Bayesian Analysis (2007) **2**, Number 1, pp. 221–238

Improving Classification When a Class Hierarchy is Available Using a Hierarchy-Based Prior

Babak Shahbaba[*] and Radford M. Neal[†]

Abstract. We introduce a new method for building classification models when we have prior knowledge of how the classes can be arranged in a hierarchy, based on how easily they can be distinguished. The new method uses a Bayesian form of the multinomial logit (MNL, a.k.a. "softmax") model, with a prior that introduces correlations between the parameters for classes that are nearby in the tree. We compare the performance on simulated data of the new method, the ordinary MNL model, and a model that uses the hierarchy in a different way. We also test the new method on page layout analysis and document classification problems, and find that it performs better than the other methods.

Keywords: Hierarchical Classification, Bayesian Models, Multinomial Logistic Regression, Page Layout Analysis, Document Classification

1 Introduction

In this paper, we consider classification problems where classes have a hierarchical structure. The hierarchy reflects our prior opinion regarding similarity of classes. Two classes are considered similar if it is difficult to distinguish them from each other on the basis of the features available. The similarity of classes increases as we descend the hierarchy.

Our original motivation for studying hierarchical classification schemes was prediction of the biological functions of genes. Functions are usually presented in a hierarchical form starting with very general classes (eg, cell processes) and becoming more specific in lower levels of the hierarchy (eg, cell division). Figure 1 shows a small part of the scheme proposed by Riley (1993) to catalogue the proteins of *Escherichia coli*. We discuss this application of our methods elsewhere (Shahbaba and Neal 2006). Here, we discuss this problem more generally, and illustrate its use for two other examples of hierarchical classification. We look at the problem of classifying regions of a page in an article, using classes such as "Section Heading", "Text", or "Figure Caption", which can be arranged in a hierarchy based on distinguishability. We also look at the problem of classifying patent documents relating to textiles, where again the classes can be arranged in a hierarchy in which, for example, a high-level category of "Weaving" contains sub-classes for "Looms" and for "Weavers' Tools".

[*]Dept. of Public Health Sciences, University of Toronto, Toronto, Canada, mailto:babak@stat.utoronto.ca

[†]Dept. of Statistics and
Dept. of Computer Science, University of Toronto, Toronto, Canada, mailto:radford@stat.utoronto.ca

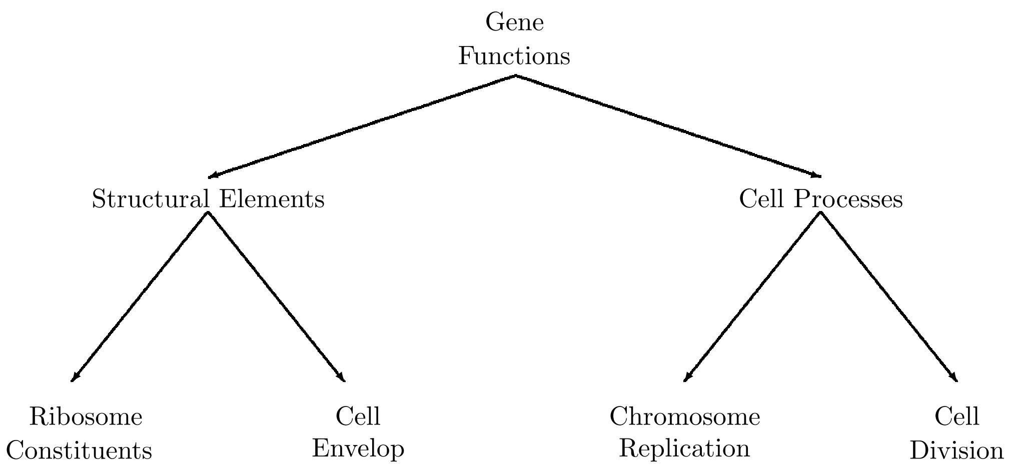

Figure 1: A part of a gene annotation hierarchy proposed by Riley (1993) for the *E. coli* genome.

In a Bayesian model, we can incorporate prior knowledge of the class hierarchy using a suitable prior distribution over parameters of the model. In this paper, we introduce a new method of this sort for the multinomial logit (MNL) model, in which the regression coeffіcents for classes that are nearby in the hierarchy are correlated in the prior.

This paper is organized as follows. In section 2, simple classification models and their extensions for analysing hierarchical classes are discussed. In section 3, using simulated data, we compare the performance of our model, the ordinary MNL model, and an alternative model that uses the hierarchy in a different way. In section 4 we compare the same models on the page region labelling and patent document classification problems. The last section summarizes our findings and presents some ideas for future research.

2 Hierarchical Classification

Consider a classification problem in which we have observed data for n cases, $(x^{(1)}, y^{(1)})$, ...,$(x^{(n)}, y^{(n)})$, where $x^{(i)} = (x_1^{(i)}, ..., x_p^{(i)})$ is the vector of p covariates (features) for case i, and $y^{(i)}$ is the associated class. Our goal is to classify future cases for which the class membership is unknown but the covariates are available. For binary classification problems, a simple logistic model can be used:

$$P(y = 1|x, \alpha, \boldsymbol{\beta}) \quad = \quad \frac{\exp(\alpha + x\boldsymbol{\beta})}{1 + \exp(\alpha + x\boldsymbol{\beta})} \qquad (1)$$

Here, α is the intercept, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown parameters and $x\boldsymbol{\beta}$ is its inner product with the covariate vector.

