Convex Calibrated Surrogates for Hierarchical Classification

Harish G. Ramaswamy HARISH_GURUP@CSA.IISC.ERNET.IN
Indian Institute of Science, Bangalore, INDIA

Ambuj Tewari TEWARIA@UMICH.EDU
University of Michigan, Ann Arbor, USA

Shivani Agarwal SHIVANI@CSA.IISC.ERNET.IN
Indian Institute of Science, Bangalore, INDIA

Abstract

Hierarchical classification problems are multiclass supervised learning problems with a predefined hierarchy over the set of class labels. In this work, we study the consistency of hierarchical classification algorithms with respect to a natural loss, namely the tree distance metric on the hierarchy tree of class labels, via the usage of calibrated surrogates. We first show that the Bayes optimal classifier for this loss classifies an instance according to the deepest node in the hierarchy such that the total conditional probability of the subtree rooted at the node is greater than $\frac{1}{2}$. We exploit this insight to develop new consistent algorithm for hierarchical classification, that makes use of an algorithm known to be consistent for the "multiclass classification with reject option (MCRO)" problem as a subroutine. Our experiments on a number of benchmark datasets show that the resulting algorithm, which we term OvA-Cascade, gives improved performance over other state-of-the-art hierarchical classification algorithms.

1. Introduction

In many practical applications of the multiclass classification problem the class labels live in a pre-defined hierarchy. For example, in document classification the class labels are topics and they form topic hierarchies; in computational biology the class labels are protein families and they are also best organized in a hierarchy. See Figure 1 for an example hierarchy used in mood classification of speech. Such problems are commonly known in the machine learning literature as *hierarchical classification*.

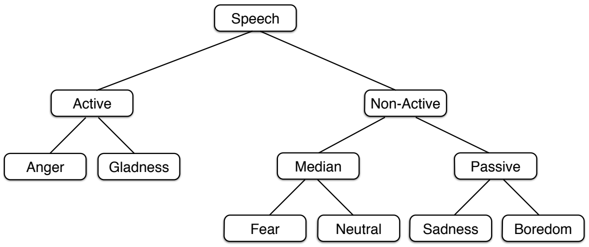

Figure 1. Speech based mood classification hierarchy in the Berlin dataset (Burkhardt et al., 2005) used by Xiao et al. (2007)

Hierarchical classification has been the subject of many studies (Wang et al., 1999; Sun & Lim, 2001; Cai & Hofmann, 2004; Dekel et al., 2004; Rousu et al., 2006; Cesa-Bianchi et al., 2006a;b; Wang et al., 2011; Gopal et al., 2012; Babbar et al., 2013; Gopal & Yang, 2013). For a detailed review and more references we refer the reader to a survey on hierarchical classification by Silla Jr. & Freitas (2011).

The label hierarchy has been incorporated into the problem in various ways in different approaches. The most prevalent and technically appealing approach is to involve the hierarchy in the final evaluation metric and design an algorithm that does well on this evaluation metric. We shall work in the setting where class labels are single nodes in a tree, and use the very natural evaluation metric that penalizes predictions according to the tree-distance between the prediction and truth (Sun & Lim, 2001; Cai & Hofmann, 2004; Dekel et al., 2004).

While hierarchical classification problems are actively studied, there is a gap between theory and practice – even basic statistical properties of hierarchical classification algorithms have not been examined in depth. This paper addresses this gap and its main contributions are summarized below:

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

- We show that the Bayes optimal classifier for the tree-distance loss classifies an instance according to the deepest node in the hierarchy such that the total conditional probability of the subtree rooted at the node is greater than $\frac{1}{2}$.
- We reduce the problem of finding the Bayes optimal classifier for the tree-distance loss to the problem of finding the Bayes optimal classifier for multiclass classification with reject option (MCRO) problem.
- We construct a convex optimization based consistent algorithm for the tree-distance loss based on the above reduction and observe that in one particular instantiation called the OvA-cascade, this optimization problem can be solved only using binary SVM solvers.
- We run the OvA-cascade algorithm on several benchmark datasets and demonstrate improved performance.

2. Preliminaries

Let the instance space be $\mathcal{X}$, and let $\mathcal{Y} = [n] = \{1, \dots, n\}$ be a finite set of class labels. Let $H = ([n], E, W)$ be a tree over the class labels, with edge set E, and positive, finite edge lengths for the edges in E given by W. Let the root node be $r \in [n]$. Let loss function $\ell^H : [n] \times [n] \to \mathbb{R}_+$ be

$$\ell^H(y, y') = \text{Shortest path length in } H \text{ between } y \text{ and } y' \,.$$

We call this the H-distance loss (or simply tree-distance loss). Given training examples $(X_1, Y_1), \dots, (X_m, Y_m)$ drawn i.i.d. from a distribution D on $\mathcal{X} \times \mathcal{Y}$, the goal is to learn a prediction model $g : \mathcal{X} \to [n]$ with low expected ℓ^H-regret defined as

$$\mathbf{R}_D^{\ell^H}[g] = \mathbf{E}[\ell^H(Y, g(X))] - \inf_{g' : \mathcal{X} \to [n]} \mathbf{E}[\ell^H(Y, g'(X))] \,,$$

where expectations are over $(X, Y) \sim D$. Ideally, one wants the ℓ^H-regret of the learned model to be close to zero. An algorithm which when given a random training sample as above produces a (random) model $h_m : \mathcal{X} \to \mathcal{T}$ is said to be *consistent* w.r.t. ℓ^H if the ℓ^H-regret of the learned model g_m converges in probability to zero.

