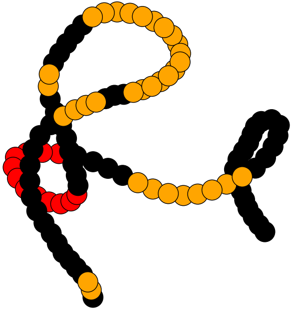


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

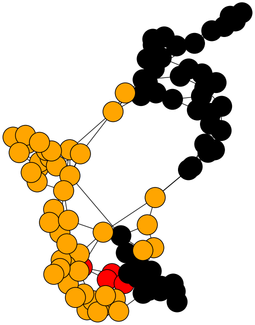


Original Graph $\mathcal{G}$

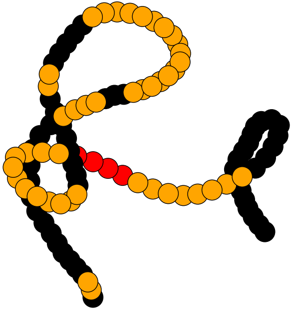


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

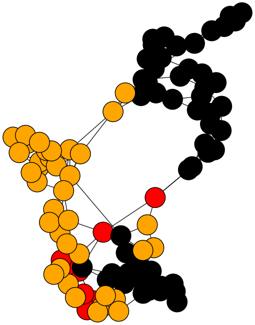


Original Graph $\mathcal{G}$

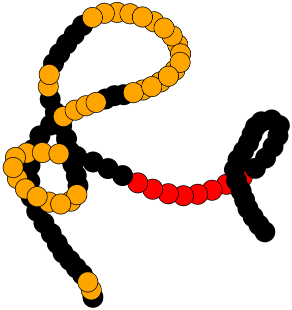


Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS

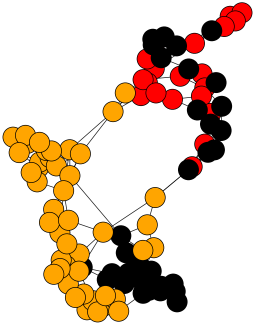


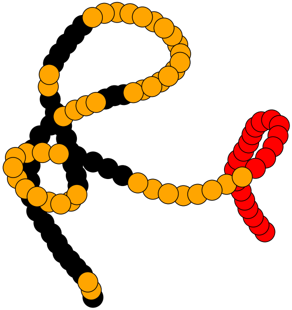
Original Graph $\mathcal{G}$



Linear Embedding (Spine) $\mathcal{S}$

Specialists Example - USPS





Original Graph $\mathcal{G}$

Linear Embedding (Spine) $\mathcal{S}$

Algorithm

Algorithm 1: SWITCHING CLUSTER SPECIALISTS

input : Specialists set $\mathcal{E}$
parameter : $\alpha \in [0, 1]$
initialize : $\boldsymbol{\omega}_1 \leftarrow \frac{1}{|\mathcal{E}|}\mathbf{1}, \dot{\boldsymbol{\omega}}_0 \leftarrow \frac{1}{|\mathcal{E}|}\mathbf{1}, \boldsymbol{p} \leftarrow \mathbf{0}, m \leftarrow 0$

for $t = 1$ **to** T **do**
- **receive** $i_t \in V$
- **set** $\mathcal{A}_t := \{\varepsilon \in \mathcal{E} : \varepsilon(i_t) \neq \square\}$
- **foreach** $\varepsilon \in \mathcal{A}_t$ **do** `// delayed share update`
  $$\omega_{t,\varepsilon} \leftarrow (1-\alpha)^{m-p_\varepsilon}\,\dot{\omega}_{t-1,\varepsilon} + \frac{1-(1-\alpha)^{m-p_\varepsilon}}{|\mathcal{E}|} \tag{1}$$
- **predict** $\hat{y}_t \leftarrow \mathrm{sign}(\sum_{\varepsilon \in \mathcal{A}_t} \omega_{t,\varepsilon}\, \varepsilon(i_t))$
- **receive** $y_t \in \{-1, 1\}$
- **set** $\mathcal{Y}_t := \{\varepsilon \in \mathcal{E} : \varepsilon(i_t) = y_t\}$
- **if** $\hat{y}_t \neq y_t$ **then** `// loss update`
  $$\dot{\omega}_{t,\varepsilon} \leftarrow \begin{cases} 0 & \varepsilon \in \mathcal{A}_t \cap \bar{\mathcal{Y}}_t \\ \dot{\omega}_{t-1,\varepsilon} & \varepsilon \notin \mathcal{A}_t \\ \omega_{t,\varepsilon}\frac{\boldsymbol{\omega}_t(\mathcal{A}_t)}{\boldsymbol{\omega}_t(\mathcal{Y}_t)} & \varepsilon \in \mathcal{Y}_t \end{cases} \tag{2}$$
  - **foreach** $\varepsilon \in \mathcal{A}_t$ **do**
    - $p_\varepsilon \leftarrow m$
  - $m \leftarrow m + 1$
- **else**
  - $\dot{\boldsymbol{\omega}}_t \leftarrow \dot{\boldsymbol{\omega}}_{t-1}$

