Learning Not to Try Too Hard

Anonymous Author(s)
Affiliation
Address
email

[IT SEEMS LIKE A LOT OF OTHER PAPERS REFER TO 'COST' AS 'LOSS'... IS THERE SOMEWHERE WHERE PEOPLE CALL IT 'COST' LIKE WE ARE? –BM] [WILL ADDRESS THIS –NAS]

[NOAH, I PUT YOUR NAME SECOND BECAUSE YOUR NAME SEEMS TO BE LAST ON ALL OF YOUR RECENT PAPERS. I DON'T CARE WHICH OUR NAMES IS FIRST THOUGH, SO FEEL FREE TO SWAP IF YOU WANT. –BM] [I USUALLY GO LAST –NAS]

Abstract

[DO LAST –NAS]

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 two labels are ill-defined by the data-generating process, then a learner can 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

[CITE EXAMPLE OF UNRELIABLE OUTPUT LABELS –BM]

[MIGHT WANT TO GIVE SOME MOTIVATING EXAMPLES? NOT SURE IF THAT'S NECESSARY OR NOT. ONE EXAMPLE OF MULTICLASS DOMAIN, AND ONE EXAMPLE WHERE COST FUNCTIONS ARE USED IN STRUCTURED DOMAINS –BM]

[MENTION EXAMPLES OF MEASURES OF 'DIFFICULTY'? CITE. –BM]

Our experiments show benefits on standard benchmarks [CHANGE IF ONLY ONE –NAS] in text classification.

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 multi-class support vector machines [1–3]. Tsochantaridis et al. [4] and Taskar et al. **??** 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 when the correct label is y'. We use the "margin rescaling" variant the multiclass SVM:

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

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.

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 $\mathbf{g}$.

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 any model in the linear family (given by $\mathbf{g}$) can learn to avoid errors in class S. Then we let:

$$\Delta(y, y') = \sum_{S \in \mathcal{S} : (y, y') \in S} e_S = \mathbf{e}^\top \mathbf{s}(y, y') \tag{3}$$

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 belongs to.

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

3 Cost Learning Model

We desire a model that estimates prediction ease $\mathbf{e}$ while estimating predictive model parameters $\mathbf{w}$. Above, we defined "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 proportional to the rate of margin violations involving S that $f(\cdot, \cdot; \mathbf{w})$ makes in the training data:

$$\left\{ i \in \{1, \ldots, D\} \mid \left(y_i, \text{argmax}_{y \in \mathcal{Y}}\, f(x_i, y; \mathbf{w}) + \mathbf{e}^\top \mathbf{s}(y, y_i)\right) \in S \right\} \tag{4}$$

[LEFT OFF HERE –NAS]

The size of $S_{\mathcal{M}, \mathbf{g}, D}$ is the number of training examples with margin violations in class S. If the size of this set is large, we might infer that the model has trouble shrinking it, and so it's not 'easy'. This might lead us to conclude that $S_{\mathcal{M}, \mathbf{g}, D}$ tends to shrink with 'easiness'. However, for many data sets and choices of $\mathcal{S}$, the size of each $S_{\mathcal{M}, \mathbf{g}, D}$ can be inherently biased by the data independently of 'easiness'. For example, for $S_{\{\mathbf{y}, \mathbf{y}'\}} \in \mathcal{S}_{[\mathcal{Y}]^2}$, the size of $S_{\{\mathbf{y}, \mathbf{y}'\}, \mathcal{M}, \mathbf{g}, D}$ is biased by the number of examples in the training data which have output labels $\mathbf{y}$ and $\mathbf{y}'$–if there are few training examples of labels $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, then the size of $S_{\{\mathbf{y}, \mathbf{y}'\}, \mathcal{M}, \mathbf{g}, D}$ will necessarily be small relative to other prediction classes. Furthermore, we expect output labels which occur infrequently in the training data to be more difficult for the model to predict correctly, so this will lead to the size of $S_{\{\mathbf{y}, \mathbf{y}'\}, \mathcal{M}, \mathbf{g}, D}$

increasing with the 'easiness' of $S_{\{\mathbf{y},\mathbf{y}'\}}$ which is opposite the conclusion that that we draw if we think of the size of $S_{\{\mathbf{y},\mathbf{y}'\},\mathcal{M},\mathbf{g},D}$ as increasing due to the model's difficulty in shrinking it. In general, this suggests that if we want the size of $S_{\mathcal{M},\mathbf{g},D}$ to vary with easiness, we need to normalize it to account for properties of the training data that introduce irrelevant biases.