However, minimizing the discrete ℓ^H-regret directly is computationally difficult; therefore one uses instead a *surrogate loss function* $\psi : [n] \times \mathbb{R}^d \to \mathbb{R}_+$ for some $d \in \mathbb{Z}_+$ and learns a model $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$ by minimizing (approximately, based on the training sample) the ψ-error $\mathbf{E}_{(X,Y)\sim D}[\psi(Y, \mathbf{f}(X))]$. Predictions on new instances $x \in \mathcal{X}$ are then made by applying the learned model $\mathbf{f}$ and mapping back to predictions in the target space $[n]$ via some mapping $\Upsilon : \mathbb{R}^d \to [n]$, giving $g(x) = \Upsilon(\mathbf{f}(x))$. Let the ψ-regret of a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$ be

$$\mathbf{R}_D^{\psi}[\mathbf{f}] = \mathbf{E}[\psi(Y, \mathbf{f}(X))] - \inf_{\mathbf{f}' : \mathcal{X} \to \mathbb{R}^d} \mathbf{E}[\psi(Y, \mathbf{f}'(X))] \,.$$

Under suitable conditions, algorithms that approximately minimize the ψ-error based on a training sample are known to be consistent with respect to ψ, i.e. the ψ-regret of the learned model $\mathbf{f}$ approaches zero with larger training data.[1] Also, if ψ is convex in its second argument, the ψ-error minimization problem becomes a convex optimization problem and can be solved efficiently.

We seek a surrogate $\psi : [n] \times \mathbb{R}^d \to \mathbb{R}_+$ for some $d \in \mathbb{Z}_+$ and a predictor $\Upsilon : \mathbb{R}^d \to [n]$ such that ψ is convex in its second argument and satisfies a bound of the following form holding for all $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$ and distributions D

$$\mathbf{R}_D^{\ell^H}[\Upsilon \circ \mathbf{f}] \quad \leq \quad \xi \cdot \mathbf{R}_D^{\psi}[\mathbf{f}], \qquad (1)$$

where $\xi > 0$ is a constant. A surrogate and a predictor (ψ, Υ), satisfying such a bound, which we call a (ψ, ℓ^H, Υ)-excess risk transform,[2] would immediately give an algorithm consistent w.r.t. ℓ^H from an algorithm consistent w.r.t. ψ. We also say that such a (ψ, Υ) is *calibrated* (Zhang, 2004; Ramaswamy & Agarwal, 2012) w.r.t. ℓ^H.

2.1. Conventions and Notations

Δ_n denotes the probability simplex in $\mathbb{R}^n$: $\Delta_n = \{\mathbf{p} \in \mathbb{R}^n_+ : \sum_i p_i = 1\}$.

For the tree $H = ([n], E, W)$ with root r we define the following several objects. For every $y \in [n]$ define the sets $D(y), C(y), U(y)$ as follows:

$$\begin{aligned} D(y) &= \text{Set of descendants of } y \text{ including } y \\ P(y) &= \text{Parent of } y \\ C(y) &= \text{Set of children of } y \\ U(y) &= \text{Set of ancestors of } y\text{, not including } y. \end{aligned}$$

For all $y \in [n]$, define the level of y denoted by lev(y), and the mapping $S_y : \Delta_n \to [0, 1]$ as follows:

$$\begin{aligned} \text{lev}(y) &= |U(y)| \\ S_y(\mathbf{p}) &= \sum_{i \in D(y)} p_i \,. \end{aligned}$$

Let the height of the tree be $h = \max_{y \in [n]} \text{lev}(y)$. Define

[1] For example, an algorithm consistent w.r.t. ψ can be obtained by minimizing the regularized empirical ψ-risk over an RKHS function class with Gaussian kernel and a regularization parameter approaching 0 with increasing sample size.

[2] An inequality which upper bounds the ℓ^H-regret in terms of a function ξ of the ψ-regret, with $\xi(0) = 0$ and ξ continuous at 0 would also qualify to be an excess risk transform.

the sets $N_{=j}, N_{\leq j}$ and scalars α_j, β_j for $0 \leq j \leq h$ as:

$$\begin{aligned} N_{=j} &= \{y \in [n] : \text{lev}(y) = j\} \\ N_{\leq j} &= \{y \in [n] : \text{lev}(y) \leq j\} \\ \alpha_j &= \max_{y,y' \in N_{=j}} \ell^H(y, y') \\ \beta_j &= \max_{y \in N_{=j}} \ell^H(y, P(y)). \end{aligned}$$

By reordering the classes we ensure that lev is a non-decreasing function and hence we always have that $N_{\leq j} = [n_j]$ for some integers n_j and $r = 1$.

For integers $0 \leq j \leq h$ define the function $\text{anc}_j : [n] \to N_{\leq j}$ and $A^j : \Delta_n \to \Delta_{n_j}$ such that for all $y \in [n], y' \in [n_j]$,

$$\begin{aligned} \text{anc}_j(y) &= \begin{cases} y & \text{if lev}(y) \leq j \\ \text{ancestor of } y \text{ at level } j & \text{otherwise} \end{cases} \\ A^j_{y'}(\mathbf{p}) &= \sum_{i \in [n] : \text{anc}_j(i) = y'} p_i \\ &= \begin{cases} p_y & \text{if lev}(y') < j \\ S_y(\mathbf{p}) & \text{if lev}(y') = j \end{cases} . \end{aligned}$$

Note that in all the above definitions the only terms that depend on the edge lengths W are the scalars α_j and β_j.