When there are three or more classes, we can use a generalization known as the multinomial logit (MNL) model (called "softmax" in the machine learning literature):

$$P(y = j|x, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\exp(\alpha_j + x\boldsymbol{\beta}_j)}{\sum_{j'=1}^{c} \exp(\alpha_{j'} + x\boldsymbol{\beta}_{j'})} \tag{2}$$

where c is the number of classes. For each class, j, there is a vector of p unknown parameters $\boldsymbol{\beta}_j$. The entire set of regression coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta_1, ..., \beta_c})$ can be presented as a $p \times c$ matrix. This representation is redundant, since one of the $\boldsymbol{\beta}_j$'s can be set to zero without changing the set of relationships expressible with the model, but removing this redundancy would make it difficult to specify a prior that treats all classes symmetrically. For this model we can use the following priors:

$$\begin{aligned} \alpha_j|\eta &\sim N(0, \eta^2) \\ \beta_{jl}|\tau &\sim N(0, \tau^2) \\ \eta^{-2} &\sim Gamma(v, V) \\ \tau^{-2} &\sim Gamma(w, W) \end{aligned}$$

where $j = 1, ..., c$ and $l = 1, ..., p$.

The MNL model treats classes as unrelated entities without any hierarchical structure. This is not always a realistic assumption. In many classification problems, like those discussed above, one can arrange classes in a hierarchical form analogous to the hierarchy of species arranged in genera, families, etc. If the classes have in fact the assumed structure, one would expect to obtain a higher performance by using this additional information. A special case is when the classes are ordered (e.g., education level). For these problems a more parsimonious model (e.g., cumulative logit model) with improved power can be used (Agresti 2002).

The importance of using the hierarchy in classification models has been emphasized by many authors (e.g., Sattath and Tversky 1977; Fox 1997; Koller and Sahami 1997). One approach for modelling hierarchical classes is to decompose the classification model into nested models (e.g., logistic or MNL). Nested MNL models are extensively discussed in econometrics (e.g., Sattath and Tversky 1977; McFadden 1980) in the context of estimating the probability of a person choosing a specific alternative (i.e., class) from a discrete set of options (e.g., different modes of transportation). These models, known as discrete choice models, aim at forecasting and explaining human decisions through optimizing an assumed utility (preference) function, which is different from our aim of maximizing classification accuracy.

Goodman (2001) showed that using hierarchical classes can significantly reduce the training time of maximum entropy-based language models and results in slightly lower perplexities. He illustrated his approach using a word labelling problem, and recommended that instead of predicting words directly, we first predict the category to which the word belongs, and then predict the word itself. Such a two-level hierarchical model was also used by Weigend *et al.* (1999) for document classification. They evaluated their model on the Reuters-22173 corpus and showed significant improvement, especially for rare classes.

For hierarchical classification problems with simple binary partitions, Fox (1997) suggested using successive logistic models for each binary class. In Figure 2 below, for example, these partitions are {12, 34}, {1, 2}, and {3, 4}. The resulting nested binary models are statistically independent, conditioned on the upper levels. The likelihood can therefore be written as the product of the likelihoods for each of the binary models. For example, in Figure 2 we have

$$P(y = 1|x) \quad = \quad P(y \in \{1, 2\}|x) \times P(y \in \{1\}|y \in \{1, 2\}, x) \tag{3}$$

Restriction to binary models is unnecessary. At each level, classes can be divided into more than two subsets and MNL can be used instead of logistic regression. We refer to methods based on decomposing the tree structure into nested MNL models as treeMNL. Consider a parent node, m, with c_m child nodes, representing sets of classes defined by S_k, for $k = 1, ..., c_m$. The portion of the nested MNL model for this node has the form:

$$\begin{aligned}
P(y \in S_k|x, \boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) &= \frac{\exp(\alpha_{mk} + x\boldsymbol{\beta}_{mk})}{\sum_{k'=1}^{c_m} \exp(\alpha_{mk'} + x\boldsymbol{\beta}_{mk'})} \\
\alpha_{mk}|\eta_m &\sim N(0, \eta_m^2) \\
\beta_{mkl}|\tau_m &\sim N(0, \tau_m^2) \\
\eta_m^{-2} &\sim Gamma(v_m, V_m) \\
\tau_m^{-2} &\sim Gamma(w_m, W_m)
\end{aligned}$$

where $l = 1, ..., p$. We calculate the probability of each end node, j, by multiplying the probabilities of all intermediate nodes leading to j.

In contrast to this treeMNL model, Mitchell (1998) showed that the hierarchical naive Bayes classifier is equivalent to the standard non-hierarchical classifier when the probability terms are estimated by maximum likelihood. To improve the hierarchical naive Bayes model, McCallum *et al.* (1998) suggested to smooth parameter estimates of each end node by shrinking its maximum likelihood estimate towards the estimates of all its ancestors in the hierarchy. More recently, new hierarchical classification models based on large margin principals, specifically support vector machine (SVM), have been proposed (Dumais and Chen 2000; Dekel *et al.* 2004; Cai and Hoffmann 2004; Tsochantaridis *et al.* 2004; Cesa-Bianchi *et al.* 2006). Dekel *et al.* (2004) introduced a large margin hierarchical classification model that uses the sum of parameters along the tree for classifying cases to the end nodes. These parameters are estimated based on a set of classifiers that assign cases to the intermediate nodes. Cai and Hoffmann (2004) suggested a similar approach based on the generalization of multiclass SVM. We also use sums of parameters along paths in the tree, but in a rather different way from this past work.

