Learning Not to Try Too Hard

Bill McDowell*
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
wmcdowel@cs.cmu.edu

Noah A. Smith
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
nasmith@cs.cmu.edu

1 Introduction

Discriminative learning algorithms are often motivated by their ability to trade off among different kinds of prediction mistakes with different costs. The cost of a mistake is usually taken to be fully defined by the task, i.e., human system designers are trusted to encode this knowledge prior to learning. Information about the inherent ease of avoiding some errors vs. others is generally not taken into account. Closely related to this, and critically important in domains where the data is constructed by humans, is the problem that the outputs in the training data may be unreliable. For example, if training data is produced by asking humans to label instances, and two labels are insufficiently well defined for human labelers to distinguish them, then a learner might be forgiven for conflating them.

We consider situations where human intuition about relative costs of different errors is insufficient. In a margin-based linear modeling framework, we propose a method for incorporating **learning of the cost function** alongside learning of the model. Our approach introduces explicit estimates of the "ease" of avoiding each type of error (for a particular model family). For error types that are "just too hard," our model is offered the possibility of giving up in favor of making other, less challenging predictions more accurately.

In practice, we found that our current method performs with approximately the same accuracy as a baseline SVM on text classification tasks, but it is able learn cost functions that intuitively make sense.

2 Background and Notation

In a prediction problem, let $\mathcal{X}$ denote the input space, $\mathcal{Y}$ denote the output space, and assume N training instances $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. We assume a linear model and prediction function:

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} \Big(f(x, y; \mathbf{w}) \triangleq \mathbf{w}^\top \mathbf{g}(x, y) \Big) \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^D$ are the parameters to be learned and $\mathbf{g} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^D$ is the feature vector function. We will let $\mathcal{M} = \{f(\cdot, \cdot; \mathbf{w}) \mid \mathbf{w} \in \mathbb{R}^D\}$ denote the model family under consideration, given a fixed choice of $\mathbf{g}$.

Our approach, which assumes $\mathcal{Y}$ is categorical, is based on the soft margin formulation of multiclass support vector machines [1–3]. Tsochantaridis et al. [4] and Taskar et al. [5] generalized this framework to allow for differences in costs between different kinds of mistakes, as found when $\mathcal{Y}$ is structured. Let the cost function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be such that $\Delta(y, y')$ is the cost of predicting y

*Alternative email: forkunited@gmail.com

when the correct label is y'. We use the "margin rescaling" variant of the multiclass SVM:

$$\min_{\boldsymbol{\xi}\geq 0,\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{1}{m}\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad \forall i, \forall y \in \mathcal{Y} \setminus \{y_i\}, f(x_i, y_i; \mathbf{w}) - f(x, y, \mathbf{w}) \geq \Delta(y, y_i) - \xi_i \tag{2}$$

This objective seeks $\mathbf{w}$ that minimizes misclassifications while maximizing the margin between correct and incorrect instances. Further, the more incorrect an (x, y) pair is, the greater the margin should be. This problem is often transformed into an unconstrained one corresponding to direct minimization of the regularized average hinge loss:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{N} -f(x_i, y_i; \mathbf{w}) + \max_{y\in\mathcal{Y}} f(x_i, y; \mathbf{w}) + \Delta(y, y_i) \tag{3}$$

We introduce some notation for errors. We let $\mathcal{S} \subseteq 2^{\mathcal{Y}\times\mathcal{Y}}$ be a collection of prediction error classes that exhausts $\mathcal{Y}^2$ (i.e., $\bigcup_{S\in\mathcal{S}} S = \mathcal{Y}^2$); the error classes need not be mutually exclusive. We let $e_S \in \mathbb{R}$ denote an estimate of the "ease" with which a learner searching in $\mathcal{M}$ can successfully avoid errors in class S. Then we let:

$$\Delta(y, y') = \sum_{S\in\mathcal{S}:(y,y')\in S} e_S = \mathbf{e}^\top \mathbf{s}(y, y') \tag{4}$$

where $\mathbf{e}$ is a vector of the e_S and $\mathbf{s}$ is a binary vector of length $\mathcal{S}$ indicating which error class(es) each possible confusion in $\mathcal{Y} \times \mathcal{Y}$ belongs to.