3. Bayes Optimal Classifier for the Tree-Distance Loss

In this section we characterize the Bayes optimal classifier minimizing the expected tree-distance loss. We show that such a predictor can be viewed as a 'greater than $\frac{1}{2}$ conditional probability subtree detector'. We then design a scheme for computing this prediction based on this observation.

The following theorem is the key result of this section. Figure 2 gives an illustration for this theorem.

Theorem 1. *Let $H = ([n], E, W)$ and let $\ell^H : [n] \times [n] \to \mathbb{R}_+$ be the tree-distance loss for the tree H. For $x \in \mathcal{X}$, let $\mathbf{p}(x) \in \Delta_n$ be the conditional probability of the label given the instance x. Then there exists a $g^* : \mathcal{X} \to [n]$ such that for all $x \in \mathcal{X}$ the following holds:*

(a) $S_{g^*(x)}(\mathbf{p}(x)) \geq \frac{1}{2}$

(b) $S_y(\mathbf{p}(x)) \leq \frac{1}{2}, \forall y \in C(g^*(x))$.

Also, g^* is a Bayes optimal classifier for the tree distance loss, i.e.

$$\mathbf{R}_D^{\ell^H}[g^*] = 0 .$$

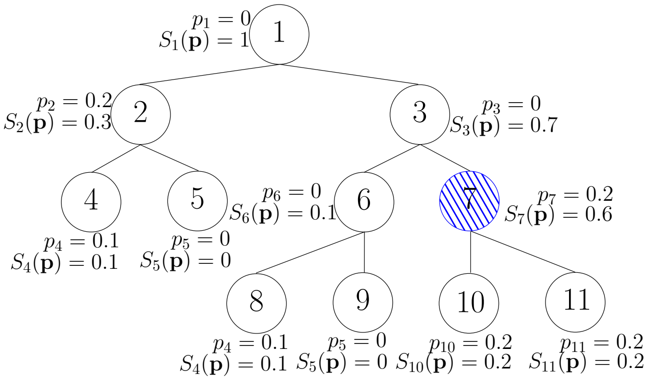

Figure 2. An example tree and an associated conditional probability vector $\mathbf{p}(x)$ for some instance x, along with $S(\mathbf{p}(x))$. The Bayes optimal prediction is shaded here.

For any instance x, with conditional probability $\mathbf{p} \in \Delta_n$, Theorem 1 says that predicting $y \in [n]$ that has the largest level and has $S_y(\mathbf{p}) \geq \frac{1}{2}$ is optimal. Surprisingly, this does not depend on the edge lengths W.

Theorem 1 suggests the following scheme to find the optimal prediction for a given instance, with conditional probability $\mathbf{p}$:

1. For each $j \in \{1, 2, \ldots, h\}$ create a multiclass problem instance with the classes being elements of $N_{\leq j} = [n_j]$, and the probability associated with each class in $y \in N_{\leq j}$ is equal to $A^j_y(\mathbf{p})$, i.e. p_y if $\text{lev}(y) < j$ and equal to $S_y(\mathbf{p})$ if $\text{lev}(y) = j$.

2. For each multiclass problem $j \in \{1, 2, \ldots, h\}$, if there exists a class with probability mass at least $\frac{1}{2}$ assign it to v_j^*, otherwise let $v_j^* = \perp$.

3. Find the largest j such that $v_j^* \neq \perp$ and return the corresponding v_j^*, or return the root 1 if $v_j^* = \perp$ for all $j \in [h]$.

We will illustrate the above procedure for the example in Figure 2.

Example 1. *From Figure 2 we have that $h = 3$. The three induced multiclass problems are given below.*

1. $n_1 = 3$, and the class probabilities are given as $\frac{1}{10}[0, 3, 7]$. Clearly, $v_1^* = 3$.

2. $n_2 = 7$, and the class probabilities are given as $\frac{1}{10}[0, 2, 0, 1, 0, 1, 6]$. Clearly $v_2^* = 7$.

3. $n_3 = 11$, and the class probabilities are given as $\frac{1}{10}[0, 2, 0, 1, 0, 0, 2, 1, 0, 2, 2]$. Clearly, $v_3^* = \perp$.

And hence the largest j such that $v_j^* \neq \perp$ is 2, and the scheme returns $v_2^* = 7$.

The reason such a scheme as the one above is of interest to us is that the second step in the above scheme exactly corresponds to the Bayes optimal classifier for the abstain loss, the evaluation metric used in the MCRO problem, which we briefly explain in the next section.

4. Multiclass Classification with Reject Option

In some multiclass problems like medical diagnosis, it is better to abstain from predicting on instances where the learner is uncertain rather than predicting the wrong class. This feature can be incorporated via an evaluation metric called the abstain loss, and designing algorithms that perform well on this evaluation metric instead of the standard zero-one loss. The n-class abstain loss $\ell^{?,n} : [n] \times ([n] \cup \{\perp\}) \to \mathbb{R}_+$, (Ramaswamy et al., 2015) is defined as

$$\ell^{?,n}(y, y') = \begin{cases} 1 & \text{if } y' \neq y \text{ and } y' \neq \perp \\ \frac{1}{2} & \text{if } y' = \perp \\ 0 & \text{if } y' = y \end{cases} .$$

It can be seen that the Bayes optimal risk for the abstain loss is attained by the function $g^* : \mathcal{X} \to ([n] \cup \{\perp\})$ given by

$$g^*(x) = \begin{cases} \text{argmax}_{y \in [n]} p_y(x) & \text{if } \max_{y \in [n]} p_y(x) \geq \frac{1}{2} \\ \perp & \text{otherwise} \end{cases} ,$$

where $p_y(x) = \mathbf{P}(Y = y | X = x)$.