Our new framework for modeling hierarchical classes is illustrated in Figure 2, which shows a hierarchical classification problem with four classes. For each branch in the hierarchy, we define a different set of parameters. In Figure 2, these parameters are denoted as $\boldsymbol{\phi}_{11}$ and $\boldsymbol{\phi}_{12}$ for branches in the first level and $\boldsymbol{\phi}_{21}$, $\boldsymbol{\phi}_{22}$, $\boldsymbol{\phi}_{23}$ and $\boldsymbol{\phi}_{24}$ for branches in the second level. We assign objects to one of the end nodes using an MNL model (Equation 2) whose regression coefficients for class j are represented by the sum of parameters on all the branches leading to that class. In Figure 2, these coefficients are $\boldsymbol{\beta_1} = \boldsymbol{\phi_{11}} + \boldsymbol{\phi_{21}}$, $\boldsymbol{\beta_2} = \boldsymbol{\phi_{11}} + \boldsymbol{\phi_{22}}$, $\boldsymbol{\beta_3} = \boldsymbol{\phi_{12}} + \boldsymbol{\phi_{23}}$ and $\boldsymbol{\beta_4} = \boldsymbol{\phi_{12}} + \boldsymbol{\phi_{24}}$ for classes $1, 2, 3$ and 4 respectively. Sharing the common terms, $\boldsymbol{\phi_{11}}$ and $\boldsymbol{\phi_{12}}$, introduces prior correlation between the parameters of nearby classes in the hierarchy.

In our model, henceforth called corMNL, $\boldsymbol{\phi}$'s are vectors with the same size as $\boldsymbol{\beta}$'s. We assume that, conditional on higher level hyperparameters, all the components of the $\boldsymbol{\phi}$'s are independent, and have normal prior distributions with zero mean. The variances of these components are regarded as hyperparameters, which control the magnitudes of coefficients. We use one hyperparameter for each non-terminal node. When a part of the hierarchy is irrelevant, we hope the posterior distribution of its corresponding hyperparameter will be concentrated near zero, so that the parameters it controls will also be close to zero.

In detail, the prior we use is as follows:

$$\begin{aligned} \alpha_j | \eta &\sim N(0, \eta^2) \\ \phi_{mkl} | \tau_m &\sim N(0, \tau_m^2) \\ \eta^{-2} &\sim Gamma(v, V) \\ \tau_m^{-2} &\sim Gamma(w_m, W_m) \end{aligned}$$

Here, $j = 1, \ldots, c$ indexes classes, and ϕ_{mkl} refers to the parameter related to covariate x_l and branch k of node m. The ϕ parameters of all the branches that share the same node are controlled by one hyperparameter, τ_m, which controls the degree to which that portion of the hierarchy is active. In figure 2, for example, when the hyperparameters in the first level are small (compared to the hyperparameters in the second level), the model reduces to simple MNL. In contrast, when these hyperparameters are relatively large, the model reinforces our assumption of hierarchical classes.

By introducing prior correlations between parameters for nearby classes, we can better handle situations in which these classes are hard to distinguish. If the hierarchy actually does provide information about how distinguishable classes are, we expect that performance will be improved. This would be especially true when the training set is small and the prior has relatively more influence on the results. Using an inappropriate hierarchy will likely lead to worse performance than a standard MNL model, but since the hyperparameters can adapt to reduce the prior correlations to near zero, the penalty may not be large.

3 Results for Synthetic Datasets

So far, we have discussed three alternative models: MNL, treeMNL, and corMNL. We first compare these models using a synthetic four-way classification problem with two covariates. Data are generated from each of these models in turn, and then fit with each model in order to test the robustness of the models when applied to data generated from other models.

All regression parameters are given normal priors with mean zero. For the MNL model, the standard deviation for all the intercepts, η, and the standard deviation for the rest of coefficients, τ, have the following priors:

$$\begin{array}{lcll} \eta^{-2} & \sim & Gamma(1,10) & (0.16, 0.38, 1.98) \\ \tau^{-2} & \sim & Gamma(1,1) & (0.52, 1.20, 6.27) \end{array}$$

We use the parameterization of the Gamma distribution in which Gamma(a,b) has density $f(x|a,b) = [b^a\Gamma(a)]^{-1}x^{a-1}e^{-x/b}$, for which the mean is ab and the standard deviation is $a^{1/2}b$. The 2.5, 50 and 97.5 percentiles of τ and η are shown in parenthesis.

For the treeMNL and corMNL models, we assume that classes are arranged in a hierarchical form as shown in Figure 2. This hierarchy implies that while it might be easy to distinguish between groups {1, 2} and {3, 4}, further separation of classes might not be as easy. As mentioned above, the treeMNL model for this hierarchy is comprised of three nested logistic models. These models are: $P(y \in \{1,2\}|\alpha_1, \boldsymbol{\beta_1}, x)$, $P(y = 1|\alpha_2, \boldsymbol{\beta_2}, x, y \in \{1,2\})$ and $P(y = 3|\alpha_3, \boldsymbol{\beta_3}, x, y \in \{3,4\})$. The priors for the treeMNL and corMNL models are discussed in section 2. The variances of the regression parameters, β, in treeMNL and of the ϕ's in corMNL are regarded as hyperparameters. For these two models, one hyperparameter controls all the parameter emerging from the same node. These hyperparameters are given the following prior distributions:

$$\begin{array}{lcll} \eta^{-2} & \sim & Gamma(1,10) & (0.16, 0.38, 1.98) \\ \tau_1^{-2} & \sim & Gamma(1,5) & (0.23, 0.54, 2.82) \\ \tau_2^{-2} & \sim & Gamma(1,20) & (0.05, 0.12, 0.63) \\ \tau_3^{-2} & \sim & Gamma(1,20) & (0.05, 0.12, 0.63) \end{array}$$

Here, τ_1, τ_2 and τ_3 correspond to nodes 1, 2, and 3 respectively (Figure 2). These parameters have a narrower prior compared to τ in the MNL model. This is to account for the fact that the role of β in the MNL model is played by more than one parameter in treeMNL and corMNL. Moreover, the regression parameters in the second level of hierarchy have a relatively smaller standard deviation τ. As a result, these parameters tend to be smaller, making separation of class 1 from 2 and class 3 from 4 more difficult.

