Discriminative Learning of Relaxed Hierarchy for Large-scale Visual Recognition — Supplementary Material

Tianshi Gao
Dept. of Electrical Engineering
Stanford University
tianshig@stanford.edu

Daphne Koller
Dept. of Computer Science
Stanford University
koller@cs.stanford.edu

1. Introduction

In this supplementary material, we first discuss how to efficiently solve the optimization problem (2) in Section 3.2 of the main paper. Then, we present two theorems for the generalization error of our model and discuss how our method is related to minimizing the error bound in Section 3. In Section 4, we prove the main theorems. Finally, we show the learned hierarchies for both the Caltech-256 and the SUN datasets in Section 5.

2. Optimization

In this section, we present an efficient algorithm to solve the optimization problem (2) in Section 3.2 of the main paper. We first ignore the constraint $\sum_{k=1}^{K}\mathbf{1}\{\mu_k>0\}\geq 1$ and $\sum_{k=1}^{K}\mathbf{1}\{\mu_k<0\}\geq 1$, and discuss how to solve (1) efficiently.

$$\begin{aligned}\min_{\{\mu_k\}}\quad & C\sum_{k=1}^{K}n_k\Big(\mathbf{1}\{\mu_k=+1\}\,(\frac{1}{n_k}\sum_{i:y_i=k}\xi_i^{+}-\frac{A}{C})\\ & \qquad+\mathbf{1}\{\mu_k=-1\}\,(\frac{1}{n_k}\sum_{i:y_i=k}\xi_i^{-}-\frac{A}{C})\Big)\\ \text{s.t.}\quad & \mu_k\in\{-1,0,+1\},\ \forall k\in\mathcal{Y}\\ & -B\leq\sum_{k=1}^{K}\mu_k\leq B\end{aligned}\tag{1}$$

Let

$$f_k(\mu_k)=\begin{cases}Cn_k(\frac{1}{n_k}\sum_{i:y_i=k}\xi_i^{-}-\rho) & \text{if } \mu_k=-1\\ 0 & \text{if } \mu_k=0\\ Cn_k(\frac{1}{n_k}\sum_{i:y_i=k}\xi_i^{+}-\rho) & \text{if } \mu_k=1\end{cases}\tag{2}$$

where $\rho=\frac{A}{C}$. Then the objective function in (1) can be written as $\sum_k f_k(\mu_k)$. Let $\mu'_k=\text{argmin}_{\mu_k}f_k(\mu_k)$. Then $\sum_k f_k(\mu'_k)$ is the lowest possible value the objective function can take regardless of the constraint. To facilitate later discussion, we define $S'_+=\{k|\mu'_k=1\}$, $S'_0=\{k|\mu'_k=0\}$ and $S'_-=\{k|\mu'_k=-1\}$. Furthermore, to simplify the discussion we assume B is a positive integer, *i.e.*, $B\geq 1$. For $B=0$, the principles of the following discussions remain the same with some special considerations of corner cases. Let $B'=\sum_k\mu'_k$. According to different relations between B' and B, there are three cases.

First, if $-B\leq B'\leq B$, then the optimal solution of (1) is $\mu^\star_k=\mu'_k,\ \forall k$.

Second, if $B'>B$, we first characterize the relations between the optimal solution $\{\mu^\star_k\}_{k=1}^K$ and $\{\mu'_k\}_{k=1}^K$, and then provide an efficient algorithm to find such an optimal solution. Note that for the third case, *i.e.*, $B'<-B$, we can use an analogous procedure to find the optimal solution similar to the case of $B'>B$ by symmetry.

We first prove the following theorem characterizing the optimal solution of (1).

Theorem 2.1. *If $B'=\sum_k\mu'_k>B$, then there exists an optimal solution $\{\mu^\star_k\}_{k=1}^K$ such that $B^\star=B$ or $B-1$, where $B^\star=\sum_k\mu^\star_k$ (note that $B-1>-B$, since we assume $B\geq 1$).*

Proof. Given an optimal solution $\{\mu^\star_k\}_{k=1}^K$, suppose $B^\star\leq B-2$. We have $\sum_k\mu'_k-\sum_k\mu^\star_k=\sum_k(\mu'_k-\mu^\star_k)=\sum_{k\in S'_-}(\mu'_k-\mu^\star_k)+\sum_{k\in S'_0}(\mu'_k-\mu^\star_k)+\sum_{k\in S'_+}(\mu'_k-\mu^\star_k)=B'-B^\star\geq B'-B+2$. Since $\sum_{k\in S'_-}(\mu'_k-\mu^\star_k)=\sum_{k\in S'_-}(-1-\mu^\star_k)\leq 0\,(\because\mu^\star_k\in\{-1,0,1\}\geq -1)$, we have $\sum_{k\in S'_0}(\mu'_k-\mu^\star_k)+\sum_{k\in S'_+}(\mu'_k-\mu^\star_k)\geq B'-B+2>2$. This implies that, there exists at least one $\tilde{k}\in S'_o\cup S'_+$ such that $\mu'_{\tilde{k}}-\mu^*_{\tilde{k}}=1$ or 2. Now we construct a solution by changing $\mu^*_{\tilde{k}}$ to $\mu'_{\tilde{k}}$, and keep other μ^*_k's unchanged. Then for the new solution, $\tilde{B}=\mu'_{\tilde{k}}+\sum_{k\neq\tilde{k}}\mu^\star_k=B^\star+\mu'_{\tilde{k}}-\mu^*_{\tilde{k}}\leq B^\star+2\leq B$. This means that the new solution is a feasible solution. In terms of the objective value, since $f_{\tilde{k}}(\mu'_{\tilde{k}})\leq f_{\tilde{k}}(\mu^*_{\tilde{k}})$ (by definition), the objective value of the new solution is no worse than that of $\{\mu^\star_k\}_{k=1}^K$. If $f_{\tilde{k}}(\mu'_{\tilde{k}})$ is