The $\ell^{?,n}$-regret of a function $g : \mathcal{X} \to ([n] \cup \{\perp\})$ is

$$\mathbf{R}_D^{\ell^{?,n}}[g] = \mathbf{E}[\ell^{?,n}(Y, g(X))] - \inf_{g'} \mathbf{E}[\ell^{?,n}(Y, g'(X))] .$$

Ramaswamy et al. (2015) give three different surrogates and predictors with excess risk transforms relating the surrogate regret to the $\ell^{?,n}$-regret, one of which we give below.

Define the surrogate $\psi^{\text{OvA},n} : [n] \times \mathbb{R}^n \to \mathbb{R}_+$ and predictor $\Upsilon_\tau^{\text{OvA},n} : \mathbb{R}^n \to ([n] \cup \perp)$ as

$$\psi^{\text{OvA},n}(y, \mathbf{u}) = \sum_{i=1}^{n} \mathbf{1}(y = i)(1 - u_i)_+ + \mathbf{1}(y \neq i)(1 + u_i)_+$$

$$\Upsilon_\tau^{\text{OvA},n}(\mathbf{u}) = \begin{cases} \text{argmax}_{i \in [n]} u_i & \text{if } \max_j u_j > \tau \\ \perp & \text{otherwise} \end{cases} ,$$

where $(a)_+ = \max(a, 0)$ and $\tau \in (-1, 1)$ is a threshold parameter, and ties are broken arbitrarily, say, in favor of the label y with the smaller index.

The following theorem by Ramaswamy et al. (2015), gives an $(\psi^{\text{OvA},n}, \ell^{?,n}, \Upsilon_\tau^{\text{OvA},n})$-excess risk transform.

Theorem 2 ((Ramaswamy et al., 2015))**.** *Let $n \in \mathbb{N}$ and $\tau \in (-1, 1)$. Let D be any distribution over $\mathcal{X} \times [n]$. Then, for all $\mathbf{f} : \mathcal{X} \to \mathbb{R}^n$*

$$\mathbf{R}_D^{\ell^{?,n}}[\Upsilon_\tau^{\text{OvA},n} \circ \mathbf{f}] \leq \frac{1}{2(1 - |\tau|)} \mathbf{R}_D^{\psi^{\text{OvA},n}}[\mathbf{f}] .$$

In the next section we use such surrogates calibrated with the abstain loss as a black box to construct calibrated surrogates for the tree-distance loss.

5. Cascade Surrogate for Hierarchical Classification

In this section we construct a template surrogate ψ^{cas} and template predictor Υ^{cas} based on the scheme in Section 3, and is constituted of simpler surrogates ψ^j and predictors $\Upsilon_j^?$. We then give a $(\psi^{\text{cas}}, \ell^H, \Upsilon^{\text{cas}})$-excess risk transform assuming the existence of abstain loss excess risk transforms for the component surrogates and predictors, i.e. $(\psi^j, \ell^{?,n_j}, \Upsilon_j^?)$-excess risk transforms.

For all $j \in \{1, 2, \ldots, h\}$, let the surrogate $\psi^j : [n_j] \times \mathbb{R}^{d_j} \to \mathbb{R}_+$ and predictor $\Upsilon_j^? : \mathbb{R}^{d_j} \to ([n_j] \cup \{\perp\})$ be such that they are calibrated w.r.t. the abstain loss with n_j classes for some integers d_j. Let $d = \sum_{i=1}^{j} d_j$. Let any $\mathbf{u} \in \mathbb{R}^d$ be decomposed as $\mathbf{u} = [\mathbf{u}_1^\top, \ldots, \mathbf{u}_h^\top]^\top$, with each $\mathbf{u}_j \in \mathbb{R}^{d_j}$. The template surrogate, that we call the cascade surrogate $\psi^{\text{cas}} : [n] \times \mathbb{R}^d \to \mathbb{R}_+$, is defined in terms of its constituent surrogates as follows:

$$\psi^{\text{cas}}(y, \mathbf{u}) \quad = \quad \sum_{j=1}^{h} \psi^j(\text{anc}_j(y), \mathbf{u}_j) . \tag{2}$$

The template predictor, Υ^{cas}, is defined via the function $\Upsilon_j^{\text{cas}} : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_j} \to [n_j]$ which is defined recursively as follows:

$$\begin{aligned} &\Upsilon_j^{\text{cas}}(\mathbf{u}_1, \ldots, \mathbf{u}_j) \\ &\quad = \begin{cases} \Upsilon_j^?(\mathbf{u}_j) & \text{if } \Upsilon_j^?(\mathbf{u}_j) \neq \perp \\ \Upsilon_{j-1}^{\text{cas}}(\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}) & \text{otherwise} \end{cases} . \end{aligned} \tag{3}$$

The function Υ_0^{cas} takes no arguments and simply returns 1 (the root node). Occasionally we abuse notation by representing $\Upsilon_j^{\text{cas}}(\mathbf{u}_1, \ldots, \mathbf{u}_j)$ simply as $\Upsilon_j^{\text{cas}}(\mathbf{u})$.