In this paper, we consider two prediction error classes, corresponding to unordered and ordered pairs of outputs. We denote them $\mathcal{S}^{\mathrm{u}}$ and $\mathcal{S}^{\mathrm{o}}$, respectively.

3 Cost Learning Model

Previous work assumes Δ follows intuitively from the prediction task. For example, in natural language dependency parsing, the number of words attached to the wrong parent (Hamming distance for the parse tree) is a sensible choice. We propose to parameterize Δ and learn its parameters jointly with $\mathbf{w}$. This learned cost function should encode distances between outputs from the perspective of the ease with which a model in the family $\mathcal{M}$ can distinguish between them. This joint learning setup is expected to be particularly useful when some classes of errors are difficult or impossible for a model in the class to resolve, due to unreliable annotations or an insufficient choice of features $\mathbf{g}$.

3.1 Ease

We desire a model that estimates prediction ease $\mathbf{e}$ while estimating predictive model parameters $\mathbf{w}$. We have used the term "ease" with respect to an arbitrary model in the family $\mathcal{M}$, but it is more sensible to consider the particular model we seek to estimate. We propose that, for error class S and a model with parameters $\mathbf{w}$, ease e_S should be inversely related to the number of of margin violations involving S that $f(\cdot, \cdot; \mathbf{w})$ makes in the training data assuming that argmax always gives a single label, breaking ties arbitrarily:

$$v_S(\mathbf{w}, \mathbf{e}) \triangleq \left|\left\{i \in \{1, \ldots, D\} \mid \left(y_i, \mathrm{argmax}_{y\in\mathcal{Y}} f(x_i, y; \mathbf{w}) + \mathbf{e}^\top \mathbf{s}(y, y_i)\right) \in S\right\}\right| \tag{5}$$

The intuition is that, when $v_S(\mathbf{w}, \mathbf{e})$ is large, it is because it is not easy for the model to shrink. Of course, we should also take into account that the distribution of the data may make some errors more frequent, inflating the size of the set in Eq. 5 even if S is "easy." Further, *infrequently* observed labels are generally expected to be harder to predict. Yet for an S that includes errors on a rarely occurring class, the set in Eq. 5 will necessarily be small, regardless of how easy it is. We therefore propose the following condition for e_S:

$$e_S = \max\left(0, 1 - \frac{v_S(\mathbf{w}, \mathbf{e})}{n_S}\right) \tag{6}$$

where n_S is a fixed, *a priori* upper bound on the count of S errors, v_S. This has the desirable property that if $v_S \geq n_S$, i.e., S is too difficult to shrink, then ease e_S goes to zero and the model is allowed to give up on S. It also keeps $e_S \in [0, 1]$, ensuring interpretability of the ease relative to the maximum possible value of $e_S = 1$.

3.2 Objective

Our approach is a modification to the SVM objective in Eq. 3; it is a joint optimization of $\mathbf{w}$ and $\mathbf{e}$:

$$\min_{\mathbf{e}\geq 0, \mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{1}{2}\|\mathbf{e}\|_{\mathbf{n}}^2 - \mathbf{e}^\top \mathbf{n} + \sum_{i=1}^{N} -f(x_i, y_i; \mathbf{w}) + \max_{y\in\mathcal{Y}} f(x_i, y; \mathbf{w}) + \mathbf{e}^\top \mathbf{s}(y, y_i) \tag{7}$$

where $\mathbf{n}$ is the vector of upper bounds on prediction error frequencies and $\|\mathbf{e}\|_{\mathbf{n}}^2 = \sum_{S\in\mathcal{S}} n_S e_S^2$.[1] The modification amounts to (1) including $\mathbf{e}$ as a free variable and (2) regularizing it with a quadratic penalty (second term in Eq. 7) and a linear "reward" (third term in Eq. 7). The linear penalty selects which e_S should be nonzero—equivalently, are not impossibly difficult. Setting $e_S = 0$ amounts to giving up on S errors.[2]