We do three tests, in which we assume that each of the MNL, treeMNL and corMNL is the correct model. This allows us to see how robust each model is when data actually come from a somewhat different model. For each test, we sample a set of parameters from the prior distribution of the corresponding model. Pairs of data items $(x^{(i)}, y^{(i)})$ are generated by first drawing 10000 independent samples $x_1^{(i)}, x_2^{(i)}$ from the uniform$(-5,5)$ distribution and then assigning each data item to one of the four possible classes. The assignment is either based on a multinomial model (for data generated from MNL and corMNL) or based on successive logistic models (for data generated from treeMNL).

All three models were trained on the first 100 data items and tested on the remaining 9900 items. The regression coefficients were sampled from their posterior distribution using MCMC methods with single-variable slice sampling (Neal 2003), using the "stepping out" procedure to find an interval around the current point, and then the

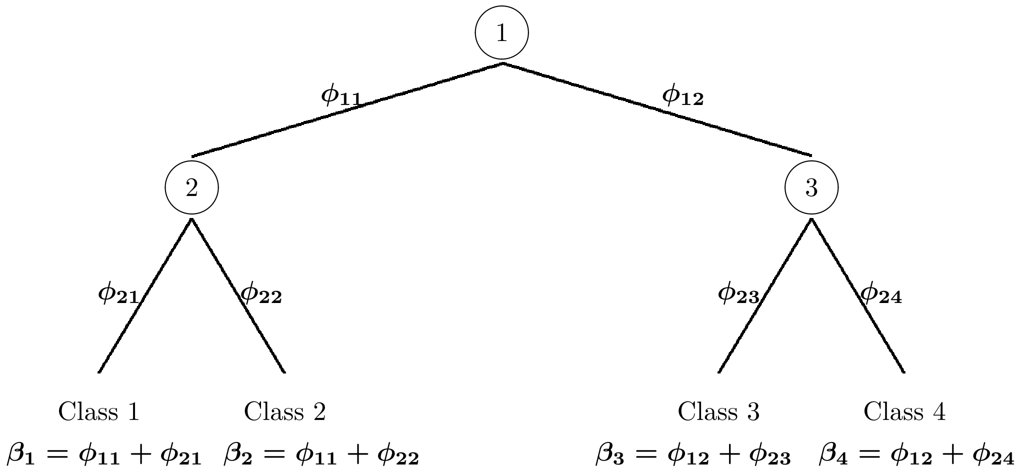
1
2
3
ϕ_{11}
ϕ_{12}
ϕ_{21}
ϕ_{22}
ϕ_{23}
ϕ_{24}
Class 1
Class 2
Class 3
Class 4
$\beta_1 = \phi_{11} + \phi_{21}$
$\beta_2 = \phi_{11} + \phi_{22}$
$\beta_3 = \phi_{12} + \phi_{23}$
$\beta_4 = \phi_{12} + \phi_{24}$

Figure 2: The simple model used for the simulation study. The coefficient parameters for each classes are presented as a sum of parameters at different level of hierarchy.

N=100		**Data**					
		MNL		treeMNL		corMNL	
		log-prob	acc	log-prob	acc	log-prob	acc
Method	MNL	**-0.7958**	**67.1**	-0.8918	58.4	-0.9168	59.4
	treeMNL	-0.8489	65.0	**-0.8770**	**58.7**	-0.9113	59.4
	corMNL	-0.7996	67.1	-0.8797	58.6	**-0.9075**	**59.5**

Table 1: Comparison of models on simulated data created based on Figure 2. Average log-probability (log-prob) and accuracy rate (acc) are estimated on the test sets.

"shrinkage" procedure to sample from this interval. Since the hyperparameters were given conjugate priors, direct Gibbs sampling could be used for them. For all tests we ran 1000 MCMC iterations to sample from the posterior distributions. We discarded the initial 250 samples and used the rest for prediction. Performance on the test set is measured in terms of average log-probability (based on the estimated probability of the correct class) and accuracy rate (defined as the percentage of the times the correct class is predicted). We make predictions based on the posterior predictive probabilities.

The above procedure was repeated 100 times. Each time, new regression parameters were sampled from priors and new pairs of data items were created based on the assumed models. The average results (over 100 repetitions) are presented in Table 1. In this table, each column corresponds to the model used for generating the data and each row corresponds to the model used for building the classifier. As we can see, the diagonal elements have the best performance in each column. That is, the model whose functional form matches the data generation mechanism performs significantly better

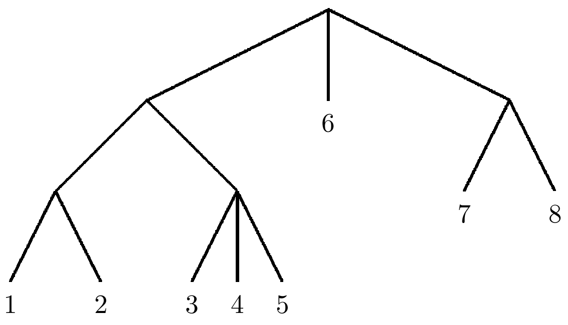

Figure 3: A hypothetical hierarchy with a more complex structure.

N=100		**Data**					
		MNL		treeMNL		corMNL	
		log-prob	acc	log-prob	acc	log-prob	acc
Method	MNL	**-0.2539**	**89.9**	-0.3473	87.7	-0.3106	88.7
	treeMNL	-0.6837	76.9	-0.2898	**90.3**	-0.3614	87.9
	corMNL	-0.2910	89.7	**-0.2854**	90.1	**-0.2841**	**90.3**

Table 2: Comparison of models on simulated data created based on Figure 3. Average log-probability (log-prob) and accuracy rate (acc) are estimated on the test sets.

than the other two models (all p-values based on average log-probability are less than 0.01 using a paired t-test with $n = 100$). Moreover, the results show that when the samples are generated according to the MNL model (i.e., classes are unrelated), corMNL has a significantly (p-value < 0.001) better performance compared to treeMNL. When treeMNL is used to generate data, corMNL performs only slightly worse than treeMNL. The conclusions remain the same when we use different priors and ranges of covariates.