The template predictor, $\Upsilon^{\text{cas}} : \mathbb{R}^d \to [n]$ is simply defined as $\Upsilon^{\text{cas}}(\mathbf{u}) = \Upsilon_h^{\text{cas}}(\mathbf{u}_1, \ldots, \mathbf{u}_h)$.

The lemma below, captures the essence of the reduction from the hierarchical classification problem to the MCRO problem. A proof outline is also provided.

Lemma 3. *Let $H = ([n], E, W)$ be a tree with height h. For all $j \in [h]$, let $\alpha_j = \max_{y, y' \in N_{=j}} \ell^H(y, y')$. For any distribution D over $\mathcal{X} \times [n]$, let $A^j(D)$ be the distribution* *over $\mathcal{X}\times[n_j]$ given by the distribution of $(X, \mathrm{anc}_j(Y))$ with $(X,Y)\sim D$. For all $j\in[h]$, let $\mathbf{f}_j:\mathcal{X}\to\mathbb{R}^{d_j}$ be such that $\mathbf{f}(x)=[\mathbf{f}_1(x)^\top,\dots,\mathbf{f}_h(x)^\top]^\top$. Then for all distributions D over $\mathcal{X}\times[n]$ and all functions $\mathbf{f}:\mathcal{X}\to\mathbb{R}^d$*

$$\mathbf{R}_D^{\ell^H}[\Upsilon^{\mathrm{cas}}\circ\mathbf{f}]\le\sum_{j=1}^{h}2\alpha_j\cdot\mathbf{R}_{A^j(D)}^{\ell^{?,n_j}}[\Upsilon_j^?\circ\mathbf{f}_j]\ .$$

Proof. (Outline:)

Due to linearity of expectation, it is sufficient to fix a singleton $\mathcal{X}$, and give proofs for all distributions $\mathbf{p}\in\Delta_n$ over class labels, instead of all distributions D over $\mathcal{X}\times\mathcal{Y}$.

For a given conditional probability vector $\mathbf{p}\in\Delta_n$, and vector $\mathbf{u}\in\mathbb{R}^d$, the analysis is based on whether the abstain loss predictor at the deepest level (level farthest from root) abstains or not.

1. If the abstain loss predictor at the deepest level does not abstain (Case 1 in the proof), then the tree-distance regret is bounded by the maximum distance between any two nodes at the deepest level α_h, with a discount factor depending on the conditional probability of the predicted class. This can be simply be bounded by $2\alpha_h$ times the abstain loss regret.

2. If the abstain loss predictor at the deepest level does abstain and the optimal prediction is not in the deepest level (Case 2a in the proof), then prediction for the deepest level is 'correct' and hence one can show that the tree-distance regret is simply bounded by the tree-distance regret for the modified problem where all the probability mass associated with the nodes in deepest level are absorbed by their parents.

3. If the abstain loss predictor at the deepest level does abstain and the optimal prediction is in the deepest level (Case 2b in the proof), then one can bound the tree-distance regret by the sum of two terms –

   (a) The abstain loss regret, weighted by twice the largest distance between any node at the deepest level and its parent β_h. This captures the error made by choosing to predict at a shallower level than the level of optimal prediction.

   (b) The tree-distance regret on the modified problem mentioned in case 2a. This captures the error made on shallower levels.

In all cases, the tree-distance regret can be bounded by the sum of the tree-distance regret on the modified problem and $2\alpha_h$ times the abstain loss regret. Applying this bound recursively gets our desired bound. □

Lemma 3 bounds the ℓ^H regret on distribution D, by a weighted sum of abstain loss regrets, each over a modified distribution derived from D. Each of the components of the surrogate ψ^{cas} is exactly designed to minimize the abstain loss for the corresponding modified distribution. Assuming a $(\psi^j,\ell^{?,n_j},\Upsilon_j^?)$-excess risk transform for all $j\in[h]$, one can easily derive $(\psi^{\mathrm{cas}},\ell^H,\Upsilon^{\mathrm{cas}})$-excess risk transform as in Equation 1. This is done in the theorem below.

Theorem 4. *Let $H=([n],E,W)$ be a tree with height h. For all $j\in[h]$, let $\psi^j:[n_j]\times\mathbb{R}^{d_j}\to\mathbb{R}_+$ and $\Upsilon_j^?:\mathbb{R}^{d_j}\to n_j$ be such that for all $\mathbf{f}_j:\mathcal{X}\to\mathbb{R}^{d_j}$, and all distributions D over $\mathcal{X}\times[n_j]$ we have*

$$\mathbf{R}_D^{\ell^{?,n_j}}[\Upsilon_j^?\circ\mathbf{f}_j]\le C\cdot\mathbf{R}_D^{\psi^j}[\mathbf{f}_j],$$

for some constant $C>0$. Then for all $\mathbf{f}:\mathcal{X}\to\mathbb{R}^d$ and distributions D over $\mathcal{X}\times[n]$,

$$\mathbf{R}_D^{\ell^H}[\Upsilon^{\mathrm{cas}}\circ\mathbf{f}]\le 2C\cdot\max_{y,y'\in[n]}\ell^H(y,y')\cdot\mathbf{R}_D^{\psi^{\mathrm{cas}}}[\mathbf{f}]\ .$$

Hence one just needs to plug in an appropriate surrogate ψ^j to get concrete consistent algorithms for hierarchical classification. The results of Ramaswamy et al. (2015) give three such surrogates, but we will focus on the one vs all hinge surrogate here, as the resulting algorithm can be easily parallelized and gives the best empirical results.

