Short proofs for online multiclass prediction on graphs

Jittat Fakcharoenphol[a,*,1], Boonserm Kijsirikul[b]

[a] *Department of Computer Engineering, Kasetsart University, Bangkok 10900, Thailand.*
[b] *Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand.*

Abstract

We present short proofs on the mistake bounds of the 1-nearest neighbor algorithm on an online prediction problem of path labels. The algorithm is one of key ingredients in the algorithm by Herbster, Lever, and Pontil for general graphs. Our proofs are combinatorial and naturally show that the algorithm works when the set of labels is not binary.

Key words: On-line learning, prediction on graphs

1. Introduction

We consider the online prediction of graph labels which can be described briefly as follows. There is a graph $G = (V, E)$ with a fixed node labelling $\mathbf{u} : V \to L$ for some label set L; initially, the algorithm does not have any information of $\mathbf{u}$. The learning process proceeds in rounds. For each round t, Nature asks for a label of node $q_t \in V$. The algorithm predicts the label $\hat{u}(q_t)$ and later receives the true label $\mathbf{u}(q_t)$. It makes a mistake when $\hat{u}(q_t) \neq \mathbf{u}(q_t)$. The goal is to minimize the number of times the algorithm makes mistakes. For motivation and applications of the problem, see, e.g., [1, 6].

The performance of the algorithm is measured against the number of cut edges on the partition of graph nodes induced by the true labeling. We denote by $\Phi_G(\mathbf{u})$ the number of edges in G whose labels on both ends are different.

The recent result of Herbster, Lever, and Pontil [6] gives an efficient algorithm with a mistake bound of

$$2\Phi_G(\mathbf{u}) \max\left[0, \log_2\left(\frac{n-1}{2\Phi_G(\mathbf{u})}\right)\right] + \frac{2\Phi_G(\mathbf{u})}{\ln 2} + 1.$$

They first embed G into a path graph S, called a *spine of* G, then they use the 1-nearest neighbor (1-NN) algorithm for label prediction. The first step only incurs a factor of 2 on the cut size ($\Phi_G(\mathbf{u})$); their mistake bound follows from the proof of the second step.

Herbster *et al.*'s proof of the mistake bound of the 1-NN algorithm is based on the result on the Halving algorithm [9]. To do so, they define a probability distribution over the possible hypotheses so that the Halving algorithm implements the 1-NN algorithm.

In this manuscript we give a short combinatorial proof that the 1-NN algorithm on an n-node path with k cut edges makes at most $O(k \log(n/k) + k)$ mistakes. This bound is off by a constant factor from the bound in [6]. We also show that with a more careful analysis, this bound can be improved to almost match the bound of [6].

Apart of being very short and combinatorial, another nice property of our proof is that the bound does not depend on the set of labels. Therefore, they also imply that the algorithm of Herbster *et al.* also works when labels are not binary.

We present our proofs in Section 2. The next section reviews other closely related work.

*Corresponding author
Email addresses: jittat@gmail.com (Jittat Fakcharoenphol), Boonserm.K@chula.ac.th (Boonserm Kijsirikul)
[1] Supported by the Thailand Research Fund Grant MRG5080318.

1.1. Other related work

Early works [8, 7] on graph prediction use algorithms based on the Perceptron algorithm using pseudoinverse of graph Laplacian as a kernel and provide mistake bounds that depend on the cut size and the largest effective resistance between any pair of vertices in the graph. Herbster [4] exploits the cluster structure of the labeling on the graph, and provides an improved mistake bounds. However, there is an example by [6] that shows that the algorithm based on this approach may make $\Theta(\sqrt{n})$ on some n-node graph. Recently, Herbster and Lever [5] explore another class of seminorms, called Laplacian p-seminorms, and show that with the right setting of p (which depends only on the graph) the mistake bound is logarithmic.

Recent work of Cesa-Bianchi, Gentile, and Vitale [1] presents an algorithm for prediction on trees whose worst-case number of mistakes over all labelling and all query sequence is optimal up to a constant factor.

We also note that our work in this paper stems from the proof of a slightly weaker bound appeared in [3] based on result in [2].

2. The 1-NN algorithm on paths

We are given a line graph $G = (V, E)$; let $n = |V|$. Without loss of generality, we label nodes in G as $1, 2, \ldots, n$, where nodes 1 and n are the only two degree-1 nodes, and there is an edge $(i, i+1) \in E$, for all $1 \leq i < n$.