Algorithm Intuition

- Weight vector $\boldsymbol{\omega}_t \in [0,1]^{|\mathcal{E}|}$ maintained
- Weight $\omega_{t,\varepsilon}$ corresponds to our 'confidence' in specialist ε
- On each trial set "active" specialists
  $\mathcal{A}_t := \{\varepsilon \in \mathcal{E} : \varepsilon(i_t) \neq \square\}$
- Take the weighted-majority vote of specialists in $\mathcal{A}_t$
- Decrease weight of *incorrect* specialists
- Increase weight of *correct* specialists
- Share some of the weight among all specialists after each update (can be done efficiently)

Mistake Bound Guarantees

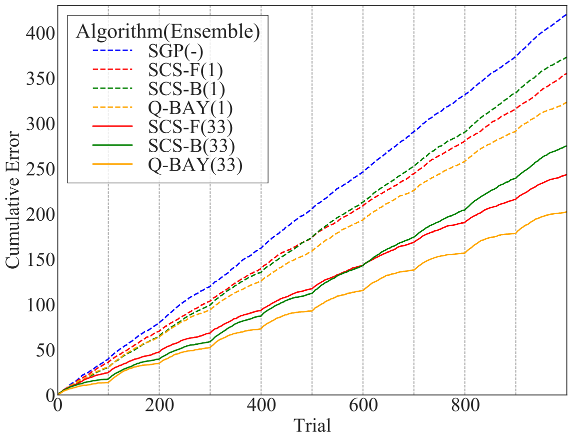
For a sequence of K distinct labelings $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^K$, let

$$H_k := \sum_{(i,j) \in E_{\mathcal{S}}} \Big[\Big[[\boldsymbol{u}^k(i) \neq \boldsymbol{u}^k(j)] \vee [\boldsymbol{u}^{k+1}(i) \neq \boldsymbol{u}^{k+1}(j)] \Big] \wedge \Big[[\boldsymbol{u}^k(i) \neq \boldsymbol{u}^{k+1}(i)] \vee [\boldsymbol{u}^k(j) \neq \boldsymbol{u}^{k+1}(j)] \Big] \Big]$$

	Static Bounds
[HLP08]	$\mathcal{O}\left(\phi_{\mathcal{G}}(\boldsymbol{u}) \log \frac{n}{\phi_{\mathcal{G}}(\boldsymbol{u})} + \phi_{\mathcal{G}}(\boldsymbol{u})\right)$
[HP06]	$\mathcal{O}\left(\phi_{\mathcal{G}}(\boldsymbol{u}) R_{\mathcal{G}}\right)$
	Switching Mistake Bounds
$\mathcal{F}_n$	$\mathcal{O}\left(\phi_{\mathcal{G}}(\boldsymbol{u}_1) \log n + \sum_{k=1}^{K-1} H_k (\log n + \log K + \log\log T)\right)$
$\mathcal{B}_{1,n}$	$\mathcal{O}\left(\left(\phi_{\mathcal{G}}(\boldsymbol{u}_1) \log n + \sum_{k=1}^{K-1} H_k (\log n + \log K + \log\log T)\right) \log n\right)$
	Time Complexity (per trial)
$\mathcal{F}_n$	$\mathcal{O}(n^2)$
$\mathcal{B}_{1,n}$	$\mathcal{O}(\log n)$

Experiments

Experiments



Mean cumulative error over 12 iterations of 10 switches every 100 trials on an 4096-vertex graph. Solid lines SCS-F and SCS-B show the mean cumulative error of an ensemble size of 33, dashed lines show the average cumulative error of a single instance (ensemble size 1).

Conclusion

Conclusion

- Solved the problem of efficient online prediction of switching graph labelings
- Described the machinery of Cluster Specialists
- Proved *smooth* mistake bounds
- Exponential speed up with $\mathcal{B}_{1,n}$
- Future work:
    - New methods of constructing specialist sets (e.g., hierarchical clustering)
    - Further experiments

Thank you!

(Thank you to Fabio Vitale for some slides)

References

N. Cesa-Bianchi, C. Gentile, and F. Vitale, *Fast and optimal prediction on a labeled tree*, Proceedings of the 22nd Annual Conference on Learning Theory, Omnipress, 2009, pp. 145–156.

M. Herbster, G. Lever, and M. Pontil, *Online prediction on large diameter graphs*, Proceedings of the 21st International Conference on Neural Information Processing Systems, NIPS '08, 2008, pp. 649–656.

M. Herbster and M. Pontil, *Prediction on a graph with a perceptron*, Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS'06, 2006, pp. 577–584.

Y. Yang, Y. Hu, and F. Wu, *Sparse and low-rank subspace data clustering with manifold regularization learned by local linear embedding*, Applied Sciences **8** (2018), 2175.