6. OvA-Cascade Algorithm

When $\psi^j=\psi^{\mathrm{OvA},n_j}$ and $\Upsilon_j^?=\Upsilon_{\tau_j}^{\mathrm{OvA},n_j}$ for some $\tau_j\in(-1,1)$, we call the resulting cascade surrogate ψ^{cas} and predictor Υ^{cas} together as OvA-Cascade. In this case we have $d_j=n_j$. In the surrogate minimizing algorithm for OvA-cascade, one solves h one-vs-all SVM problems. Problem j has n_j classes, with the classes corresponding to the n_{j-1} nodes in the hierarchy at level less than j, and n_j-n_{j-1} 'super-nodes' in the hierarchy at level j which also absorb the nodes of its descendants. The resulting training and prediction algorithms can thus be simplified and they are presented in Algorithms 1 and 2. The training phase requires an SVM optimization sub-routine, SVM-Train, which takes in a binary dataset and a regularization parameter C and returns a real valued function over the instance space minimizing the regularized hinge loss over an appropriate function space.

Theorems 2 and 4 immediately give the following corollary.

Corollary 5. *Let $H=([n],E,W)$ be a tree with height h. Let the component surrogates and predictors of ψ^{cas} and Υ^{cas} be $\psi^j=\psi^{\mathrm{OvA},n_j}$ and $\Upsilon^j=\Upsilon_{\tau_j}^{\mathrm{OvA},n_j}$. Then, for all distributions D and functions $\mathbf{f}:\mathcal{X}\to\mathbb{R}^d$,*

$$\mathbf{R}_D^{\ell^H}[\Upsilon^{\mathrm{cas}}\circ\mathbf{f}]\le\frac{\max_{y,y'\in[n]}\ell^H(y,y')}{1-\max_j|\tau_j|}\cdot\mathbf{R}_D^{\psi^{\mathrm{cas}}}[\mathbf{f}]\ .$$

Algorithm 1 OVA-Cascade Training

```
**Input:** $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$, $H = ([n], E)$.
**Parameters:** Regularization parameter $C > 0$
**for** $i = 1 : n$
    Let $t_j = 2 \cdot \mathbf{1}(y_j \in D(i)) - 1, \;\; \forall j \in [m]$
    $T_i = ((x_1, t_1), \ldots, (x_m, t_m)) \in (\mathcal{X} \times \{+1, -1\})^m$.
    $f_i$=SVM-Train$(T_i, C)$
    Let $t'_j = 2 \cdot \mathbf{1}(y_j = i) - 1, \;\; \forall j \in [m]$
    $T'_i = ((x_1, t'_1), \ldots, (x_m, t'_m)) \in (\mathcal{X} \times \{+1, -1\})^m$.
    $f'_i$=SVM-Train$(T'_i, C)$
**end for**
```

Algorithm 2 OVA-Cascade Prediction

```
**Input:** $x \in \mathcal{X}$, $H = ([n], E)$, trained models $f_i, f'_i$ for all $i \in [n]$
**Parameters:** Scalars $\tau_1, \ldots, \tau_h$ in $(-1, 1)$
**for** $j = h$ **down to** $1$
    Construct $\mathbf{u} \in \mathbb{R}^{n_j}$ such that,
    $u_i = \begin{cases} f_i(x) & \text{if lev}(i) = j \\ f'_i(x) & \text{if lev}(i) < j \end{cases}$
    **if** $\max_i u_i > \tau_j$
        **return** $\text{argmax}_i u_i$
    **end if**
**end for**
**return** $1$
```

To get the best bound from Corollary 5, one must set $\tau_j = 0$ for all $j \in [h]$. However, using a slightly more intricate version of Theorem 2 and Lemma 3 one can give a better upper bound for the ℓ^H-regret than in Theorem 4, and this tighter upper bound is minimized for a different τ_j. This observation is captured by the Theorem below.

Theorem 6. *Let $H = ([n], E, W)$ be a tree with height h. For all $j \in [h]$, let $\alpha_j = \max_{y, y' \in N_{=j}} \ell^H(y, y')$ and let $\beta_j = \max_{y \in N_{=j}} \ell^H(y, P(y))$. For $j \in [h]$, let $\tau_j = \frac{\alpha_j - \beta_j}{\alpha_j + \beta_j}$. Let the component surrogates and predictors of ψ^{cas} and Υ^{cas} be $\psi^j = \psi^{\text{OvA}, n_j}$ and $\Upsilon^j = \Upsilon^{\text{OvA}, n_j}_{\tau_j}$. Then, for all distributions D and functions $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$,*

$$\mathbf{R}^{\ell^H}_D [\Upsilon^{\text{cas}} \circ \mathbf{f}] \le \frac{1}{2} \max_{j \in [h]} (\alpha_j + \beta_j) \cdot \mathbf{R}^{\psi^{\text{cas}}}_D [\mathbf{f}] \; .$$

One can clearly see the effect of improved bounds given by setting τ_j as in Theorem 6 for the unweighted hierarchy, in which case $\alpha_j = 2j$ and $\beta_j = 1$.