strictly less than $f_{\tilde{k}}(\mu^*_{\tilde{k}})$, then we find another feasible solution with better objective value, which is contradictory to the optimality of $\{\mu^\star_k\}_{k=1}^K$. Thus, $B^\star > B-2$, *i.e.*, $B^\star = B$ or $B-1$. Otherwise if $f_{\tilde{k}}(\mu'_{\tilde{k}}) = f_{\tilde{k}}(\mu^*_{\tilde{k}})$, if $\tilde{B} > B-2$, then we find an optimal solution satisfying the statement in the theorem, otherwise if $\tilde{B} \leq B-2$, we can recursively apply the same argument above based on the newly constructed optimal solution, which will lead to contradiction or eventually find an optimal solution satisfying the condition in the theorem. □

We present another theorem which shows another property of the optimal solution:

Theorem 2.2. *If $B' = \sum_k \mu'_k > B$, among the optimal solutions satisfying the condition in Theorem 2.1, there exists an optimal solution $\{\mu^\star_k\}_{k=1}^K$ such that $\forall k \in S'_-,\ \mu^\star_k = \mu'_k$.*

Proof. Suppose there exists an optimal solution (satisfying the condition in Theorem 2.1) with some $\tilde{k} \in S'_-$ such that $\mu^\star_{\tilde{k}} \neq \mu'_{\tilde{k}} = -1$. Similar to the analysis in the proof of Theorem 2.1, we have $\sum_k \mu'_k - \sum_k \mu^\star_k = \sum_k (\mu'_k - \mu^\star_k) = \sum_{k\in S'_-}(\mu'_k - \mu^\star_k) + \sum_{k\in S'_0}(\mu'_k - \mu^\star_k) + \sum_{k\in S'_+}(\mu'_k - \mu^\star_k) = B' - B^\star \geq B' - B \geq 1$. Thus, $\sum_{k\in S'_0}(\mu'_k - \mu^\star_k) + \sum_{k\in S'_+}(\mu'_k - \mu^\star_k) \geq 1 - \sum_{k\in S'_-}(\mu'_k - \mu^\star_k) \geq 2$ ($\because \mu'_{\tilde{k}} - \mu^\star_{\tilde{k}} \leq -1$). This implies that, there exists at least two $j \in S'_o \cup S'_+$ such that $\mu'_j - \mu^\star_j = 1$ and/or at least one $i \in S'_+$ such that $\mu'_i - \mu^\star_i = 2$. Now we construct a new solution by first changing $\mu^\star_{\tilde{k}}$ to $\mu'_{\tilde{k}} = -1$. If $\mu^\star_{\tilde{k}}$ was 1, then we also change two $\mu^\star_j$ to $\mu'_j = \mu^\star_j + 1$, or change one $\mu^\star_i$ to $\mu'_i = \mu^\star_i + 2$. If $\mu^\star_{\tilde{k}}$ was 0, we change one $\mu^\star_j$ to $\mu^\star_j + 1$ or change one $\mu^\star_i$ to $\mu'_i = \mu^\star_i + 2$ (only when there is no single j such that $\mu'_j - \mu^\star_j = 1$). The new solution still has $B_{\text{new}} = B^\star$ or $B^\star + 1$ (only when there is no single j such that $\mu'_j - \mu^\star_j = 1$, which implies $B^\star = B - 1$ as shown in the algorithm). Therefore, the new solution is still feasible. However, since we changed $\mu^\star_{\tilde{k}}$ to $\mu'_{\tilde{k}}$ and some $\mu^\star_j$ to μ'_j or $\mu^\star_i$ to μ'_i, the objective value becomes no worse. If the new solution has lower objective value, then by contradiction, we conclude that $\forall k \in S'_-,\ \mu^\star_k = \mu'_k$. If the new solution has the same objective value, we can use the same argument above based on this new solution, which will lead to contradiction (thus proving the theorem) or construct an optimal solution such that $\forall k \in S'_-,\ \mu^\star_k = \mu'_k$, which is consistent with the statement in the theorem. □

Similar to Theorem 2.2, the following statement is also true.