While statistically significant, the results presented in Table 1 might not be significant for some practical purposes. This is mostly due to the simplicity of the hierarchical structure. We repeated the above tests with a more complex hierarchy, shown in Figure 3. For this problem we used four covariates randomly generated from the uniform(0, 1) distribution. In all three models, we used the same prior as before for the intercepts. For the MNL model we set $\tau^{-2} \sim Gamma(1, 1)$. The hyperparameters of treeMNL and corMNL were given $Gamma(1, 5)$, $Gamma(1, 20)$ and $Gamma(1, 100)$ priors for the first, second and third level of the hierarchy respectively.

Table 2 shows the average results over 100 datasets for each test. As we can see, the differences between models are more accentuated. When data are generated by other models, corMNL again performs well, being outperformed only by the true model. When data come from treeMNL, the results from corMNL are very close to those of the true model (i.e., treeMNL), and are actually better for log-probability, though this must of course be due to chance, and is not statistically significant. In contrast, treeMNL's performance on data generated by corMNL is substantially worse than corMNL's performance, and is also worse than that of the non-hierarchical MNL model.

4 Results for Real Datasets

In this section, we test our approach on two real classification tasks. The first task is labelling the regions of a page using a predefined set of labels. The dataset used for this problem was collected by Laven *et al.* (2005) and is derived from the page images of 58 articles (460 pages) appearing in the proceedings of the Neural Information Processing Systems (NIPS) conference in 2001 and 2002. The second task is classification of documents into different groups based on their contents. For this task we use patent documents released by World Intellectual Property Organization (WIPO). The MATLAB files for MNL, treeMNL and corMNL, along with the NIPS dataset, are available online at `http://www.utstat.utoronto.ca/~babak`.

4.1 Performance Measures

To compare the performance of different models, we use average log-probability and accuracy rate as described above. We also employ several other measurements including macroaverage F_1 (van Rijsbergen 1972), parent accuracy, precision, and taxonomy-based loss (Cai and Hoffmann 2004). F_1 is a common measurement in document labelling and is defined as:

$$F_1 = \frac{1}{J}\sum_{j=1}^{J}\frac{2A_j}{2A_j + B_j + C_j}$$

where A_j is the number of cases which are correctly assigned to class j, B_j is the number cases incorrectly assigned to class j, and C_j is the number of cases which belong to the class j but are assigned to other classes. The taxonomy-based loss is equal to half the distance between the predicted and the actual class in the tree. Parent accuracy is accuracy after grouping the end nodes with the same parent. While accuracy measurements are based on the top-ranked (based on the posterior predictive probabilities) category only, precision measures the quality of ranking and is defined as follows:

$$\mathit{precision} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{|y : P(y|x^{(i)}) \geq P(y^{(i)}|x^{(i)})|}\right)$$

Here, y ranges over all classes and $y^{(i)}$ is the correct class of case i. The denominator is, therefore, the number of classes with equal or higher rank compared to the correct class.

Except for average log-probability and precision, all these measurements require that each test case be assigned to one specific class. For this purpose, we assigned each test case to the end node with the highest posterior predictive probability, as would be

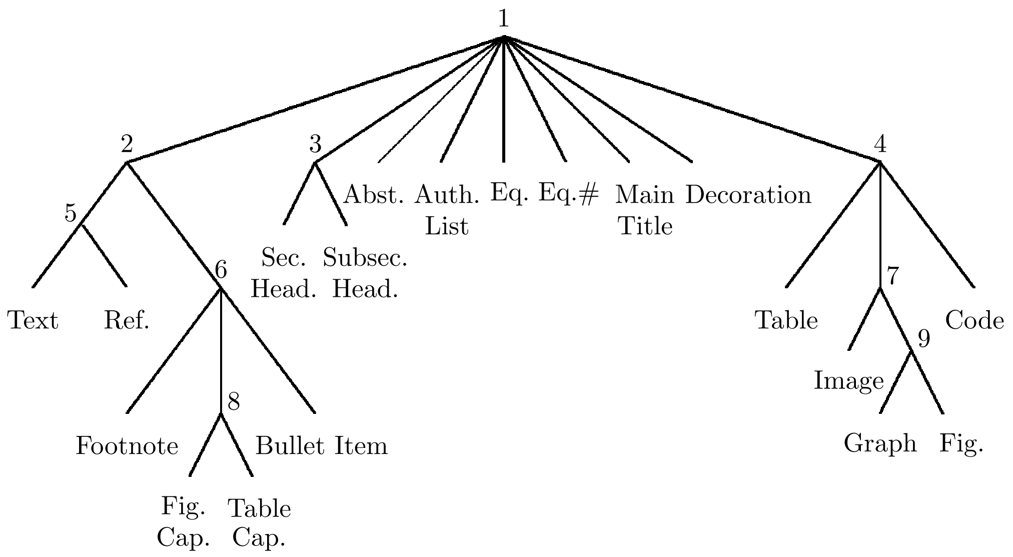

Figure 4: Hierarchical structure of document labels.

optimal for a simple 0/1 loss function. It is, of course, possible to tailor predictions to different performance measures. For example, to improve the parent accuracy, we can still use the 0/1 loss function but make prediction based on the posterior predictive probability of parent nodes. For the taxonomy-based loss, we can predict the class that minimizes the expected distance when a case is misclassified. We tried these tailored predictions, but since the improvements were negligible, we report only the results based on classifying to the end node with highest predictive probability.

