# Online Prediction of Switching Graph Labelings with Cluster Specialists

Mark Herbster and James Robinson
Department of Computer Science, University College London
{m.herbster}{j.robinson}@cs.ucl.ac.uk

## Abstract

We address the problem of predicting the labeling of a graph in an online setting when the labeling is changing over time. We present an algorithm based on a *specialist* approach; we develop the machinery of cluster specialists which probabilistically exploits the cluster structure in the graph. Our algorithm has two variants, one of which surprisingly only requires $\mathcal{O}(\log n)$ time on any trial $t$ on an $n$-vertex graph, an exponential speed up over existing methods. We prove switching mistake-bound guarantees for both variants of our algorithm. Furthermore these mistake bounds *smoothly* vary with the magnitude of the change between successive labelings. We perform experiments on Chicago Divvy Bicycle Sharing data and show that our algorithms significantly outperform an existing algorithm (a kernelized Perceptron) as well as several natural benchmarks.

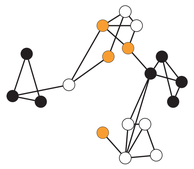
## Main Contributions

- Mistake-bounded adaptive algorithm for switching graph prediction
- *Fast* algorithm - $\mathcal{O}(\log n)$ per-trial time complexity
- *Smooth* switching - mistake bounds scale smoothly with the sequence of labelings

## Online Graph Prediction

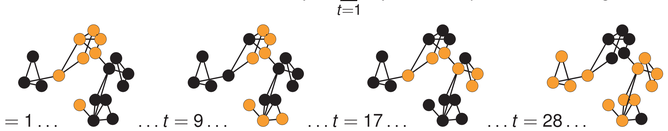
Given an undirected, connected graph $\mathcal{G} = (V, E)$, learn a function $\boldsymbol{u} : V \mapsto \{-1, 1\}$.
**For** $t = 1, \ldots, T$ **do**:

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 $m_t = [\boldsymbol{u}_t(i_t) \neq \hat{y}_t] \in \{0, 1\}$



## Switching Graph Labelings

**Goal**: Minimize the number of *total* mistakes $M_t := \sum_{t=1}^{T} m_t$ over a *sequence* of labelings.

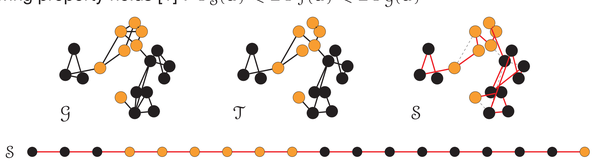



Define $K := \{t \in \{2, \ldots, T\} : \boldsymbol{u}_t \neq \boldsymbol{u}_{t-1}\} \cup \{1\}$, (e.g., $K = \{1, 9, 17, 28\}$).

## Random Spanning Trees and Line Graphs

Define the cut-size of a labeling $\boldsymbol{u}$ on $\mathcal{G}$ to be $\Phi_{\mathcal{G}}(\boldsymbol{u}) := |\{(i, j) \in E : u_i \neq u_j\}|$.
Transform graph $\mathcal{G} \to \mathcal{T} \to \mathcal{S}$:

- Random Spanning Tree $\mathcal{T}$ sampled *uniformly at random*
- Line graph $\mathcal{S}$ sampled from $\mathcal{T}$ using *depth-first search*
- The following property holds [1] : $\Phi_{\mathcal{S}}(\boldsymbol{u}) \leqslant 2\Phi_{\mathcal{T}}(\boldsymbol{u}) \leqslant 2\Phi_{\mathcal{G}}(\boldsymbol{u})$

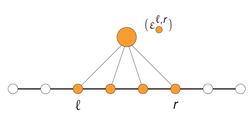



Segments on $\mathcal{S}$ probabilistically correspond to clusters in $\mathcal{G}$!

## Cluster Specialists

Define a **cluster specialist** $\varepsilon_y^{\ell,r} : V \to \{-1, 1, \square\}$

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




- $\varepsilon_y^{\ell,r}$ is a basis function specialized to a segment on $\mathcal{S}$, which roughly corresponds to a cluster in $\mathcal{G}$
- A specialists set, $\mathcal{E}$, should be *complete* (any labeling $\boldsymbol{u} \in \{-1, 1\}^{|V|}$ is covered)
- The smallest subset of specialists required to cover $\boldsymbol{u}$ should not be too large