Theorem 2.3. *If $B' = \sum_k \mu'_k > B$, among the optimal solutions satisfying both the conditions in Theorem 2.1 and Theorem 2.2, there exists an optimal solution $\{\mu^\star_k\}_{k=1}^K$ such that $\forall k \in S'_0,\ \mu^\star_k = \mu'_k = 0$ or $\mu^\star_k = -1$. In other words, there is no $k \in S'_0$ such that $\mu^\star_k = 1$.*

Proof. Follow the same principle of the proof for Theorem 2.2. Basically, if there is some $\tilde{k} \in S'_0$ such that $\mu^\star_{\tilde{k}} = 1 > \mu'_{\tilde{k}} = 0$, then we can always construct another solution by changing $\mu^\star_{\tilde{k}}$ to $\mu'_{\tilde{k}} = 0$ and increase some other $\mu^\star_j$ to μ'_j where $j \in S'_o \cup S'_+$. This new solution will be feasible and has an objective value that is no worse than that of the previous optimal solution. If the new solution has lower objective value, then we have a contradiction, thus proving the theorem. Otherwise, we can repeat such construction, which will lead to contradiction or eventually construct another optimal solution satisfying the condition in the theorem. □

Based on the above theorems, we can transform the original problem (1) to:

$$\begin{aligned} \min_{\{\mu_k\}} \quad & \sum_{k=1}^K f_k(\mu_k) - f_k(\mu'_k) \\ \text{s.t.} \quad & \forall k \in S'_-,\ \mu_k = \mu'_k = -1 \\ & \forall k \in S'_0,\ , \mu_k \in \{-1, 0\} \\ & \forall k \in S'_+,\ , \mu_k \in \{-1, 0, 1\} \\ & \sum_{k\in S'_0} (\mu'_k - \mu_k) + \sum_{k\in S'_+} (\mu'_k - \mu_k) \\ & = B' - B \text{ or } B' - B + 1 \end{aligned} \tag{3}$$

Since $f_k(\mu_k) \geq f_k(\mu'_k)$, for any k such that $\mu_k \neq \mu'_k$, it will incur a non-negative increase in the objective. Thus, the optimal solution $\{\mu^\star_k\}$ should have as few differences from $\{\mu'_k\}$ as possible. However, the last constraint in (3) requires that the sum of the difference between $\{\mu'_k\}$ and $\{\mu_k\}$ is $B'-B$ or $B'-B+1$. Then the problem (3) is equivalent to finding a subset of k's from $S'_0 \cup S'_+$ and change the values of those μ_k from μ'_k to some other values such that the sum of the difference is $B' - B$ or $B' - B + 1$ while the cost of these changes is the minimum (corresponding to the minimum objective value). To achieve this requirement, for $k \in S'_0$, if we change μ_k from $\mu'_k = 0$ to -1, it will contribute $\mu'_k - \mu_k = 1$. However, it incurs a delta increase in the objective $\Delta_k = f_k(-1) - f_k(0)$. For $k \in S'_+$, there are two cases. The first is to change μ_k from $\mu'_k = 1$ to 0, which will contribute $\mu'_k - \mu_k = 1$ to the difference while incurring a delta increase in the objective $\Delta_{k,0} = f_k(0) - f_k(1)$. The second case is to change μ_k from $\mu'_k = 1$ to -1, which will contribute $\mu'_k - \mu_k = 2$ to the difference while incurring a delta increase in the objective $\Delta_{k,-} = f_k(-1) - f_k(1)$. We also denote $\Delta_k = f_k(-1) - f_k(0)$ for $k \in S'_+$. Note that $\Delta_{k,-} = \Delta_{k,0} + \Delta_k$. The interpretation of Δ_k, $\Delta_{k,0}$ and $\frac{\Delta_{k,-}}{2}$ is that they are the cost per unit increase

in $\sum_{k\in S'_0}(\mu'_k - \mu_k) + \sum_{k\in S'_+}(\mu'_k - \mu_k)$ for different k's. Therefore, we can sort these costs and select those changes with the smallest costs until the constraint is satisfied. We formulate this idea in Algorithm 1. Note that the computations in Algorithm 1 consists of some linear scan components, *i.e.*, computing and selecting Δ's, and a sorting component. Therefore, the overall complexity is $O(K \log K)$.

Now we discuss the case where we add another constraint $\sum_{k=1}^{K} \mathbf{1}\{\mu_k > 0\} \geq 1$ and $\sum_{k=1}^{K} \mathbf{1}\{\mu_k < 0\} \geq 1$ to (1). If the optimal solution $\{\mu_k^\star\}_{k=1}^K$ of (1) already satisfies the new constraint, then there is no modification needed. Otherwise, we discuss three cases corresponding to different relations between B' and B.

First, for $-B \leq B' \leq B$ ($\mu_k^\star = \mu'_k$), if there is no $\mu_k^\star$ equals to -1, then we select j such that the delta increase by changing $\mu_j^\star$ to -1 is the smallest, and set $\mu_j^\star$ to -1. It's easy to show that this new solution always satisfies the constraint $-B \leq \sum_{k=1}^{K} \mu_k^\star \leq B$. We do the analogous step if there is no $\mu_k^\star$ equals to 1. Since the delta changes (with respect to the best possible value $\sum_k f_k(\mu'_k)$) in the objective function is the minimum, the new solution is optimal.

Second, for $B' > B$, we focus on the case of $B > 1$, since there will be at least one $\mu_k^\star = 1$ (since $B^\star = \sum_k \mu_k^\star = B$ or $B-1 > 0$). In addition, as long as S'_- is not empty, then there is at least one $\mu_k^\star = -1$, where $k \in S'_-$. If S'_- is empty and no $\mu_k^\star$ from the solution of (1) equals to -1, we can find the smallest Δ_j from S_Δ when Algorithm 1 terminates, and the largest $\Delta_{i,0}$ from $\bar{S}_\Delta$, and set $\mu_j^\star = -1$ and $\mu_i^\star = 1$.

Third, for $B' < -B$, we can use an analogous procedure to find the optimal solution similar to the case of $B' > B$ by symmetry.