To provide a baseline for interpreting the results, for each task we present the performance of a model that ignores the covariates and whose likelihood is solely based on the observed frequency of classes. For this model, we use a vague Dirichlet prior with parameter 1 for all classes. This is a conjugate prior for multinomial parameters. The posterior distribution is also a Dirichlet distribution with parameter $n_j + 1$ for class j. Here, n_j is the frequency of class j in the training set.

4.2 NIPS Dataset

As mentioned above, the NIPS dataset contains page images of 58 articles. Each page was segmented to several regions, and each region was manually classified to one of 19 possible classes. Figure 4 presents these classes in a hierarchical form. The hierarchy is based on our beliefs regarding how difficult it is to separate classes from each other using the available covariates.

The covariates are 59 features such as the location of the region on the page and the density of the ink inside the region. We normalized all features so they have zero mean and standard deviation 1. Laven *et al.* (2005) considered the items from the same article as independent even though they clearly are not. Although this may cause overfitting problems, we follow the same assumption in order to produce comparable results.

Laven *et al.* (2005) divided the dataset into a training set (3445 regions), and a test set (1473 regions). We also trained our three models (MNL, corMNL and treeMNL) on the same training set and evaluated performance on the test set.

The coefficient parameters in the MNL models were given normal priors with mean zero. The variances of these parameters were regarded as hyperparameters. For this problem, since the number of covariates, $p = 59$, is relatively large, we use the Automatic Relevance Determination (ARD) method suggested by Neal (1996). ARD employs a hierarchical prior to determine how relevant each covariate is in classification of objects. In this method, one hyperparameter, σ_l, is used to control the variance of all coefficients, β_{jl} $(j = 1, ..., c)$, for covariate x_l. If a covariate is irrelevant, its hyperparameter will tend to be small, forcing the coefficients for that covariate be near zero. We also use one hyperparameter, τ, to control the overall magnitude of the β's in the MNL model, so that the the standard deviation of β_{jl} is equal to $\tau\sigma_l$. Therefore, while σ_l controls the relevance of covariate x_l compared to other covariates, τ, controls the overall usefulness of all covariates in separating classes. In detail, the prior for the MNL model was as follows:

$$\begin{array}{rcll} \alpha_j|\eta & \sim & N(0, \eta^2) & j = 1, ..., 19 \\ \beta_{jl}|\tau, \sigma_l & \sim & N(0, \tau^2\sigma_l^2) & l = 1, ..., 59 \\ \eta^{-2} & \sim & Gamma(0.5, 1) & (0.63, 2.09, 46.31) \\ \tau^{-2} & \sim & Gamma(0.5, 20) & (0.14, 0.47, 10.07) \\ \sigma_l^{-2} & \sim & Gamma(1, 10) & (0.16, 0.38, 1.98) \end{array}$$

Similar priors are used for the parameters of treeMNL and corMNL. For these two models, we again used one hyperparameter, $\sigma_l^{-2} \sim Gamma(1, 10)$ to control all parameters related to covariate x_l. We also used one scale parameter $\tau_m^{-2} \sim Gamma(0.5, 100)$, with 2.5, 50 and 97.5 percentiles of 0.06, 0.21 and 5.57, for all parameters (β's in treeMNL, ϕ's in corMNL) sharing the same node m. The prior for the intercepts was the same as in the MNL model.

We used Hamiltonian dynamics (Neal 1993) for sampling from the posterior distribution of coefficients. To reduce the random walk aspect of sampling procedure, we use a reasonably large number of leapfrog steps ($L = 500$). The stepsizes, ϵ's, are set to 0.02 in order to maintain an acceptance rate of about 90%. In the MNL and corMNL models, new values are proposed for all regression parameters simultaneously. Nested MNL models in treeMNL are updated separately since they are regarded as independent models. The coefficient parameters within each nested model, however, are updated at the same time. Gibbs sampling was used for sampling from the posterior distribution of hyperparameters. Convergence of the Markov chain simulations was assessed by plotting the values of hyperparameters and the average log-likelihood (on training cases). We ran each chain for 2500 iterations, of which the first 500 were discarded. Simulating the

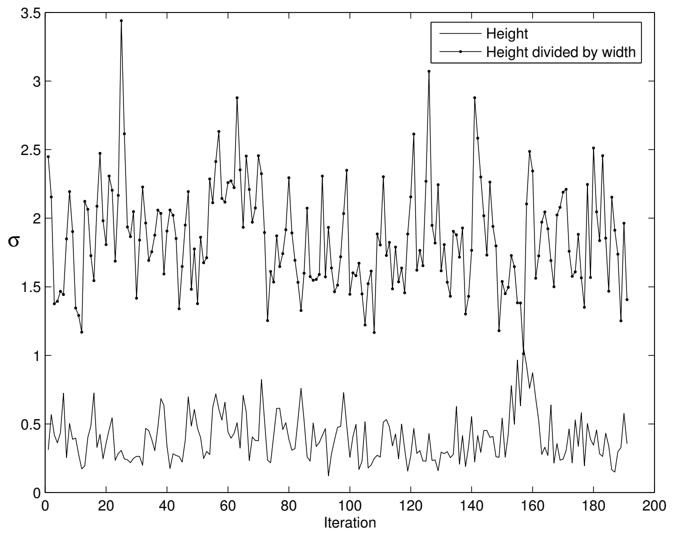

Figure 5: Trace plots of the ARD hyperparameters, σ_l, for two covariates of the corMNL model applied to the NIPS dataset.