3.1 A measure of 'easiness'

[MIGHT WANT TO DISCUSS OTHER POSSIBILITIES, OR GENERALLY REFER TO OTHER POSSIBILITIES IN FUTURE WORK –BM]

The above observations suggest the following as a possible measure of 'easiness':

$$\mathcal{E}(S,\mathcal{M},\mathbf{g},D) = \max\left(0, 1 - \frac{|S_{\mathcal{M},\mathbf{g},D}|}{n_{S,\mathcal{M},\mathbf{g},D}}\right) \tag{5}$$

Where $n_{S,\mathcal{M},\mathbf{g},D}$ is a normalization constant which gives the maximum possible value we expect for the size of $S_{\mathcal{M},\mathbf{g},D}$, accounting for irrelevant biases introduced by the data as discussed above. This measure of easiness is in $[0,1]$, and it has the property that if $|S_{\mathcal{M},\mathbf{g},D}| \geq n_{S,\mathcal{M},\mathbf{g},D}$, then $\mathcal{E}(S,\mathcal{M},\mathbf{g},D) = 0$, indicating that $S_{\mathcal{M},\mathbf{g},D}$ is so difficult to shrink that its size is greater than our expected upper bound.

3.2 A cost learning objective

[CITE SELF-PACED LEARNING FOR INSPIRATION FOR NEW OBJECTIVE FUNCTION? –BM]

[ADD FOOTNOTE ABOUT RELATIONSHIP BETWEEN NORM IN OBJECTIVE AND MAHALANOBIS NORM –BM]

[IS THERE ANY MATH I SHOULD BE MORE EXPLICIT ABOUT? –BM]

We can modify the margin re-scaling SVM learning procedure given by Quadratic Program 2 to learn the cost function according to the easiness measure shown in Equation 5. First, we transform the quadratic program into the equivalent unconstrained optimization problem:

$$\min_{\mathbf{w}} \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m}\left(-F(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}) + \max_{\mathbf{y}\in\mathcal{Y}}\left(F(\mathbf{x}_i,\mathbf{y};\mathbf{w}) + \Delta(\mathbf{y}_i,\mathbf{y})\right)\right) \tag{6}$$

And we modify this function to include the cost learning as:

$$\min_{\hat{\mathcal{E}}\geq 0,\mathbf{w}} \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m}\left(-F(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}) + \max_{\mathbf{y}\in\mathcal{Y}}\left(F(\mathbf{x}_i,\mathbf{y};\mathbf{w}) + \hat{\mathcal{E}}^\top\mathcal{S}(\mathbf{y}_i,\mathbf{y})\right)\right) - \hat{\mathcal{E}}^\top\mathbf{n} + \frac{1}{2}\|\hat{\mathcal{E}}\|_{\mathbf{n}}^2 \tag{7}$$

Where $\mathbf{n}$ is the vector of normalization constants, and $\|\hat{\mathcal{E}}\|_n^2 = \sum_{S\in\mathcal{S}} n_S \hat{\mathcal{E}}_S^2$.[1] The idea for this new objective function is to use the $-\hat{\mathcal{E}}^\top\mathbf{n}$ term to select which $\hat{\mathcal{E}}_S$ should be non-zero (or correspondingly which S are not impossibly difficult), and use the $\|\hat{\mathcal{E}}\|_{\mathbf{n}}^2$ to limit the magnitude of $\hat{\mathcal{E}}_S$.

If the solution to Objective 7 is at point that is differentiable with respect to $\hat{\mathcal{E}}_S$, then it's easy to show that $\hat{\mathcal{E}}_S$ has exactly the value given by Equation 5. Otherwise, if the solution is at a non-differentiable point, then $\hat{\mathcal{E}}_S$ has a value that approximates Equation 5 in a sensible way. Specifically, the non-differentiable points of Objective 7 occur due to ties between multiple sequences of labels for the minimum value solution, and each of these equally good label sequences has differently sized incorrect prediction classes, giving different values of 'easiness' for each class according to Equation 5. The values of $\hat{\mathcal{E}}_S$ at the non-differentiable points are between the values given by Equation 5 for each tied best label sequence.

[1] We abbreviate $\hat{\mathcal{E}}(S,\mathcal{M},\mathbf{g},D)$ and $n_{S,\mathcal{M},\mathbf{g},D}$ as $\hat{\mathcal{E}}_S$ and n_S for readability.