3. Analysis of Generalization Error

In this section we analyze the generalization error of our model based on the techniques developed for perceptron decision tree [2]. Given a model represented by G, for each node $j \in G$, define the *margin* at node j as

$$\gamma_j = \min_{\{\mathbf{x}_i : \mu_{j,y_i} = \pm 1, (\mathbf{x}_i, y_i) \in D\}} |\mathbf{w}_j^T \mathbf{x}_i + b_j| \tag{4}$$

where D is a given training set, $\{\mu_{j,k}\}$ is the coloring at node j and $(\mathbf{w}_j, b_j)$ specifies the decision hyperplane.

Theorem 3.1. *Suppose m random samples are correctly classified using G on K classes containing N_G decision nodes with margins $\{\gamma_j, \forall j \in G\}$, then the generalization error with probability greater than $1-\delta$ is less than*

$$\frac{130R^2}{m}\big(D' \log(4em)\log(4m) + N_G \log(2m) - \log(\frac{\delta}{2})\big) \tag{5}$$

where $D' = \sum_j^{N_G} \frac{1}{\gamma_j^2}$, and R is the radius of a ball containing the distribution's support.

We first discuss the implications of this theorem and then present the proof in Section 4. Theorem 3.1 reveals two important factors to reduce generalization error bound: the first is to enlarge the margin γ_j and the second is to control the complexity of the model by reducing the number of decision nodes N_G. There is a trade-off between these two quantities. Consider two extreme cases: on one hand, to reduce N_G, we want to have a balanced binary tree where each node encourages all classes to participate. However, in this case, the margins are likely to be small since a large subset of classes has to be separated from another subset. On the other hand, to increase the margins, each node can consider only two classes, which is the case of DAGSVM. However, in this case, the number of nodes is quadratic in the number of classes K, while the first case is only linear in K. Therefore, a good balance has to be established to enjoy a good generalization error. Our method characterized by the optimization problem (1) in the main paper is seeking such a good trade-off. The objective function encourages classes to participate to control the model complexity and ignores a subset of classes with small margins to maximize the resulting margins. A particular trade-off between these two parts is controlled by the parameter ρ in our model. Compared to previous methods, for DAGSVM, our method takes advantage of the hierarchical structure in the label space to reduce the model complexity, and for other hierarchical methods, our method achieves larger margins by ignoring a subset of classes at each node and the joint max-margin optimization of the partition and the binary classifier.

Instead of considering the overall error for all classes, we can also bound the generalization error for a specific class k in terms of the nodes in which class k is active. We define the *probability of error for class k* as

$$\begin{aligned}\epsilon_k(G) = P_{(\mathbf{x},y)\sim\mathcal{D}}\{&(y = k \text{ but } f_G(x) \neq k)\\ &\text{or } (y \neq k \text{ but } f_G(x) = k)\}\end{aligned} \tag{6}$$

where $f_G : \mathcal{X} \to \mathcal{Y}$ is the decision function induced by G and $\mathcal{D}$ is some distribution from which $(\mathbf{x}, y)$'s are drawn.

Theorem 3.2. *Suppose in a random m-sample, we are able to correctly distinguish class k from the other classes using G containing N_G decision nodes with margins $\{\gamma_j, \forall j \in G\}$. In addition, suppose the number of nodes in which class k is active is $N_G(k)$. Then with probability greater than $1-\delta$,*

$$\begin{aligned}\epsilon_k(G) \leq \frac{130R^2}{m}\big(&D'_k \log(4em)\log(4m)\\ &+ N_G(k)\log(2m) - \log(\frac{\delta}{2})\big)\end{aligned} \tag{7}$$

where $D'_k = \sum_{j \in k\text{-nodes}} \frac{1}{\gamma_j^2}$ *(k-nodes are the set of nodes in which class* k *is active), and* R *is the radius of a ball containing the distribution's support.*

Please refer to the next section for the proof. This theorem implies that the error for a class k will be small if it has a short decision path with large margins along the path, and vice versa.

4. Proof of Theorems

The proof of Theorem 3.1 and Theorem 3.2 closely resembles the proof of the Theorem 1 in [3] which is based on the technique developed for studying perceptron decision trees [2]. We present the key steps for proving the theorem and more details on this style of proof can be found in [2].

Definition 1. Let $\mathcal{F}$ be a set of real valued functions. We say that a set of points X is γ-shattered by $\mathcal{F}$ relative to $r = (r_x)_{x \in X}$ if there are real numbers r_x indexed by $x \in X$ such that for all binary vectors b indexed by X, there is a function $f_b \in \mathcal{F}$ satisfying $f_b(x) \geq r_x + \gamma$, if $b_x = 1$, and $f_b(x) \leq r_x - \gamma$ otherwise. The *fat shattering dimension* $fat_{\mathcal{F}}$ of the set $\mathcal{F}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite, or infinity otherwise.

We consider the class of linear function: $\mathcal{F}_{\text{lin}} = \{x \to \langle w, x \rangle + \theta : \|w\| = 1\}$.

Theorem 4.1 (Bartlett and Shawe-Taylor [1])**.** *Let* $\mathcal{F}_{lin}$ *be restricted to points in a ball of* n *dimensions of radius* R *about the origin. Then*

$$fat_{\mathcal{F}_{lin}}(\gamma) \leq \min\{\frac{R^2}{\gamma^2}, n+1\} \tag{8}$$

We now give a lemma which bounds the probability of error over a sequence of $2m$ samples where the first m samples $\bar{\mathbf{x}}_1$ have zero error, and the second m samples $\bar{\mathbf{x}}_2$ have error greater than an appropriate ϵ. We denote the model as G and the number of decision nodes as N_G.