Most importantly, Eq. 7 is minimized at ease $\mathbf{e}$ that matches Eq. 6. To see this, we define $C(\mathbf{w}, \mathbf{e})$ as the objective minimized in Eq 7, and derive the optimal ease values $\mathbf{e}^*$ such that $\mathbf{0} \in \partial C(\mathbf{w}^*, \mathbf{e}^*)$. The subdifferential ∂C is given by:[3]

$$\begin{aligned}\partial C(\mathbf{w}, \mathbf{e}) =& \frac{\lambda}{2}\nabla(\|\mathbf{w}\|_2^2) + \frac{1}{2}\nabla(\|\mathbf{e}\|_{\mathbf{n}}^2) - \nabla(\mathbf{e}^\top \mathbf{n}) + \\ & \sum_{i=1}^{N} -\nabla(f(x_i, y_i; \mathbf{w})) + \mathbf{Co}(\nabla F^{\Delta}_{\max}(x_i, y_i; \mathbf{w}, \mathbf{e}))\end{aligned} \tag{8}$$

Where addition and subtraction operators are overloaded to perform additions and subtractions over sets, $\mathbf{Co}(\cdot)$ gives the convex hull, and $\nabla F^{\Delta}_{\max}$ is defined as:

$$\nabla F^{\Delta}_{\max}(x, y; \mathbf{w}, \mathbf{e}) = \{\nabla f(x, \hat{y}; \mathbf{w}) + \nabla \mathbf{e}^\top \mathbf{s}(y, \hat{y}) | \hat{y} \in \operatorname*{argmax}_{y'\in\mathcal{Y}} f(x, y'; \mathbf{w}) + \mathbf{e}^\top \mathbf{s}(y, y')\} \tag{9}$$

We are only interested in the value of $\mathbf{e}$, so we can consider Eq. 8 reduced to a simpler subdifferential with respect to a single component e_S of $\mathbf{e}$:

$$\partial_{e_S} C(\mathbf{w}, \mathbf{e}) = n_S e_S - n_S + \sum_{i=1}^{N} \mathbf{Co}\left(\frac{\partial F^{\Delta}_{\max}(x_i, y_i; \mathbf{w}, \mathbf{e})}{\partial e_S}\right) \tag{10}$$

Given that $\mathbf{0} \in \partial C(\mathbf{w}^*, \mathbf{e}^*)$, Eq 10 leads to:

$$e_S^* \in 1 - \frac{\sum_{i=1}^{N} \mathbf{Co}\left(\frac{\partial F^{\Delta}_{\max}(x_i, y_i; \mathbf{w}^*, \mathbf{e}^*)}{\partial e_S}\right)}{n_S} \tag{11}$$

If we allow the argmax in Eq. 9 to arbitrarily break ties as in Section 3.1, and we constrain $e_S^* \geq 0$ as in Eq. 7, then Eq. 11 suggests that e_S^* takes on the value given by Eq. 6 when the objective is minimized. So our objective chooses values of $\mathbf{e}$ according to our intuitions from Section 3.1.

[1] $\|\cdot\|_{\mathbf{n}}$ norm is a Mahalanobis norm, as described in [6].

[2] The meaning of "giving up" is unclear. It seems that setting $e_S = 0$ actually amounts to giving up on margin maximization, but not error minimization with respect to S, but we don't work this out in detail. **[NOAH, DO YOU UNDERSTAND THIS DIFFERENTLY? –BM]**

[3] See `http://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf` for help in understanding subgradients.

3.3 Constants n

The appropriate choice for the normalization vector $\mathbf{n}$ in Eq. 7 depends on the prediction classes in $\mathcal{S}$ and the types of bias we seek to avoid when estimating $\mathbf{e}$. For $\mathcal{S}^{\text{u}}$ and $\mathcal{S}^{\text{o}}$, we are most concerned with unbalanced marginal distributions over labels. Let c_y be the frequency of the label y in the training data, $|\{i \mid y_i = y\}|$. We propose two choices of $\mathbf{n}$, both based on the training data:

1. **Logical n**: an upper bound on $v_S(\cdot, \cdot)$ based on frequencies in the training data. For $\mathcal{S}^{\text{u}}$, let $n_{S_{\{y,y'\}}} = c_y + c_{y'}$. For $\mathcal{S}^{\text{o}}$, let $n_{S_{y,y'}} = c_y$ where $S_{y,y'}$ corresponds to an erroneous label of y' in place of the correct y.
2. **Expected n**: an upper bound calculated by assuming that our learner can perform better than a random classifier that uses label proportions observed in the training data. For $\mathcal{S}^{\text{u}}$, let $n_{S_{\{y,y'\}}} = 2c_y c_{y'}/N$. For $\mathcal{S}^{\text{o}}$, let $n_{S_{y,y'}} = c_y c_{y'}/N$.