[IS THIS NON-DIFFERENTIABLE PART UNDERSTANDABLE? IS IT ENOUGH, OR DO I NEED TO GIVE THE PROOF? I STILL HAVEN'T ACTUALLY GONE THROUGH THE PROOF COMPLETELY, SO MAYBE WE SHOULD AT LEAST GO OVER THE REASONING TO MAKE SURE I'M NOT CRAZY. –BM]

3.3 Some 'easiness' normalization constants

[MENTION CHOOSING N USING BASELINE SVM ESPECIALLY IF THIS IS INCLUDED IN RESULTS. BUT OTHERWISE MAYBE PUT IT IN FUTURE WORK SINCE CHOOSING N BASED ON OTHER MODELS IS A GENERAL TOPIC TO EXPLORE –BM]

The appropriate choice for the normalization vector $\mathbf{n}$ in Objective 7 depends on the prediction classes $\mathcal{S}$ and the type of irrelevant bias we are trying to remove from 'easiness' measure. For $\mathcal{S}_{[\mathcal{Y}]^2}$ and $\mathcal{S}_{\mathcal{Y}^2}$, we want to remove the bias introduced into the size of the prediction classes by non-uniform label distributions, as discussed at the beginning of Section 3. Otherwise, the model will tend to over-estimate the costs of incorrect predictions involving labels that occur infrequently in the training data. The following are two plausible choices of $\mathbf{n}$ to achieve this goal. In each, assume that $D_{\mathbf{y}}$ is the set of training examples for which $\mathbf{y}$ is the correct label.

1. **Logical** $\mathbf{n}$: Choose n_S as an upper bound on $|S|$ that cannot possibly be violated given the training data. For prediction classes $S_{\mathbf{y},\mathbf{y}'} \in \mathcal{S}_{\mathcal{Y}^2}$, choose $n_{S_{\mathbf{y},\mathbf{y}'}} = |D_{\mathbf{y}}|$, and for prediction classes $S_{\{\mathbf{y},\mathbf{y}'\}} \in \mathcal{S}_{[\mathcal{Y}]^2}$, choose $n_{S_{\{\mathbf{y},\mathbf{y}'\}}} = \max\left(|D_{\mathbf{y}}|, |D_{\mathbf{y}'}|\right)$.
2. **Expected** $\mathbf{n}$: Choose n_S as the expected number of times a model makes a mistake in S if it predicts labels at random according to their distribution in the training data. For prediction classes $S_{\mathbf{y},\mathbf{y}'} \in \mathcal{S}_{\mathcal{Y}^2}$, choose $n_{S_{\mathbf{y},\mathbf{y}'}} = \frac{|D_{\mathbf{y}}||D_{\mathbf{y}'}|}{m}$, and for prediction classes $S_{\{\mathbf{y},\mathbf{y}'\}} \in \mathcal{S}_{[\mathcal{Y}]^2}$, choose $n_{S_{\{\mathbf{y},\mathbf{y}'\}}} = \frac{2|D_{\mathbf{y}}||D_{\mathbf{y}'}|}{m}$. Other variations of this idea might choose alternative models by which to compute the expectation, but model that makes predictions at according to the training data label distribution seems reasonable for getting rid of the label distribution bias.

In general, the **Logical** choice of $\mathbf{n}$ is an upper bound on the **Expected** choice. The **Logical** choice will tend to over-estimate the maximum size of each prediction class, but we might choose it over **Expected** if we have reason to believe that it is difficult to estimate the baseline rate at which the model is biased to predict certain labels by the label distribution independent of the inputs.

[MORE DETAIL ON WHY EXPECTED MIGHT BE A BETTER CHOICE THAN LOGICAL –BM]

4 Experiments

[ADD GRAPHS SHOWING CONVERGENCE OF SGD? –BM]

We implemented the multiclass SVM and normalized cost learning SVM (SVMCLN) using stochastic (sub-)gradient descent (SGD) to approximate Objectives 6 and 7 (see [6] on SGD). Our SGD implementation used a learning rate of $\frac{1}{\lambda_2 t}$ at time-step t following the Pegasos algorithm in [7]. We ran several experiments to compare these models on standard text classification corpora. In each experiment, we compared the accuracies of the models on standard train/test split given by each respective data corpus while also randomly selecting a 10% subset of the training data to use as a dev set. We used the dev set to perform a grid search over 16 possible values ranging from 10^{-6} to 0.5×10^2 for the λ_2 hyper-parameter in each model. After the grid search, we fixed the best value for λ_2, and retrained on the full training data, evaluating the final performance of the model on the test set.