Lemma 1. *Let* G *be a model on* K *classes with* N_G *decision nodes with margins* $\{\gamma_1, \gamma_2, \ldots, \gamma_{N_G}\}$*. Let* $k_i = fat_{\mathcal{F}_{lin}}(\frac{\gamma_i}{8})$*, where fat is continuous from the right. Then the following bound holds,* $P^{2m}\{\bar{\mathbf{x}}_1\bar{\mathbf{x}}_2 : \exists$ *a graph* G*:* $\forall x \in \bar{\mathbf{x}}_1$ *and* $\forall i$*,* G *correctly separates classes* S_i^- *and* S_i^+ *at node* i*, then a fraction of points misclassified in* $\bar{\mathbf{x}}_2 > \epsilon(m, K, \delta)\} < \delta$*, where* $\epsilon(m, K, \delta) = \frac{1}{m}(D\log(8m) + \log(\frac{2^{N_G}}{\delta}))$*,* $D = \sum_{i=1}^{N_G} k_i \log(\frac{4em}{k_i})$ *and* S_i^- *and* S_i^+ *are the negative and positive classes at node* i*.*

Proof. The proof follows the same arguments used by the proof of Lemma 3.7 in [2] or the proof of Lemma 4 in [3]. □

Lemma 1 applies to a particular G with specified margins $\{\gamma_i\}$. In practice, we observe the margins after learning the model. To obtain a bound that can be better applied in practice, we want to bound the probabilities uniformly over all possible margins that can arise. This gives rise to Theorem 3.1. The proof follows the same arguments by the proof of Theorem 1 in [3] based on Lemma 1.

Now we proceed to discuss the generalization error bound for the probability of error for a single class k. Let the $S_{k\text{-nodes}}$ be the set of nodes in G where class k is active. We denote the probability of error for $S_{k\text{-nodes}}$ as $\epsilon'_k(G)$, *i.e.*, $\epsilon'_k(G)$ is the probability that any of $i \in S_{k\text{-nodes}}$ makes mistake. We first present a lemma for the generalization error bound for $\epsilon'_k(G)$.

Lemma 2. *Suppose in a random* m*-sample, there is no mistake made by any node* $i \in S_{k\text{-nodes}}$*. In addition,* G *has margins* $\{\gamma_j, \forall j \in G\}$*. Denote* $N_G(k) = |S_{k\text{-nodes}}|$*,* i.e.*, the number of nodes in* G *where class* k *is active. Then with probability greater than* $1 - \delta$*,*

$$\epsilon'_k(G) \leq \frac{130R^2}{m}\big(D'_k \log(4em)\log(4m) + N_G(k)\log(2m) - \log(\frac{\delta}{2})\big) \tag{9}$$

where $D'_k = \sum_{j \in S_{k\text{-nodes}}} \frac{1}{\gamma_j^2}$*, and* R *is the radius of a ball containing the distribution's support.*

Proof. The proof follows exactly Lemma 1 and Theorem 3.1. □

Now we prove the last theorem by exploiting the relation between $\epsilon'_k(G)$ and $\epsilon_k(G)$:

Proof of Theorem 3.2. We first prove that $\epsilon'_k(G) \geq \epsilon_k(G)$. Consider a node $i \in S_{k\text{-nodes}}$, whose positive and negative classes are S_i^+ and S_i^- respectively. We study how a binary error in i relates to the error for class k. Without loss of generality, we assume $k \in S_i^+$. Suppose there is a mistake made on instance x from $S_i^+/\{k\}$, *i.e.*, x is classified as negative by the binary classifier. This implies that x will not be classified as class k which is a correct classification given the definition of the error for class k. Furthermore, if there is a mistake made on instance x from S_i^-, *i.e.*, x is classified as positive in node i, then it is still possible that lower levels will prune away class k for x, leading to correct classification if we are only considering the error for class k. Finally, if there is a binary mistake on instance x from class k, then it also counts as an error in terms of the error for class k. In summary, if there are some binary classification errors in some nodes in $S_{k\text{-nodes}}$, these "errors" may not hold true if we consider the error only for class k. On the other hand, if there is an error for class k either because

we classified instances from class k not as k or because we classified instances from non-k classes as k, there must be at least one binary error in some $i \in S_{k\text{-nodes}}$ (since no error in all binary classifications implies no error for class k). Therefore, $\epsilon'_k(G) \geq \epsilon_k(G)$. With this relation and based on Lemma 2, the statement of Theorem 3.2 is true. □

5. Learned Hierarchy

We present three learned hierarchies using different representations on two datasets. Figure 1 shows the first five levels of the learned hierarchy using dense SIFT based feature with extended Gaussian kernel based on χ^2 distance on the Caltech-256 dataset. Figure 2 shows the first five levels of the learned hierarchy using GIST with RBF kernel on the SUN dataset. Figure 3 shows the first five levels of the learned hierarchy using spatial HOG with histogram intersection kernel on the SUN dataset. Note that the hierarchies learned on the same dataset with different representations are different (see Figure 2 and Figure 3).

