Online Prediction of Switching Graph Labelings with Cluster Specialists

Mark Herbster and James Robinson
Department of Computer Science, University College London

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 structure in the graph. Our algorithm has two variants, one of which surprisingly only requires $O(\log n)$ time on any trial $t$ and 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 - $O(\log n)$ per time complexity
- Smooth switching - mistake bounds scale smoothly with the sequence of labelings

Online Graph Prediction
Given an undirected, connected graph $S = (V, E)$, learn a function $u : V \rightarrow \{-1, 1\}$.

Switching Graph Labelings

Basis Set $J_g$

**Goal:** Minimize the number of total mistakes $M_T = \sum_t m_t$ over a sequence of labelings.

**Switching Graph Labelings**

**Random Spanning Trees and Line Graphs**

Define the cut-size of a labeling $u$ on $S$ to be $\Phi_1(u) = |\{i \in E : u_i \neq u_j\}|$.

Transform graph $S$ to $T$ and $S$:

- Random Spanning Tree $T$ sampled uniformly at random
- Line graph $\delta$ sampled from $T$ using depth-first search
- The following property holds: $|J_1| = \Phi_2(u) \leq 20\phi(u) \leq 20\Phi_1(u)$

Segments on $\delta$ probabilistically correspond to clusters in $S$

Clustering Specialists

Define a cluster specialist $j_g : V \rightarrow \{-1, 1, 0\}$

**Basis Set $J_{1,n}$**

**SCS Algorithm**

```
Input
- Specialists set $\mathcal{C}$
- Parameter $\varepsilon > 0$

Initialize
- $M_T = 0$, $\delta, \gamma = 0$, $\varepsilon = 0$, $p, q, r = 0$

for $t = 1$ to $\Theta$ do
    receive $y \in V$
    set $J_g = \{u \in E : c(u) \neq y\}$
    foreach $\phi \in J_g$ do
        // delayed share update
        $\phi_i = \phi_i - \frac{1}{n} \sum_{j \neq i} \varepsilon_j$.
        predict $y_i = \sum_{c \in J_g} \varepsilon_{i,c} \phi_c$
        receive $y_i \in \{-1, 1\}$
        if $y_i \neq y_i$ then
            // loss update
            $\delta_j = \delta_j + \varepsilon_i$
            $\gamma_i = \gamma_i + \varepsilon_i$
            $\varepsilon_i = \varepsilon_i + \varepsilon_i$
        end if
    end foreach
end for
```

**Mistake Bounds**

For a connected $n$-vertex graph $S$ and with randomly sampled line graph $\delta$, the number of mistakes made in predicting the online sequence $(y_t)_{t=1}^\Theta$ by the SCS algorithm with an optimally-tuned parameter $\varepsilon$ is upper bounded by $\Theta_\delta$ by

$$\Theta_\delta \leq \delta \sum_{t=1}^\Theta \log n + \log |S| + \log \log T$$

and with basis $\delta_j$ by

$$\Theta_\delta \leq \delta \sum_{t=1}^\Theta \log n + \log |S| + \log \log T$$

for any sequence of labelings $y_1, ..., y_\Theta \in \{-1, 1\}$ such that $y_{t+1} \neq y_t$ for all $t \in [\Theta]$.

**Smooth Switching**

Mistake bounds scale with the quantity $\|H(u, u_t)\|_{\infty} = \sum_{i,j} |H_{i,j}(u_t, u)_{i,j}|$, where $H(u, u_t) = \sum_{i,j} |H_{i,j}(u_t, u)_{i,j} \wedge |H_{i,j}(u, u_t)| \wedge \overline{|H_{i,j}(u, u_t)|} \wedge |\overline{H_{i,j}(u, u_t)}|$. $\|H(u, u_t)\|_{\infty} \leq 2\|u - u_t\|_{\infty}$, 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 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 outperform 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