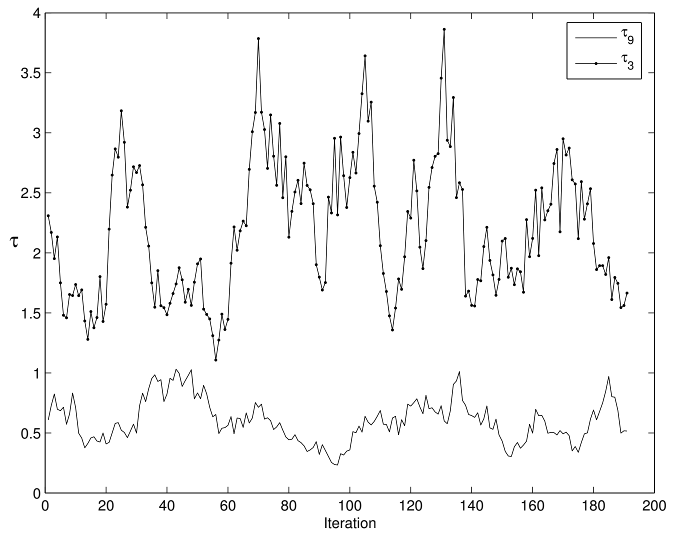

Figure 6: Trace plots of two overall scale hyperparameters, τ_m, for two nodes of the corMNL model applied to the NIPS dataset.

	log-prob	acc (%)	pacc (%)	precision (%)	F_1 (%)	Δ-loss
Baseline	-2.38	29.4	44.1	47.2	2.4	1.44
LR (ML)	–	88.1	–	–	–	–
MNL	-0.429	88.8	93.0	93.1	74.6	0.22
treeMNL	-0.466	87.2	91.7	92.3	68.9	0.23
corMNL	**-0.405**	**89.5**	**93.6**	**93.5**	**76.0**	**0.20**

Table 3: Performance of models based on NIPS dataset. Here, “acc”, “pacc” and “Δ-loss” refer to accuracy, parent accuracy and taxonomy-based loss respectively. Larger values are better except for Δ-loss.

Markov chain for 10 iterations took about 5 minutes for MNL, 4 minutes for treeMNL and 9 minutes for corMNL, using a MATLAB implementation on an UltraSPARC III machine.

Table 3 compares the results from different models. In this table, we present the logistic regression (LR) model (based on maximum likelihood) developed by Laven *et al.* (2005) as the benchmark. As we can see, the corMNL model outperforms all other models. In contrast, treeMNL performs worse than the non-hierarchical MNL model.

To illustrate the effect of hyperparameter σ in identifying relevant features, in Figure 5 we show the trace plots of σ_l for two covariates in the corMNL model. These hyperprameters correspond to two features of a region: “height” and “height divided by width”. The latter is clearly a more relevant feature for this task. To show the effects of the τ_m hyperparemeters, in Figure 6 we present the trace plots of τ_3 and τ_9 (i.e., scale parameter of node 3 and node 9 in Figure 4). As we can see, the scale parameter in node 3 is large compared to the scale parameter in node 9.

4.3 WIPO-alpha Dataset

We also evaluated our models on the WIPO-alpha dataset of patent documents (available at http://www.wipo.int/ibis/datasets). These documents are classified according to a standard taxonomy known as the International Patent Classification (available at http://www.wipo.int/classifications/en/). The classes in this taxonomy are arranged in a four-level tree structure. At the highest level, documents are divided to 8 sections. For our experiment, we use the documents in section “D” (textile; paper), which has 1710 documents and a total of 160 classes. A pre-processed dataset based on these documents was provided by Cai and Hoffmann (2004). This dataset was generated by indexing the title and claim contents. Document parsing, tokenization, and term normalization were performed using the MindServer retrieval engine. The result is a set of word (i.e., token) counts for each document. Overall, there are 18077 unique words, whose counts are used as covariates. We use a square-root transformation for these covariates in order to emphasize more the occurrence of a word rather than its frequency.

The number of covariates is quite large compared to the number of documents.

	log-prob	acc (%)	pacc (%)	precision (%)	F_1 (%)	Δ-loss
Baseline	-4.492	3.6	13.8	12.1	0.04	2.47
SVM	–	41.8	65.4	52.3	–	1.20
hSVM	–	42.8	69.1	54.4	–	1.08
MNL	-2.622	42.0	66.9	55.1	18.4	1.10
treeMNL	-2.408	42.3	68.7	56.0	17.7	1.05
corMNL	**-2.397**	**43.4**	**69.8**	**56.9**	**18.7**	**1.02**

Table 4: Performance of models based on WIPO-alpha dataset, section "D". Here, "acc", "pacc" and "Δ-loss" refer to accuracy, parent accuracy and taxonomy-based loss respectively. The SVM and hierarchical SVM (hSVM) are developed by Cai and Hoffmann (2004). Larger values are better except for Δ-loss.

Cai and Hoffmann (2004) devised a variable selection strategy which provides an optimal solution according to their SVM model. Here, we apply Principal Component Analysis (PCA). We first centred the covariates but did not scale them to have variance one. We then projected the covariates on the 300 principal component directions with the highest variance, and used these 300 principal component scores as covariates for our models, rather than the original covariates. For all models, we use the same priors as discussed in section 4.2, with the exception of ARD hyperparameters, σ_l. For these hyperparameters, we used $\sigma_l^{-2} \sim Gamma(2, 1)$, where $l = 1, ..., 300$ (2.5, 50 and 97.5 percentiles of σ_l are 0.45, 0.78 and 1.98). Compared to the priors used in section 4.2, these priors are more concentrated close to 1. We used these priors to minimize the effect of the ARD hyperparameters since the task of variable selection and relevance determination are mainly performed through PCA. As before, Hamiltonian dynamics ($L = 100$ and $\epsilon = 0.005$) and Gibbs sampling were used for sampling from the posterior distribution of coefficients and hyperparameters respectively. We ran each chain for 3000 iterations and discarded the first 500 iterations. For 10 iterations, the Markov chain simulations took about 6 minutes for MNL, 2 minutes for treeMNL and 10 minutes for corMNL.