References

[1] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel Methods - Support Vector Learning*, 1998. 4

[2] P. Bennett, N. Cristianini, J. Shawe-taylor, and W. D. Enlarging the margins in perceptron decision trees. In *Machine Learning*, 2000. 3, 4

[3] J. C. Platt, N. Cristianini, and J. Shawe-taylor. Large margin dags for multiclass classification. In *NIPS*, 2000. 4

Algorithm 1 An efficient algorithm for solving (3)

Initialize $\forall k,\ \mu_k = \mu'_k$. Let $d = 0$, and let S_Δ and $\bar{S}_\Delta$ be two empty sets.
For each $k \in S'_0$, compute $\Delta_k = f_k(-1) - f_k(0)$, and let $S_\Delta \leftarrow S_\Delta \cup \{\Delta_k\}$.
For each $k \in S'_+$, compute $\Delta_k = f_k(-1) - f_k(0)$, $\Delta_{k,0} = f_k(0) - f_k(1)$ and $\Delta_{k,-} = \Delta_{k,0} + \Delta_k$. If $\Delta_{k,0} \leq \Delta_k$, then let $S_\Delta \leftarrow S_\Delta \cup \{\Delta_{k,0}, \Delta_k\}$. Otherwise, let $S_\Delta \leftarrow S_\Delta \cup \{\frac{\Delta_{k,-}}{2}, \Delta_{k,0}\}$.
repeat
  Pop out the first (minimum) element $\Delta_{\min}$ from S_Δ. $S_\Delta \leftarrow S_\Delta \backslash \{\Delta_{\min}\}$
  if $\Delta_{\min} = \Delta_k$ **then**
    Let $\mu_k = -1$ and $d \leftarrow d + 1$.
  end if
  if $\Delta_{\min} = \Delta_{k,0}$ **then**
    Let $\mu_k = 0$ and $d \leftarrow d + 1$.
  end if
  if $\Delta_{\min} = \frac{\Delta_{k,-}}{2}$ **then**
    if $d \leq B' - B - 2$ **then**
      Let $\mu_k = -1$ and $d \leftarrow d + 2$.
      $S_\Delta \leftarrow S_\Delta \backslash \{\Delta_{k,0}\}$
    else // $d = B' - B - 1$
      Find the next smallest Δ_j or $\Delta_{j,0}$ from S_Δ and let it be $\Delta_{j,\min}$.
      Find the largest Δ_i or $\Delta_{i,0}$ from $\bar{S}_\Delta$, and let it be $\Delta_{i,\max}$ (if there does not exist such i, then let $\Delta_{i,\max} = 0$).
      For $\Delta_{l,-} \in \bar{S}_\Delta$, find the l with the largest $\Delta_{l,-} - \Delta_{l,0}$, denote the largest difference as $\Delta_{l,\max}$ (if $\bar{S}_\Delta$ does not contain $\Delta_{l,-}$, then let $\Delta_{l,\max} = 0$).
      Let $\Delta_{\text{one-step-min}}$ be the minimum of $\{\Delta_{j,\min}, \Delta_{k,-} - \Delta_{i,\max}, \Delta_{k,-} - \Delta_{l,\max}\}$.
      if $\Delta_{\text{one-step-min}} = \Delta_{j,\min}$ **then**
        if $\Delta_{j,\min} = \Delta_j$ **then**
          Let $\mu_j = -1$
        else
          Let $\mu_j = 0$
        end if
        $d \leftarrow d + 1$
      else if $\Delta_{\text{one-step-min}} = \Delta_{k,-} - \Delta_{i,\max}$ **then**
        Let $\mu_k = -1$.
        if i exists **then**
          $\mu_i = \mu'_i$
          $d \leftarrow d + 1$
        else
          $d \leftarrow d + 2$
        end if
      else // $\Delta_{\text{one-step-min}} = \Delta_{k,-} - \Delta_{l,\max}$
        Let $\mu_l = 0$ and $\mu_k = -1$
        $d \leftarrow d + 1$
      end if
    end if
  end if
  $\bar{S}_\Delta \leftarrow \bar{S}_\Delta \cup \{\Delta_{\min}\}$
until $d = B' - B$ or $d = B' - B + 1$
Return: $\{\mu_k\}_{k=1}^K$