The **Logical** choice will tend to dramatically overestimate the maximum count of each prediction error, but we might choose it over **Expected** if we believe that the random classifier would not give a baseline rate at which our model is biased to predict certain labels by the label distribution independent of the inputs. A third option, not explored here, might use a more sophisticated model to estimate bounds on error counts.

4 Experiments

We implemented the multiclass SVM (Eq. 3) and variations of our method—which we refer to as normalized cost learning (NCL; Eq. 7)— and we compared each of these implementations on standard text classification datasets in several experiments. In each experiment, we used a standard train/test split for the dataset, but with 10% of the training set held out to perform a grid search for λ over 16 values in $[10^{-6}, 50]$, choosing the value that gives the best accuracy, and then fixing λ and retraining on the whole training set. For training, ran stochastic gradient descent (SGD) with a learning rate determined by AdaGrad for 50 passes over random permutations of the data [7][8]. We observed that during the last ten passes, accuracy varied by < 0.01 and fewer than 10% of predictions changed.[4]

4.1 Datasets

We considered two datasets with relatively large output label sets: 20 Newsgroups (20NG; 20 category labels corresponding to newsgroups) [5] and Reuters-21578 (R52; 52 topic labels). [6] We followed [9] in preprocessing the text corpora (including downcasing, removing symbols, etc.), and we let features in $\mathbf{g}$ correspond to tf-idf computed over unigrams [10]. [7]

The 20NG dataset consists of 18,846 documents, sorted by date, with 60% used for training. Though the categories are roughly uniformly distributed, the topics vary greatly in their relatedness, following a hierarchical labeling scheme (e.g., *rec.autos* and *rec.motorcycles* are likely more closely related

[4] We were running for 150 iterations for previous buggy versions of the models, but we adjusted down to 50 iterations as our remaining working time began to dwindle, and we realized that these experiments were going to give negative results. It's unlikely that increasing the number of iterations from 50 will change the current conclusions, but if this work is ever published, we might want to rerun the algorithms with more iterations—just to be safe.

[5] `http://qwone.com/~jason/20Newsgroups`

[6] `http://www.csmining.org/index.php/r52-and-r8-of-reuters-21578.html`.

[7] We also ran experiments on synthetic data sets, but the results for those experiments are similar to the results for the text-classification data sets, and so they are not included in this document. Specifically, the synthetic data set results show at most minor performance gains on some versions of the data although the cost function seems to be learned appropriately. Furthermore, the synthetic data includes only four features in each task, which is unrealistic, and causes SGD difficulty in convergence due to the model's predictions being super sensitive to each weight. **[DOES THIS MAKE SENSE? –BM]** In the future, it would be beneficial to extend these datasets to include tasks with many more features, and use them to make sure our current cost learning models are functioning properly. See the code documentation at `https://github.com/forkunited/CostFunctionLearning/` for more detail on how to use and generate the synthetic data.

Table 1: Micro-averaged accuracies of different learners.

Learner	20NG		R52
	full	clusters	
SVM	0.834	0.920	0.945
NCL: $\mathcal{S}^{\text{o}}$, none	0.834	0.919	0.946
NCL: $\mathcal{S}^{\text{u}}$, none	0.832	0.921	0.945
NCL: $\mathcal{S}^{\text{o}}$, logical	0.836	0.920	0.948
NCL: $\mathcal{S}^{\text{u}}$, logical	0.830	0.919	0.948
NCL: $\mathcal{S}^{\text{o}}$, expected	0.830	0.920	0.949
NCL: $\mathcal{S}^{\text{u}}$, expected	0.820	0.911	0.946

than either to *sci.space*). This offers a way to measure the effectiveness of NCL at learning "ease": the less closely related two categories are, the greater the ease in learning to distinguish them.

The R52 dataset contains 9,100 documents; we use the ModApte split (70% training). The label distribution is skewed, with 43% of documents assigned to *earn* and 37 topics receiving fewer than 50 examples.