The online prediction problem proceeds in rounds. Each round t, when Nature asks for a label of node q_t, the 1-NN algorithm finds the closest node s whose true label is known and returns s's label. For that round, we call q_t the query node and s the source node. Later on, when the true label of i is revealed, the algorithm updates i's label on the graph. If the predicted label of q_t is not the same as the revealed label, the algorithm makes a mistake.

In our analysis, a *distance* from a given node i to another node j is the number of edges on the unique path from i to j, i.e., it is $|i - j|$. A *distance from i to edge* $(j, j+1)$ is the minimum of the distances from i to j or from i to $j+1$, i.e., it is $\min(|i-j|, |i-j-1|)$.

We first present a simpler theorem that shows the same asymptotic bound but with a higher constant.

Theorem 1. *The 1-NN algorithm makes at most $O(k + k \log(n/k))$ mistakes where $n = |V|$ and k is the number cut edges.*

Proof: First assume that $k < n - 1$, otherwise the bound holds trivially.

Denote all cut edges by $e_1, e_2, \ldots, e_k$; we assume that they are ordered by the smaller indices of their end points. These cut edges partition G into $k+1$ connected subgraphs. Call them $C_0, C_1, \ldots, C_k$; note that each edge e_i is adjacent to C_{i-1} and C_i.

We start our analysis after the first mistake is made.

For each mistake the algorithm makes after that, we have that the true labels of query node i and source node s are different. Therefore, there exists some cut edge along the unique path P from i to s. We charge this mistake to the closest cut edge on P from i.

To see that each edge e_i is charged by at most $1 + \log |C_i|$ times by nodes in C_i, consider the sequence of nodes that charge to e_i: $v_1, v_2, \ldots$. For $j > 1$, in order for v_j to make a mistake, e_i must be closer to v_j than all other known nodes, including v_{j-1}. Thus, we have that the distance from a node in C_i charging to e_i decreases by at least a factor of 2 each time e_i is charged. Thus, e_i can be charged at most $1 + \log |C_i|$ times by nodes in C_i. We can use the same argument to show that e_i is charged by at most $1 + \log |C_{i-1}|$ times from nodes in C_{i-1}.

Note that only mistakes on nodes in C_{i-1} or C_i can be charged to e_i. Therefore e_i is charged by at most $2 + \log |C_{i-1}| + \log |C_i|$ times.

Summing over all cut edges, the number of mistakes charged to any cut edge is at most $2k + \sum_{i=0}^{k} 2 \log |C_i|$. Since $\sum_{i=0}^{k} |C_i| = n$, the number of mistakes maximized when every subgraph is of the same size, i.e., the number of mistakes is at most $2k + (k+1)(2 \log(n/(k+1)))$.

Accounting for the first mistake, we have that the number of mistake is at most $1 + 2k + (k+1)(2 \log(n/(k+1))) = O(k \log(n/k) + k)$ as claimed. ∎

The next theorem shows a tighter bound. To prove it, we need more notations.

First denote the end points of each edge e_i, for $1 \leq i \leq k$, by p_i and $p_i + 1$. For simplicity, we set $p_0 = 0$. Note that nodes in C_i are $p_{i-1} + 1, p_{i-1} + 2, \ldots, p_i$. We also refers to a set of contiguous nodes as an *interval* of nodes.

We call any node on which the algorithm makes a mistake a *blue* node; note that the number of blue nodes at any time equals the number of mistakes the algorithm makes so far.

As in the proof of Theorem 1, we shall trace the execution of the algorithm.

Theorem 2. *The 1-NN algorithm makes at most* $2k + k \log(n/k) + 1$ *mistakes where* $n = |V|$ *and* k *is the number cut edges.*

Proof: We use a slightly different charging scheme. For each component C_i, we charge the first mistake from nodes in C_i to the component itself. For other mistakes, we use the same charging scheme, i.e., we charge them to the first cut edges encountered on the paths to the source nodes.

For each i, $1 \leq i \leq k$, we will define an interval of nodes W_i that contains all node charging to e_i such that the sets $W_1, W_2, \ldots, W_k$ are pair-wise disjoint.

The total number of mistakes charged to the components is at most $k + 1$. Using the same argument on the maximum of the sum of logarithms as in the end of the proof of Theorem 1, to prove the theorem, it suffices to show that the number of times each cut edge e_i is charged is $1 + \log |W_i|$.

Consider edge e_i that has at least one mistake charged to it.

Let v be the first blue node that charges to e_i. Note that v is either from C_{i-1} or C_i; let C' be one of these subgraphs that contains v, and let C'' be another subgraph.