## SCS Algorithm

**input** : Specialists set $\mathcal{E}$
**parameter** : $\alpha \in [0, 1]$
**initialize** : $\omega_1 \leftarrow \frac{1}{|\mathcal{E}|}\mathbf{1}$, $\dot{\omega}_0 \leftarrow \frac{1}{|\mathcal{E}|}\mathbf{1}$, $\boldsymbol{p} \leftarrow \mathbf{0}$, $m \leftarrow 0$
**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`

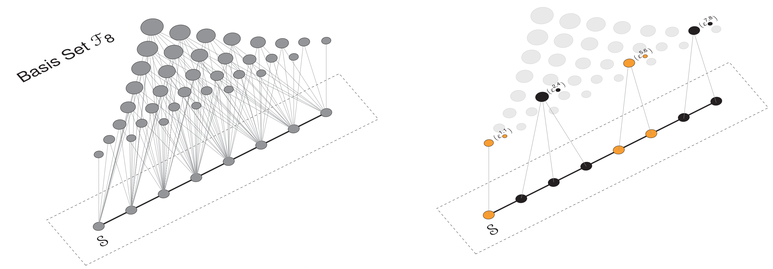
$$\omega_{t,\varepsilon} \leftarrow (1-\alpha)^{m-p_\varepsilon}\,\dot{\omega}_{t-1,\varepsilon} + \frac{1-(1-\alpha)^{m-p_\varepsilon}}{|\mathcal{E}|} \quad (1)$$

  **predict** $\hat{y}_t \leftarrow \operatorname{sign}\left(\sum_{\varepsilon \in \mathcal{A}_t} \omega_{t,\varepsilon}\, \varepsilon(i_t)\right)$
  **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`

$$\dot{\omega}_{t,\varepsilon} \leftarrow \begin{cases} 0 & \varepsilon \in \mathcal{A}_t \cap \mathcal{Y}_t \\ \dot{\omega}_{t-1,\varepsilon} & \varepsilon \notin \mathcal{A}_t \\ \omega_{t,\varepsilon}\frac{\sum_{\varepsilon \in \mathcal{A}_t} \omega_{t,\varepsilon}}{\sum_{\varepsilon \in \mathcal{Y}_t} \omega_{t,\varepsilon}} & \varepsilon \in \mathcal{Y}_t \end{cases} \quad (2)$$

    **foreach** $\varepsilon \in \mathcal{A}_t$ **do**
      $p_\varepsilon \leftarrow m$
    $m \leftarrow m + 1$
  **else**
    $\dot{\boldsymbol{\omega}}_t \leftarrow \dot{\boldsymbol{\omega}}_{t-1}$

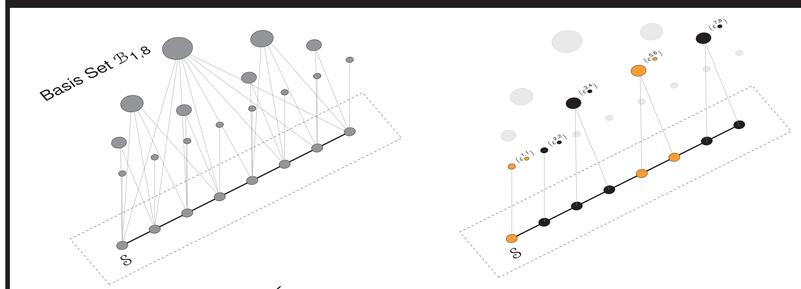
## Basis Set $\mathcal{F}_n$




$$\mathcal{F}_n := \{\varepsilon_y^{\ell,r} : \ell, r \in [n], \ell \leqslant r; y \in \{-1, 1\}\}$$

- Over-complete basis set
- $|\mathcal{F}_n| = \mathcal{O}(n^2)$
- $\mathcal{O}(n^2)$ specialists active per trial
- Minimum number of specialists required to cover a labeling $\boldsymbol{u} \in \{-1, 1\}^{|V|}$ is always $\Phi_{\mathcal{S}}(\boldsymbol{u}) + 1$
- Stronger bound than $\mathcal{B}_{1,n}$, but ***slower***

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




$$\mathcal{B}_{p,q} := \begin{cases} \{\varepsilon_{-1}^{p,q}, \varepsilon_{1}^{p,q}\} & p = q, \\ \{\varepsilon_{-1}^{p,q}, \varepsilon_{1}^{p,q}\} \cup \mathcal{B}_{p,\lfloor\frac{p+q}{2}\rfloor} \cup \mathcal{B}_{\lfloor\frac{p+q}{2}\rfloor+1,q} & p \neq q\,. \end{cases}$$