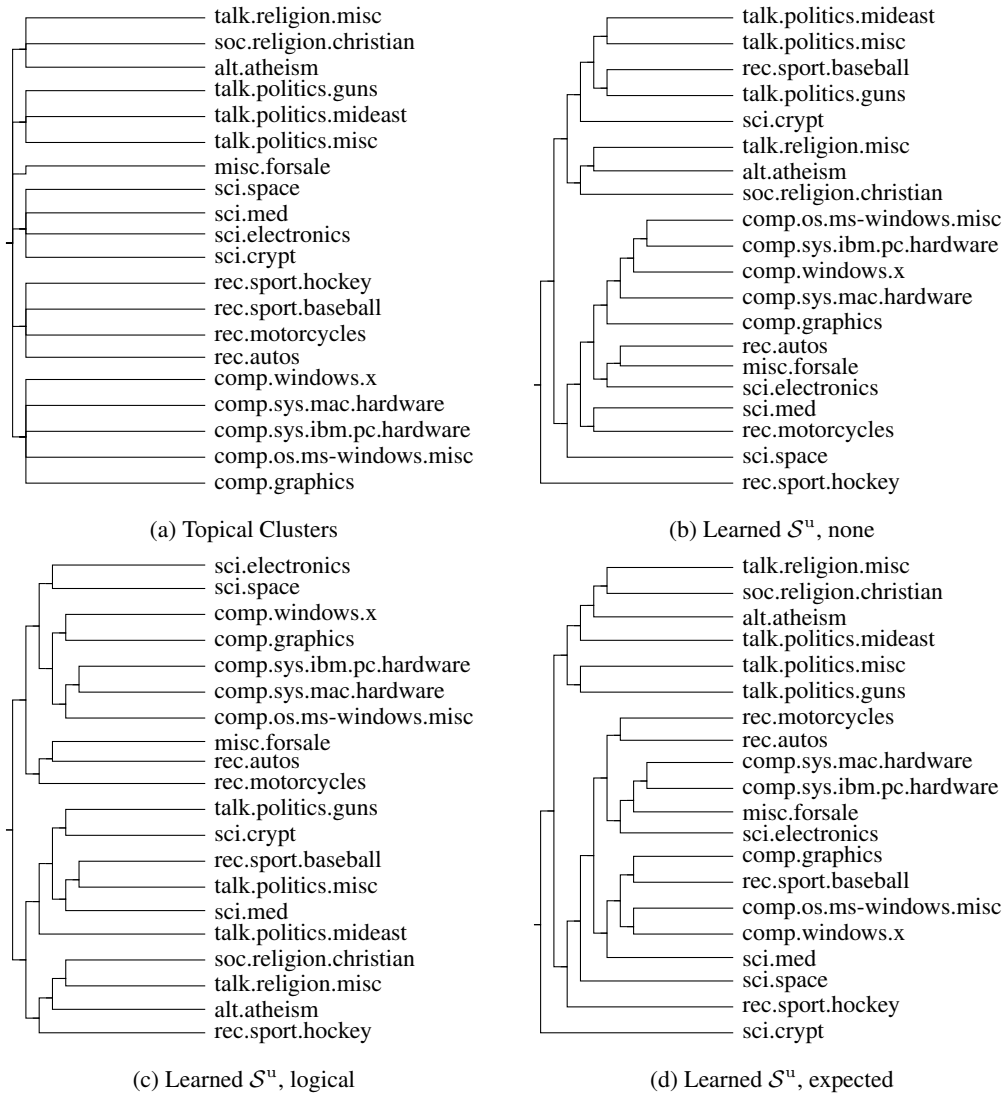
4.2 Results

Table 1 shows the micro-averaged accuracies on the Reuters and 20 newsgroups tasks for the SVM baseline model and NCL with various prediction error classes ($\mathcal{S}^{\text{u}}$; $\mathcal{S}^{\text{o}}$) and normalization constants ($\mathbf{1}$, i.e., none; logical; expected). We had expected that NCL might give an overall increase in accuracy on each task, but unfortunately, NCL generally gave approximately the same accuracy as the SVM, which suggests that the learned cost-scaled margins did not help NCL to generalize better than the SVM's constant margin.