Corollary 7. *Let the hierarchy H be an unweighted tree with all edges having length 1. Let the component surrogates and predictors of ψ^{cas} and Υ^{cas} be $\psi^j = \psi^{\text{OvA}, n_j}$ and $\Upsilon^j = \Upsilon^{\text{OvA}, n_j}_{\tau_j}$.*

a. For all $j \in [h]$ let $\tau_j = 0$, then, for all distributions D and functions $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$,

$$\mathbf{R}^{\ell^H}_D [\Upsilon^{\text{cas}} \circ \mathbf{f}] \le 2h \cdot \mathbf{R}^{\psi^{\text{cas}}}_D [\mathbf{f}] \; .$$

b. For all $j \in [h]$ let $\tau_j = \frac{2j-1}{2j+1}$, then, for all distributions D and functions $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$,

$$\mathbf{R}^{\ell^H}_D [\Upsilon^{\text{cas}} \circ \mathbf{f}] \le \left(h + \frac{1}{2} \right) \cdot \mathbf{R}^{\psi^{\text{cas}}}_D [\mathbf{f}] \; .$$

Thus setting $\tau_j = \frac{2j-1}{2j+1}$ gives almost a factor 2 improvement over setting $\tau_j = 0$. This threshold setting is also intuitively satisfying as it says to use a higher threshold and predict conservatively (abstain more often) in deeper levels and to use a lower threshold and predict aggressively in levels nearer to the root. In practice, the optimal thresholds are distribution dependent and are best obtained via cross-validation.

7. Experiments

We run our cascade surrogate based algorithm for hierarchical classification on some standard document classification tasks with a class hierarchy and compare the results against other standard algorithms. We use the unweighted tree-distance loss as the evaluation metric. The details of the datasets and the algorithms are given below.

7.1. Datasets

We used several standard multiclass document classification datasets, all of which have one class label per example. The basic statistics of the datasets is given in Table 1.

- **CLEF** (Dimitrovski et al., 2011) Medical X-ray images organized according to a hierarchy.
- **IPC** [3] Patents organized according to the International Patent Classification Hierarchy.
- **LSHTC-small, DMOZ-2010 and DMOZ-2012** [4] Web-pages, from the LSHTC (Large-Scale Hierarchical Text Classification) challenges 2010-12, organized according to a hierarchy.

We used the standard train-test splits wherever available and possible. For the DMOZ-2010 and 2012 datasets however we created our own train-test splits because the given test sets do not contain class labels and the oracle for evaluating submissions does not accept interior nodes as predictions.

[3] http://www.wipo.int/classifications/ipc/en/support/

[4] http://lshtc.iit.demokritos.gr/node/3

Table 1. Dataset Statistics

Dataset	#Train	#Validation	#Test	#Labels	#Leaf-Labels	Depth	#Features
CLEF	9,000	1,000	1,006	97	63	3	89
LSHTC-small	4,463	1,860	1,858	2,388	1,139	5	51,033
IPC	35,000	11,324	28,926	553	451	3	541,869
DMOZ-2010	80,000	13,805	34,905	17,222	12,294	5	381,580
DMOZ-2012	250,000	50,000	83,408	13,347	11,947	5	348,548

Table 2. Average tree-distance loss on the test set. Runs that failed due to memory issues are denoted by a '-'.

	Root	**OVA**	**HSVM-margin**	**HSVM-slack**	**CS-Cascade**	**OVA-Cascade**	**Plug-in**
CLEF	3.00	1.10	0.98	1.00	**0.91**	0.95	0.97
LSHTC-small	4.77	4.12	3.47	3.54	3.20	**3.19**	3.26
IPC	2.97	2.29	-	-	-	2.06	**2.05**
DMOZ-2010	4.65	3.96	-	-	-	**3.12**	3.16
DMOZ-2012	4.75	2.83	-	-	-	**2.46**	2.48

Table 3. Training times (not including validation) in hours (h) or seconds (s). Runs that failed due to memory issues are denoted by a '-'.

	Root	**OVA**	**HSVM-margin**	**HSVM-slack**	**CS-Cascade**	**OVA-Cascade**	**Plug-in**
CLEF	0 s	35s	50 s	45 s	20 s	50 s	66 s
LSHTC-small	0 h	0.24 h	2.1 h	1.8 h	1.7 h	0.3 h	0.5 h
IPC	0 h	2.6 h	-	-	-	2.9 h	4.2 h
DMOZ-2010	0 h	36 h	-	-	-	59 h	146 h
DMOZ-2012	0 h	201 h	-	-	-	220 h	361 h

7.2. Algorithms

We run a variety of algorithms on the above datasets. The details of the algorithms are given below.

Root: This is a simple baseline method where the returned classifier always predicts the root of the hierarchy.

OVA: This is the standard One vs All algorithm which completely ignores the hierarchy information and treats the problem as one of standard multiclass classification.

HSVM-margin and HSVM-slack : These algorithms are Struct-SVM like (Tsochantaridis et al., 2005) algorithms for the tree-distance loss as proposed in Cai & Hofmann (2004). HSVM-margin and HSVM-slack use margin and slack rescaling respectively, and are considered among the state-of-the-art algorithms for hierarchical classification.

OVA-Cascade: This is the algorithm in which we minimize the surrogate ψ^{cas} with the component surrogates being $\psi^j = \psi^{\text{OvA},n_j}$ and is detailed as Algorithms 1 and 2. All the datasets in Table 1 have the property that all instances are associated only with a leaf-label (note however that we can still predict interior nodes), and hence the step of computing f'_i in Algorithm 1 can be skipped, and f'_i can be set to be identically equal to negative infinity for all $i \in [n]$. Note that, in this case, the training phase is very similar to the 'less-inclusive policy' using the 'local node approach' (Silla Jr. & Freitas, 2011). We use LIBLINEAR (Fan et al., 2008) for the SVM-train subroutine and use the simple linear kernel. The regularization parameter C is chosen via a separate validation set. The thresholds τ_j for $j \in [h]$ are also chosen via a coarse grid search using the validation set.