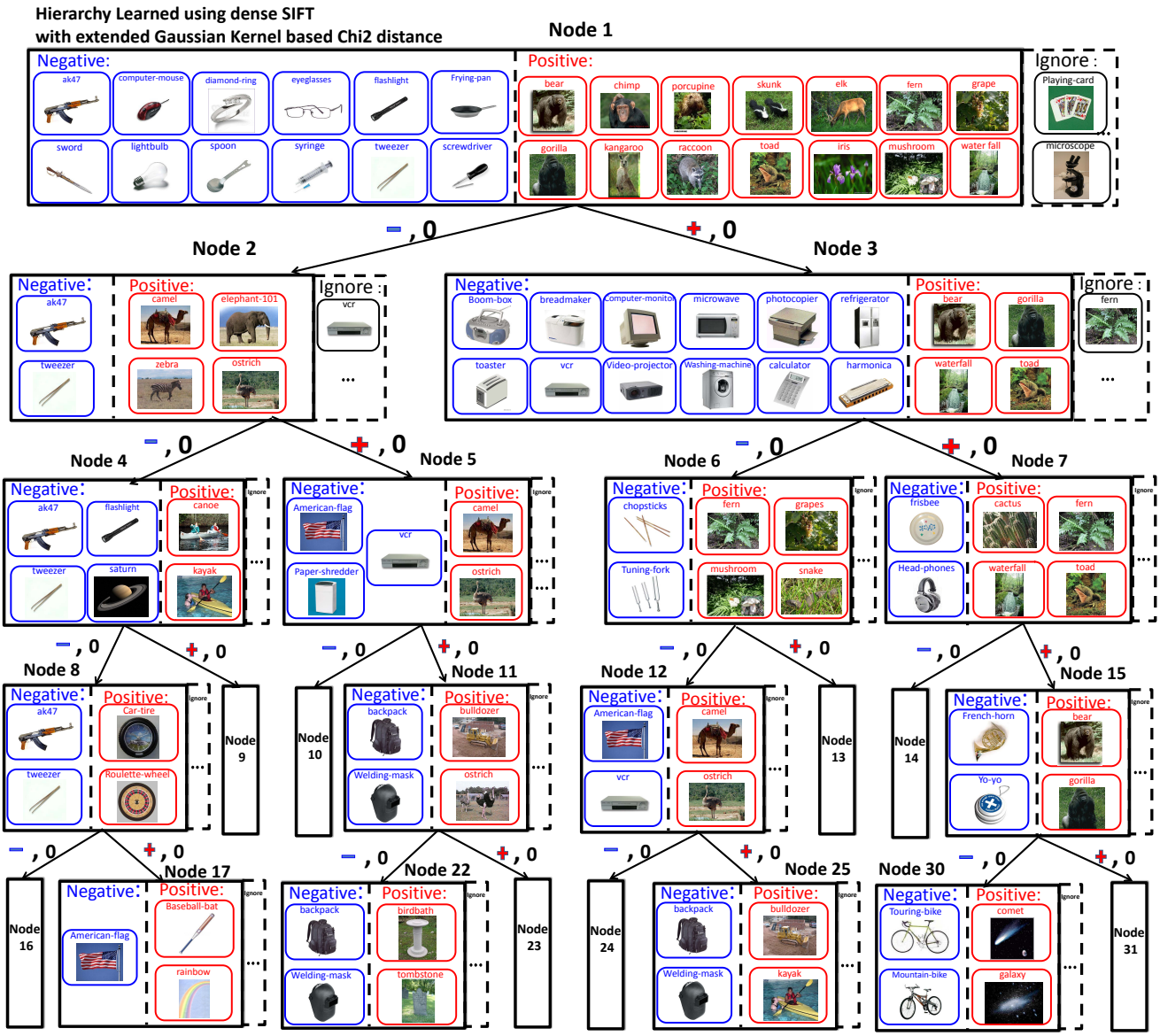

Figure 1. The first five levels of the learned hierarchy using dense SIFT based feature with extended Gaussian kernel based on χ^2 kernel on the Caltech-256 dataset.