We had especially hoped to see accuracy gains in the "clusters" column under "20NG". Whereas the "full" column shows accuracies computed over all categories in the 20 newsgroups task, the "clusters" column shows accuracies computed with all categories from a single topical cluster mapped to a single label—where the topical clusters are shown in Figure 1a and defined at `http://qwone.com/~jason/20Newsgroups/`. Assuming NCL's learned cost function gives higher costs for mislabelings between topical clusters and lower costs for mislabelings within topical clusters, we had expected that the model would focus on adjusting its feature weights to distinguish between categories in separate clusters, which would have resulted in an overall increase in accuracy in the "clusters" column.

We inspected the learned cost functions from the $\mathcal{S}^{\text{u}}$ versions of NCL to check our assumption that the costs reflect the relative ease of distinguishing between categories from separate topical clusters in Figure 1a. To observe the relationship between the learned costs and the topical clusters, we constructed average linkage hierarchical clusters (UPGMA) using the values of learned cost function weights $\mathbf{e}$ as distances [11]. The resulting cost learned hierarchical clusters shown in Figures 1b, 1c, and 1d seem qualitatively similar to to the topical clusters in Figure 1a, suggesting that our assumption that the learned costs reflect the topical clusters is correct—at least approximately. [8] However, even if each learned cost function's category hierarchy seems reasonable, it is difficult to see which normalization constants give the best hierarchy. Regardless, there is evidence that the normalization affects the learned $\mathbf{e}$ as expected from Eq 6 in that the "expected" normalization constants give components of $\mathbf{e}$ in $[0.759, 1]$, the "logical" constants give component of $\mathbf{e}$ in $[0.907, 1]$, and the "none" constants gives components of $\mathbf{e}$ in $[0.999, 1]$.

[8] It's difficult to make a precise comparison between the cost learned hierarchies and the topical clusters due to the fact that hierarchical clustering only outputs binary hierarchies, but the topical clusters do not form a binary hierarchy. It would be more appropriate to use a method that outputs non-binary clusterings given a distance metric, but we did not have time to explore such alternatives. Nevertheless, the hierarchies that we generated from learned costs still seem qualitatively compelling.



(a) Topical Clusters

(b) Learned $\mathcal{S}^{\mathrm{u}}$, none

(c) Learned $\mathcal{S}^{\mathrm{u}}$, logical

(d) Learned $\mathcal{S}^{\mathrm{u}}$, expected

Figure 1: Clusterings of 20 Newsgroup categories. 1a shows the topical clusters suggested at `http://qwone.com/~jason/20Newsgroups/`. 1b, 1c, and 1d show hierarchical clusterings constructed using distances from the "ease" $\mathbf{e}$ learned by NCL with $\mathcal{S}^{\mathrm{u}}$ prediction classes and "none" ($\mathbf{1}$), "logical", and "expected" normalization constants.

4.3 Discussion

Given that NCL's learned cost function approximates the relative ease of distinguishing between categories in separate topical clusters, it is surprising that NCL does not give an increased accuracy over the SVM baseline in the "clusters" column of Table 1. This could be an artifact of the 20 newsgroups data set—for example, the SVM's choice of weights might be near optimal for distinguishing between categories in separate topical clusters given its features, and so there might be nothing that NCL can do to improve these weights.

Alternatively, it's possible that the NCL adjusts its weights given the learned costs in a sub-optimal way. The baseline multiclass SVM from [2] generalizes the binary maximum margin-principle from [1] in a way that does not preserve the binary principle's original geometric interpretation.[9] Furthermore, [4] shows both "slack-rescaling" and "margin-rescaling" methods for incorporating the cost function into the SVM, and we've only experimented with the possibly inferior "margin-rescaling" method. Investigations of the generalization from binary to multi-class, the generalization from uniform to varying cost, and the interaction between these might suggest ways to use the cost learning to gain feature weights which generalize better.

A third alternative possible cause for the lack of an accuracy gain by NCL is that learned costs do not give an accurate enough encoding of the topical clusters, despite the fact that their derived hierarchies seem to roughly reflect those clusters. This failure might be corrected through a better choice of normalization constants $\mathbf{n}$.