- Hierarchical basis set analogous to a binary tree
- $|\mathcal{B}_n| = \mathcal{O}(n)$
- Only $\mathcal{O}(\log n)$ specialists active per trial
- Minimum number of specialists required to cover a labeling $\boldsymbol{u} \in \{-1, 1\}^{|V|}$ is bounded above by $2(\Phi_{\mathcal{S}}(\boldsymbol{u}) + 1)\lceil \log_2 \frac{n}{2} \rceil$
- Weaker bound than $\mathcal{F}_n$, but ***exponentially faster***

## Mistake Bounds

For a connected $n$-vertex graph $\mathcal{G}$ and with randomly sampled line graph $\mathcal{S}$, the number of mistakes made in predicting the online sequence $(i_1, y_1), \ldots, (i_T, y_T)$ by the SCS algorithm with an optimally-tuned parameter $\alpha$ is upper bounded with basis $\mathcal{F}_n$ by

$$\mathcal{O}\left(\Phi_1 \log n + \sum_{i=1}^{|K|-1} \left|\!\left|\!\left| H(\boldsymbol{u}_{k_i}, \boldsymbol{u}_{k_{i+1}}) \right|\!\right|\!\right| (\log n + \log|K| + \log\log T)\right)$$

and with basis $\mathcal{B}_n$ by

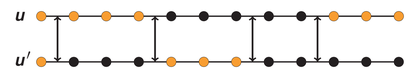
$$\mathcal{O}\left(\left(\Phi_1 \log n + \sum_{i=i}^{|K|-1} \left|\!\left|\!\left| H(\boldsymbol{u}_{k_i}, \boldsymbol{u}_{k_{i+1}}) \right|\!\right|\!\right| (\log n + \log|K| + \log\log T)\right) \log n\right)$$

for any sequence of labelings $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T \in \{-1, 1\}^n$ such that $u_{t,i_t} = y_t$ for all $t \in [T]$.

## Smooth Switching

Mistake bounds scale with the quantity $\sum_{i=1}^{|K|-1} \left|\!\left|\!\left| H(\boldsymbol{u}_{k_i}, \boldsymbol{u}_{k_{i+1}}) \right|\!\right|\!\right|$, where

$$\left|\!\left|\!\left| H(\boldsymbol{u}, \boldsymbol{u}') \right|\!\right|\!\right| := \sum_{(i,j) \in E_{\mathcal{S}}} [\,[[u_i \neq u_j] \vee [u'_i \neq u'_j]] \;\wedge\; [[u_i \neq u'_i] \vee [u_j \neq u'_j]]\,]\,.$$




$\left|\!\left|\!\left| H(\boldsymbol{u}, \boldsymbol{u}') \right|\!\right|\!\right| \leqslant 2\|\boldsymbol{u} - \boldsymbol{u}'\|$, and is often significantly smaller.

## Experiments

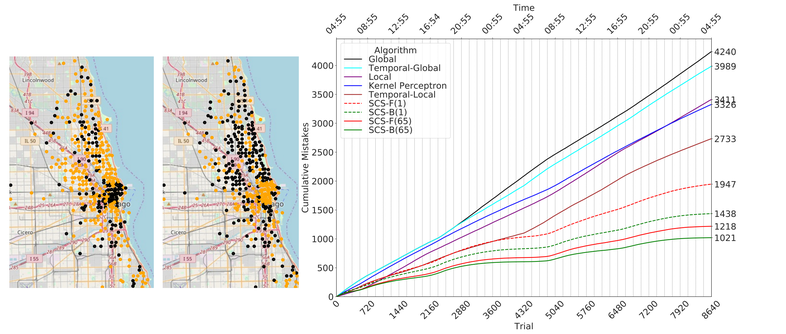



Figure: Two snapshots of labelings of bicycle-sharing stations in Chicago observed at two different times in a 24 hour period (left). Mean cumulative mistakes over 25 iterations for our algorithms, a kernelized perceptron, and several natural benchmarks over 48 hours (right).

- Experiments were performed on Chicago Divvy Bicycle Sharing data
- Nodes were bicycle stations, the labels to be predicted were "*mostly full*" and "*mostly empty*"
- A 404-vertex graph was built using the union of a $k$-nearest neighbor graph ($k = 3$) and a minimum spanning tree
- 8640 nodes were predicted over 72 hours of data
- Our algorithms significantly outperformed a kernel Perceptron algorithm as well as several natural benchmarks
- Using ensembles of independently drawn random spanning trees significantly improved performance (ensembles of size 1 and 65 shown above)

## References

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