Table 4 compares the performance of different models. Following Cai and Hoffmann (2004), the results are presented based on a three-fold cross-validation where singular classes (i.e., $n_j = 1$) appear only in the training set. As we can see, the corMNL model outperforms both MNL and treeMNL. Although the results reported for the corMNL model are better than the hiearchical SVM (hSVM) developed by Cai and Hoffmann (2004), the difference could be due to several factors, such as randomness of cross-validation, transformation of variables, or the efficiency of variable selection strategy, as well as the different approach to using the hierarchy. For this problem, treeMNL did by most measures improve on the non-hierarchical MNL model, though it was not as good as corMNL.

5 Conclusions and Future Directions

In this paper, we have introduced a new approach for modelling hierarchical classes. Our experiments show that when the hierarchy actually does provide information regarding the similarity of classes, our approach outperforms both the simple MNL model and models based on decomposing the hierarchy into nested MNL models. Our method can be applied to many classification problems where there is prior knowledge regarding the structure of classes.

Our method can be applied to many classification problems where there is prior knowledge regarding the structure of classes. One such problem, which was our original motivation, is annotation of gene function. Using a prior based on the class hierarchy, we have been able to predict gene function with a higher accuracy (Shahbaba and Neal 2006). For this problem, we used a more elaborate prior for hyperparameters. We also introduced a new method for combining different data sources, which is a common issue in gene function classification.

More experiments are needed to compare corMNL with other approaches to utilizing the hierarchy, such as hSVM (Cai and Hoffmann 2004). One difference in our approach is that it is based on a probability model for hierarchical prior information, not on any particular hierarchy-based loss function. If a good probability model is used, the probability distributions obtained should provide the information needed to obtain good performance with any loss function.

So far, we have focused only on simple tree-like structures. There are other hierarchical structures that are more complex than a tree. For example, one of the most commonly used gene annotation schemes, known as Gene Ontology (GO), is implemented as a directed acyclic graph (DAG). In this structure a node can have more than one parent. Our method, as it is, cannot be applied to these problems, but it should be possible to extend the idea of summing coefficients along a path to the class in order to allow for multiple paths.

In our approach, we considered only one structure for each hierarchical classification problem. However, we might sometimes be able to think of more than one possible class hierarchy. It is possible to generalize our method to multiple hierarchies. As for the generalization to DAG's, it should be possible to sum coefficients along the multiple paths within different hierarchies. We can further use a set of hyperparameters to discover the relevance of each hierarchy. If we have prior knowledge that leads us to prefer some structures over others, we can incorporate that knowledge into the priors for these hyperparameters.

Finally, although the results presented in this paper are for linear models, we expect that a similar approach can be used in non-linear models such as neural networks.

References

Agresti, A. (2002) *Categorical Data Analysis*. John Willey and Son, Hoboken, New Jersy. 223

Cai, L. and Hoffmann, T. (2004) Hierarchical document categorization with support vector machines. *ACM 13th Conference on Information and Knowledge Management*. 224, 229, 233, 234, 235

Cesa-Bianchi, N., Gentile, C. and Zaniboni, L. (2006) Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, **7**, 31–54. 224

Dekel, O., Keshet, J. and Singer, Y. (2004) Large margin hierarchical classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*. 224

Dumais, S. T. and Chen, H. (2000) Hierachical classification of web content. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 256–263. 224

Fox, J. (1997) *Applied Regression Analysis, Linear Models and Related Methods*. Sage. 223, 224

Goodman, J. (2001) Classes for fast maximum entropy training. *Proceedings of the IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP), IEEE press*. 223

Koller, D. and Sahami, M. (1997) Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*. 223

Laven, K., Leishman, S. and Roweis, S. (2005) A statistical learning approach to document image analysis. *Conference on Document Analysis and Recognition (ICDAR), Seoul, South Korea*. 229, 231, 233

McCallum, A., Rosenfeld, R., Mitchell, T. and A., N. (1998) Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 359–360. 224

McFadden, D. (1980) Econometric models for probabilistic choice among products. *Journal of Business*, **53**, 13–36. 223

Mitchell, T. M. (1998) Conditions for the equivalence of hierarchical and flat Bayesian classifiers. URL `http://www.cs.cmu.edu/$\sim$tom/hierproof.ps`. 224

Neal, R. M. (1993) *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto. 231

Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. Springer Verlag, New York. 231

Neal, R. M. (2003) Slice sampling. *Annals of Statistics*, **31**, 705–767. 226

Riley, M. (1993) Functions of the gene products of Escherichia coli. *Microbiology Review*, **57**, 862–952. 221, 222

Sattath, S. and Tversky, A. (1977) Additive similarity trees. *Psychometrika*, **42**, 319–345. 223

Shahbaba, B. and Neal, R. M. (2006) Gene function classification using Bayesian models with hierarchy-based priors. *BMC Bioinformatics*, **7:448**. 221, 235

Tsochantaridis, I., Hoffmann, T., Joachims, T. and Altum, Y. (2004) Support vector machine learning for independent and structured output spaces. *Proceedings of the 21st International Conference on Machine Learning (ICML)*. 224

van Rijsbergen, C. J. (1972) *Automatic Information Structuring and Retrieval*. Ph.D. thesis, King's College, Cambridge. 229

Weigend, A. S., Wiener, E. D. and Pedersen, J. O. (1999) Exploiting hierarchy in text categorization. *Information Retrieval*, **1**, 193–216. 223

Acknowledgments

We thank Kevin Laven, Sam Roweis and Scott Leishman for providing the NIPS data. We also thank Lijuan Cai and Thomas Hofmann for providing the processed WIPO-alpha dataset. This research was supported by the Natural Sciences and Engineering Research Council of Canada. Radford Neal holds a Canada Research Chair in Statistics and Machine Learning.