5 Related Work

Here is a list of related work:[10]

1. Self-paced learning [12][11]
2. Curriculum learning [13]
3. Confidence weighted learning [14]
4. Inter-annotator agreement cost [15]
5. State splitting [16]
6. Finite state output encodings [17]

Also, see `http://www.cs.cmu.edu/~nasmith/papers/career-proposal-2010.pdf` for more motivations for the cost function learning.

5.1 Future Work

In future work, richer representations of prediction error types ($\mathcal{S}$) might be pursued. For example, types might be constructed based on frequencies of labels, with the rarest labels forming a group. For structured output spaces such as natural language parsing, the domain might suggest groups of errors; post hoc analysis of $\mathbf{e}$ might, in turn, suggest ways to improve the model through feature

[9] There may be an alternative multi-class SVM formulation which generalizes the maximum-margin principle in way that preserves the original geometric interpretation from the binary case. One way to start thinking about this is to consider that the difference between feature weights for two classes in the multi-class setting form a hyper-plane between those classes, and the margins around those planes can be maximized. However, there are $\binom{|\mathcal{Y}|}{2}$ such hyper-planes, but the current multi-class SVM only has $|\mathcal{Y}|$ sets of feature weights (one for each label), and so the adjustment of one of these planes and its margins has some effect on the others—not all choices of planes and margins are possible. A new generalized multi-class maximum margin principle might prescribe an optimal way to trade-off the margin sizes and errors between all of the planes. This new generalization might even involve some notion of learned cost, given that we want to more aggressively maximize the margins between classes which are more easily distinguishable. It's unclear at this point how any of this would work in detail, and we also haven't reviewed the literature prior to [2] that attempted other generalizations of the SVM to multi-class settings, so they it's possible that they tried something like this already.

[10] Add details if/when publishing.

[11] The objective function used in self-paced learning provided some inspiration for our objective function.

engineering.[12] Our framework is easily extended to let these classes depend on the input or metadata as well, allowing very rich parameterizations of learnable cost functions. Recall that these classes need not be mutually exclusive.

Alternative ways to estimate $\mathbf{n}$ might also be considered, such as using a more sophisticated model to estimate bounds on error frequencies in the training set. More generally, characterizations of ease might be developed through alternate means, such as the stability measure from learning theory [18], which might offer insight into the generalizability of predictions involving a particular label.

We concede that our notion of "ease" merges several concepts that might be treated separately. These include the reliability of the labels in training data, the distinctiveness of the labels given the model family (choice of features), the learnability of each label given the number of instances it has in the training set, and the overall similarity of the training distribution to the "true" one. We believe it is an open theoretical question how these various notions might relate to learning guarantees.

Finally, given our negative results on the text-classification data, it would be good to experiment with other datasets. It would be useful to experiment with synthetic data to try to find out what the model is doing, or also consider datasets with an even larger number of labels where there are even more places for the cost learning to make a difference.

References

[1] Vladimir N Vapnik. Statistical learning theory (adaptive and learning systems for signal processing, communications and control series), 1998.

[2] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[3] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[4] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[5] Ben Taskar Carlos Guestrin Daphne Koller. Max-margin markov networks. 2003.

[6] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[7] G George Yin and Harold Joseph Kushner. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

[8] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[9] Ana Cardoso-Cachopo. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[10] Man Lan, Chew Lim Tan, and Hwee-Boon Low. Proposing a new term weighting scheme for text categorization. In *AAAI*, volume 6, pages 763–768, 2006.

[11] P Legendre and L Legendre. Numerical ecology, 1998. *2nd English edition*, page 853.

[12] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, volume 1, page 3, 2010.

[13] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[14] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM, 2008.

[15] Barbara Plank, Dirk Hovy, and Anders Søgaard. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of EACL*, 2014.

[16] Slav Petrov. *Coarse-to-fine natural language processing*. Springer, 2011.

[17] Edward Loper. *Encoding structured output values*. PhD thesis, University of Pennsylvania, 2008.

[12] We note an interesting parallel to the *ceteris paribus* reasoning suggested by inspection of linear model weights $\mathbf{w}$; inspecting $\mathbf{e}$ shows, "all other things equal," a scaling of error types by ease-of-avoidance.

[18] Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3):161–193, 2006.