Plug-in classifier: This algorithm is based on estimating the conditional probabilities using a logistic loss. Specifically, it estimates $S_y(\mathbf{p})$ for all non-root nodes y. This is done by creating a binary dataset for each y, with instances having labels which are the descendants of y being positive and the rest being negative, and running a logistic regression algorithm on this dataset. The final predictor is simply based on Theorem 1, it chooses the deepest node y such that the estimated value of $S_y(\mathbf{p})$ is greater than $\frac{1}{2}$.

CS-Cascade: This algorithm also minimizes the cascade surrogate ψ^{cas}, but with the component surrogates ψ^j being the Crammer-Singer surrogate (Crammer & Singer, 2001). From the results of Ramaswamy et al. (2015), one can derive excess risk transforms for the resulting cascade surrogate as well. As all instances have labels which are leaf nodes, the h subproblems all turn out to be multiclass learning problems with n_j classes for each of which we use the Crammer-Singer algorithm. We optimize the Crammer-Singer surrogate over the standard multiclass linear function class using the LIBLINEAR software. Once again we use the same regularization parameter C for all the h problems which we choose using the validation set. We also use a threshold vector tuned on the validation set over a coarse grid.

The three algorithms OvA-Cascade, Probability estimation and CS-cascade are all motivated by our analysis and would form consistent algorithms for the tree-distance loss if used with an appropriate function class.

7.3. Discussion of Results

Table 2 gives the average tree-distance loss incurred by various algorithms on some standard datasets and Table 3 gives the times taken for running these algorithms on a 4-core CPU.[5] Some of the algorithms, like HSVM, and CS-cascade could not be run on the larger datasets due to memory issues. In the smaller datasets of `CLEF` and `LSHTC-small` where all the algorithms could be run, the algorithms motivated by our analysis – OvA-cascade, Plug-in and CS-cascade – perform the best. In the bigger datasets, only the OvA-cascade, plug-in and the flat OvA algorithms could be run, and both OvA-cascade and Plug-in perform significantly better than the flat OvA. While both OvA-cascade and Plug-in give comparable error performance, the OvA-cascade only takes about half as much time as the Plug-in and hence is more preferable.

8. Conclusion

The reduction of the hierarchical classification problem to the problem of multiclass classification with a reject option gives an interesting and powerful family of algorithms. Extending such results to other related settings, such as the case where there is a graph over the set of class labels, or where a subset of the label set is allowed to be predicted instead of a single label, are interesting future directions.

Acknowledgements

HGR and AT gratefully acknowledge the support of NSF via grant CCF-1422157. HGR also acknowledges support from TCS via a PhD Fellowship and a grant from the Indo-US virtual institute for mathematical and statistical sciences (VIMSS) . SA acknowledges support from the Department of Science & Technology (DST) of the Indian Government under a Ramanujan Fellowship, from the Indo-US Science & Technology Forum (IUSSTF), and from Yahoo in the form of an unrestricted grant.

[5] HSVM, and CS-cascade effectively only use a single core due to lack of parallelization.

References

Babbar, R., Partalas, I., Gaussier, E., and Amin, M.-R. On flat versus hierarchical classification in large-scale taxonomies. In *Advances in Neural Information Processing Systems 26*, 2013.

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., and Weiss, B. A database of german emotional speech. In *Proceedings of the 9th European conference on speech communication and technology*, 2005.

Cai, L. and Hofmann, T. Hierarchical document categorization with support vector machines. In *International Conference on Information and Knowledge Management*, 2004.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. Hierarchical classification: combining Bayes with SVM. In *International Conference on Machine Learning*, 2006a.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7:31–54, 2006b.

Crammer, K. and Singer, Y. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 2001.

Dekel, O., Keshet, J., and Singer, Y. Large margin hierarchical classification. In *International Conference on Machine Learning*, 2004.

Dimitrovski, I., Kocev, D., Suzana, L., and Dzeroski, S. Hierchical annotation of medical images. *Pattern Recognition*, 2011.

Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Gopal, S. and Yang, Y. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *International Conference on Knowledge Discovery and Data Mining*, 2013.

Gopal, S., Bai, B., Yang, Y., and Niculescu-Mizil, A. Bayesian models for large-scale hierarchical classification. In *Advances in Neural Information Processing Systems 25*, 2012.

Ramaswamy, H. G. and Agarwal, S. Classification calibration dimension for general multiclass losses. In *Advances in Neural Information Processing Systems 25*, 2012.

Ramaswamy, H. G., Tewari, A., and Agarwal, S. Consistent algorithms for multiclass classification with a reject option. arXiv:1505.04137, 2015.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006.

Silla Jr., C. N. and Freitas, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2011.

Sun, A. and Lim, E.-P. Hierarchical text classification and evaluation. In *International Conference on Data Mining*, 2001.

Tsochantaridis, I., Joachims, T., Hoffman, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

Wang, H., Shen, X., and Pan, W. Large margin hierarchical classification with mutually exclusive class membership. *Journal of Machine Learning Research*, 12:2721–2748, 2011.

Wang, K., Zhou, S., and Liew, S. C. Building hierarchical classifiers using class proximity. In *International Conference on Very Large Data Bases*, 1999.

Xiao, Z., Dellandréa, E., Dou, W., and Chen, L. Hierarchical classification of emotional speech. *IEEE Transactions on Multimedia*, 2007.

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004.