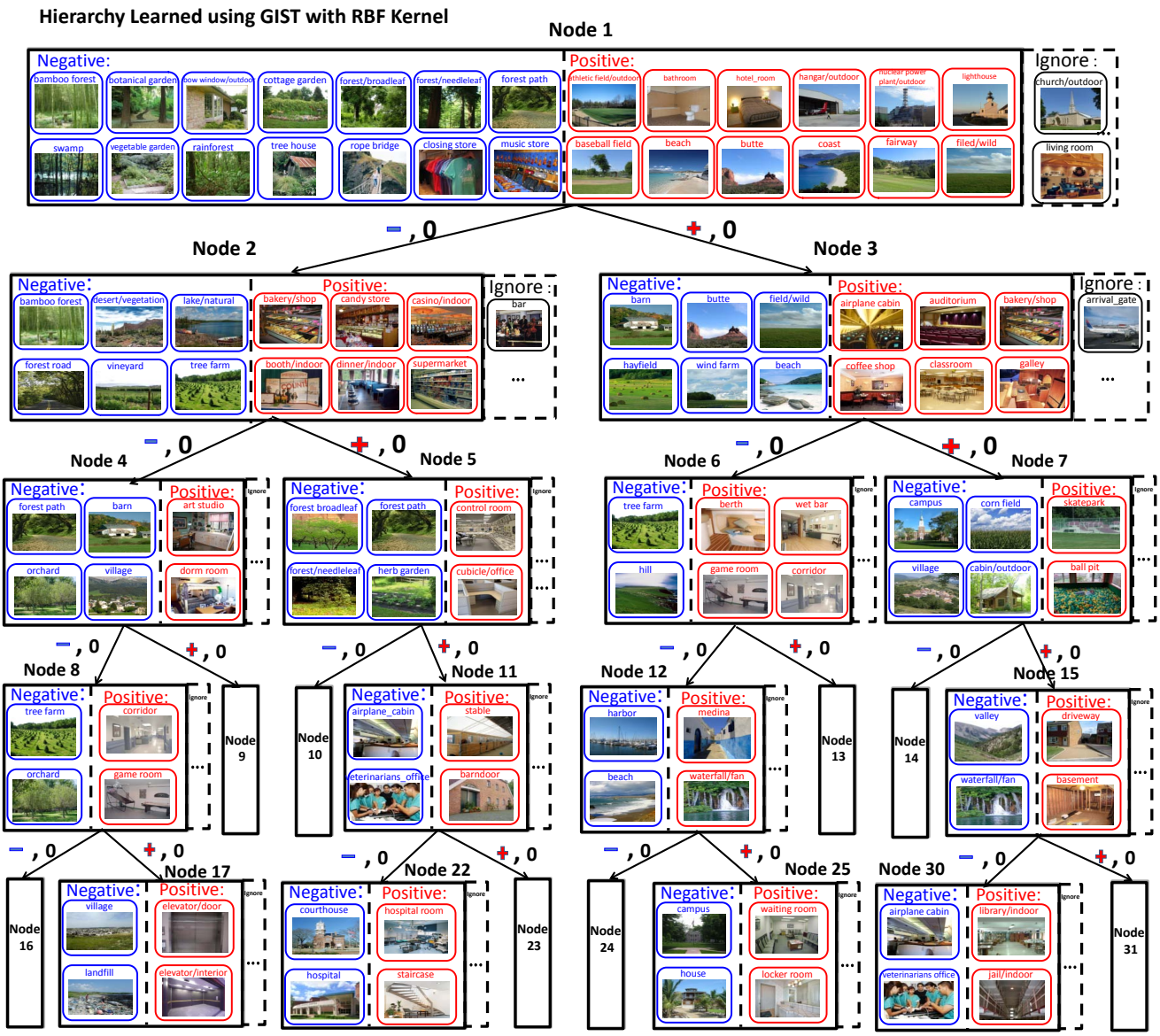

Figure 2. The first five levels of the learned hierarchy using GIST with RBF kernel on the SUN dataset.

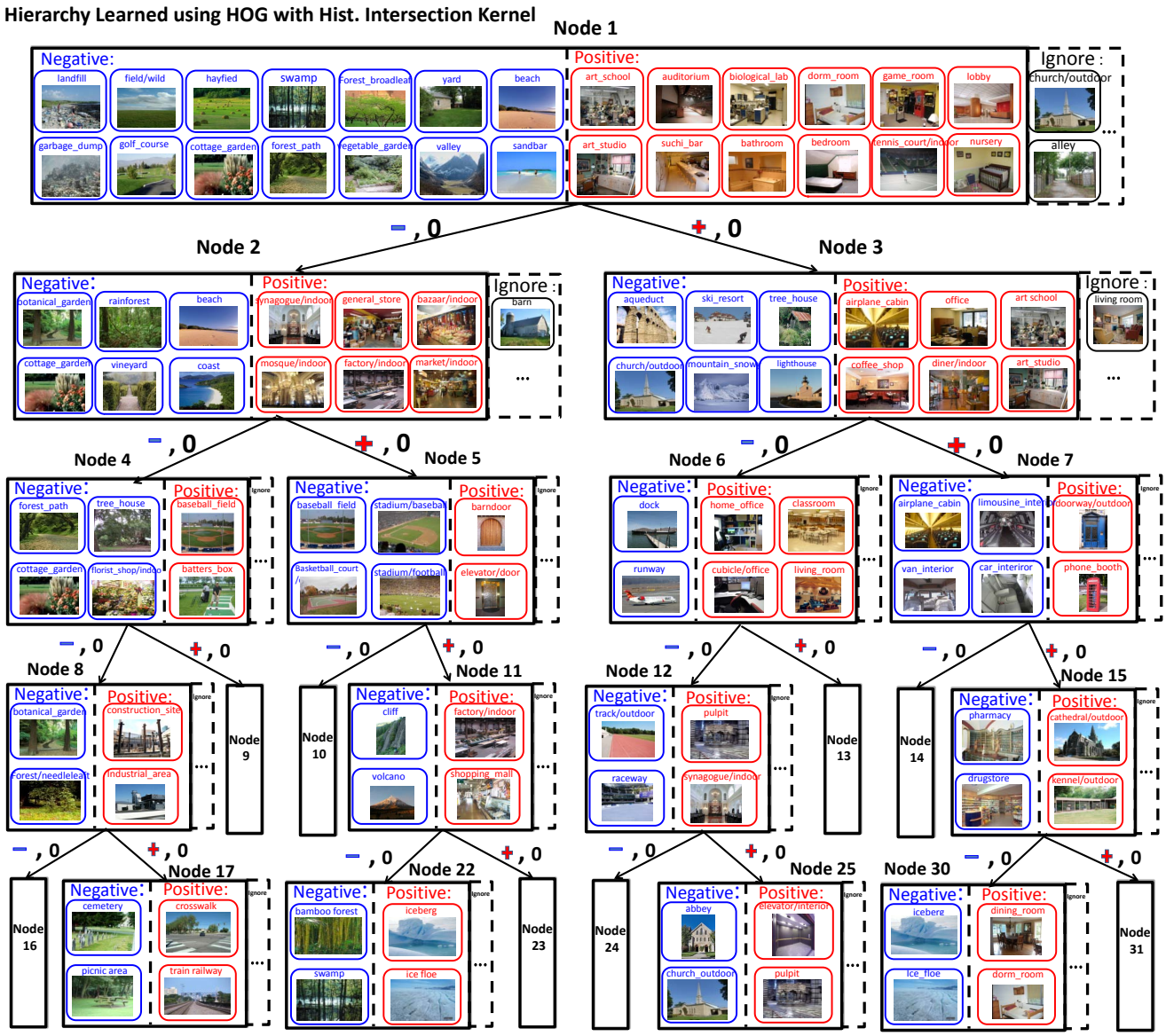

Figure 3. The first five levels of the learned hierarchy using spatial HOG with histogram intersection kernel on the SUN dataset.