Online Graph Prediction with Random Trees

Fabio Vitale
Università di Milano
fabio.vitale@unimi.it

Joint work with
Nicolò Cesa-Bianchi, Claudio Gentile
Classifying the nodes of a known graph

- Classification based only on graphical information (no side information)
- Online learning on a **known** connected undirected graph $G(V,E)$
- **Binary** labels
Online Protocol
Performance measurement

- At each time step \( t \)
  1) Adversary asks for the label of any node \( i_t \)
  2) Learner predicts the label of \( i_t \)
  3) Learner observes the label of \( i_t \)

- Algorithm performances = \#[mistakes]
- Linked entities tend to belong to the same class
- $y \in \{-1,+1\}^n$ is an unknown assignment of binary labels to $V$
- Labeling $y$ induces a cut in $G$ whose size is $\Phi_G(y)$
Random spanning trees

- **Tree-based approximations** of graphs give good bounds and fast algorithms – see e.g.

  M. Herbster, M. Pontil, S. Rojas Galeano - NIPS 2008
  M. Herbster, G. Lever, and M. Pontil - NIPS 2008

- **Uniformly generated random spanning trees** can be built via **random walk** on G

  Expected time $= O(n \log n)$ for most graphs
Random spanning trees

Advantages

- Random spanning tree **robust** against adversarial assignment of labels

- In the worst case $\Phi_T(y) \approx \Phi_G(y)$ for any given tree $T$

- When the graph is dense

$$E\Phi_T(y) = O(\Phi_G(y) / n)$$
Predicting on a labeled tree

- **Main intuition**
  
  Find a (local) partial labeling minimizing the current cutsize and use a NN method.

- **Computationally efficient**
  
  Worst-case time per trial = $O(n)$
  
  Space = $O(n)$

- **Modular**
  
  Can be combined with other methods for obtaining a tree from $G$.
Predicting on a labeled tree (2)

- Updating connected **partial covering** of $T$ made up of disjoint subtrees

- Def.: **Lb-tree** (Label-bordered tree) = Maximal subtree having all and only terminal nodes with a **revealed** label
Lb-trees

lb-trees:

forks:

?
Prediction rule

- Fork label estimation procedure
- Fork label estimation + NN method
Mistake bound

Def.: \textbf{Cluster} = maximal subtree without \( \Phi \)-edges

Def.: \( D = \max \text{ diameter of clusters} \)

\[# \text{mistakes in } T \# = O(\Phi_T(y) \log D) \leq \log(\text{diameter of } T) \]

Thank you