We assume that v is the first blue node in C'; thus the mistake on v can be charged to C'. We shall deal with the case when v is not the first one later.

There are two cases. The first case is when every node charging to e_i is from C'. Define W_i to be the minimal interval containing one of the endpoint of e_i in C' and v. The proof from Theorem 1 shows that e_i is charged at most $1 + \log |W_i|$ times.

Now consider the second case. Let u be the first node in C'' that charges to e_i. We also assume that u is also the first blue node C''. Recall that u's mistake can be charged to C''. We define W_i to be the minimal interval containing u and v.

At any step t in the execution of the algorithm, let D_i^t be a set of nodes that can possibly charge to e_i *after* step t. The argument as in Theorem 1 implies that after u charges to e_i, in each step t that some node charges to e_i, the size of D_i^t decreases at least by a factor of two, i.e., $|D_i^t| \leq |D_i^{t-1}|/2$.

Now consider each step t before u charges to e_i. Observe that every candidate discarded in this step must be in W_i, i.e., $D_i^{t-1} - D_i^t \subseteq W_i$; thus, in those steps, we also have that

$$|D_i^t \cap W_i| \leq |D_i^{t-1} \cap W_i|/2.$$

Since u does not charges to e_i, we have that e_i is charged by at most $1 + \log |W_i|$ times.

We are left with the case that v or u (or both) are not the first blue nodes on C' or C''. We only consider the case where there exists some mistake in C'' that charges to e_i. Similar argument can be used when all mistakes charged to e_i are from nodes in C' but u is not the first blue node in C'.

Let w' and w'' be the first blue nodes in C' and C''. If $v \neq w'$, we let v' be the node adjacent to w' which is closer to e_i; otherwise we let $v' = v$. We define u' similarly, i.e., u' is the node adjacent to w'' closer to e_i or u itself if $u' = w''$. We define W_i to be the minimal interval containing v' and u'.

We are done if we can show that for each $x \in \{u, v\}$ such that x is not the first node, when the algorithm makes mistakes on x, the set $|D_i^t \cap W_i|$ also shrinks by a factor of two. We look at the case where $x = u$, the other case is similar. To see that this claim is true, note that one of the candidate sources of u is w'', but u chooses another node which is as far as the furthest node on the direction to e_i in $D_i^t \cap W_i$. ∎

Using Theorem 2 with the algorithm of Herbster *et al.*, we obtain the mistake bound of

$$2(\Phi_G(\mathbf{u})) \max \left[0, \log_2 \left(\frac{n}{2\Phi_G(\mathbf{u})} \right) \right] + 4\Phi_G(\mathbf{u}) + 1,$$

which is comparable to the original bound except on the constant of the second term. (We have 4, [6] has $\frac{2}{\ln 2} \approx 2.88$.)

3. Acknowledgments

We would like to thank anonymous referees who gave us useful feedback and suggested the idea for the proof of theorem 2. We also thank Parinya Chalermsook and Mark Herbster for useful discussions.

This work is supported by the Thailand Research Fund Grant MRG5080318.

References

[1] Nicolò Cesa-Bianchi, Claudio Gentile, and Fabio Vitale. Fast and optimal prediction on a labeled tree. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT'09)*, 2009.

[2] Parinya Chalermsook and Jittat Fakcharoenphol. Simple distributed algorithms for approximating minimum steiner trees. In *Proceedings of the 11th Annual International Conference on Computing and Combinatorics (COCOON'05)*, pages 380–389, 2005.

[3] Jittat Fakcharoenphol and Boonserm Kijsirikul. Low congestion online routing and an improved mistake bound for online prediction of graph labeling. *CoRR*, abs/0809.2075, 2008.

[4] Mark Herbster. Exploiting cluster-structure to predict the labeling of a graph. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory (ALT'08)*, pages 54–69, 2008.

[5] Mark Herbster and Guy Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT'09)*, 2009.

[6] Mark Herbster, Guy Lever, and Massimiliano Pontil. On-line prediction on large diameter graphs. In *Proceedings of the 22th Annual Conference on Neural Information Processing Systems (NIPS'08)*, 2008.

[7] Mark Herbster and Massimiliano Pontil. Prediction on a graph with a perceptron. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 577–584. MIT Press, Cambridge, MA, 2006.

[8] Mark Herbster, Massimiliano Pontil, and Lisa Wainer. Online learning over graphs. In *Proceedings of the 22nd international conference on Machine learning (ICML'05)*, pages 305–312, New York, NY, USA, 2005. ACM.

[9] J.M.Barzdin and R.V.Frievald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.