Due to the stochasticity of the optimization algorithm, we expected some noisiness in its convergence. In general, we ran SGD for 150 iterations over the random permutations of the full data. After the 150 iterations, the accuracy and predictions of each model on the dev/test sets were relatively stable; the accuracy varied by at most 0.001 and fewer 5% of the predictions changed during the last 10 iterations over the training data.

4.1 Datasets

[CITE HTTP://WWW.AAAI.ORG/PAPERS/AAAI/2006/AAAI06-121.PDF ON RELATIVELY LOW DIFFERENCE IN PERFORMANCE BETWEEN LOG(TF) AND TFIDF –BM]

[CITE HTTP://QWONE.COM/~JASON/WRITING/LOOCV.PDF FOR EXAMPLE USE OF LOG(TF) –BM]

[PREPROCESSED USING METHOD FROM HTTP://WEB.IST.UTL.PT/ACARDOSO/DOCS/2007-PHD-THESIS.PDF –BM]

We compared the performance of the models on the 20 Newsgroups and the Reuters-21578 (R52) data sets. We chose these two data sets because they both have a relatively large number of output labels, some of which might not be distinctive enough for the SVM to plausibly learn well given its features, in which case we might expect a gain in accuracy from SVMCLN. As shown below, we found out that this expectation was correct for the 20 Newsgroups data, but incorrect for the Reuters data, for reasons that became apparent to us after we reviewed the results.

For both of these data sets, the target labels $\mathcal{Y}$ are single document topics/categories, and the inputs $\mathcal{X}$ are documents. We preprocessed each text document from both corpora using the procedure given in [FILL IN] (convert to lower-case, remove symbols, etc), and then we computed $\mathbf{g}$ as normalized log unigram term-frequency features. In particular:

$$\mathbf{g}_{\mathbf{y}'}(\mathbf{x}, \mathbf{y}) = \mathbb{1}(\mathbf{y} = \mathbf{y}')\mathbf{f}(\mathbf{x}) \tag{8}$$

Where $\mathbf{f}(\mathbf{x})$ is a normalized vector whose elements are $\log(1 + tf(\mathbf{x}, v))$ when v is a unigram in the corpus vocabulary and $tf(\mathbf{x}, v)$ is the frequency of v in document $\mathbf{x}$.

4.1.1 Reuters 21578 (R52)

The Reuters-21578 R52 corpus contains 9100 documents from the original Reuters-21578 data, each of which is labeled with one of 52 possible topics. [2] The documents are divided along the Reuters-21578 'ModApte' split into $\tilde{7}0\%$ training and $\tilde{3}0\%$ test. The label distribution is extremely non-uniform, with 43% of the documents assigned to a single topic, and 71% of the other topics each assigned fewer than 50 (0.5%) documents.

4.1.2 20 newsgroups

The 'by date' version of the 20 Newsgroups data contains 18846 documents sorted by date into 60% for training and 40% for testing. [3] In comparison to the Reuters data, the newsgroups distribution of 20 categories is nearly uniform, with some topics highly related and some topics entirely unrelated. The relatedness of the topics is approximately captured by their hierarchical naming scheme (e.g. there is *rec.autos* and *rec.motorcycles* which are likely to be highly related to each other, but mostly unrelated to *sci.space*). We expected that this hierarchy would be encoded by the learned 'easiness' approximations, since the 'easiness' with which the model discriminates two categories likely decreases with their relatedness.

4.2 Results

[ALL 20NEWS TABLE NUMBERS ARE WRONG. NEED TO FIX THEM AFTER EXPERIMENTS RERUN. –BM]

Table 1 shows the micro-averaged accuracies on the Reuters and 20 newsgroups tasks for the SVM baseline model and versions of SVMCLN with different choices of normalization constants $\mathbf{n}$ and incorrect prediction classes $\mathcal{S}$. The last column in the table shows that none of the SVMCLN variations improve the ccuracy over the SVM on the Reuters task, but inspection of the confusion matrix for the SVM baseline provides some intuition for this lack of improvement. The matrix shows that 53% of the SVM's mistakes were on examples with whose topics had 10 or fewer test examples, all

[2] See `http://www.csmining.org/index.php/r52-and-r8-of-reuters-21578.html`.
[3] See `http://qwone.com/~jason/20Newsgroups/`.

Table 1: Micro-averaged Accuracies

Model	n	$\mathcal{S}$	20news	20news level 2	20news level 1	Reuters
SVM	N/A	N/A	0.7760	0.8008	0.8654	0.9213
SVMCLN	None	$\mathcal{Y}^2$	0.8008	0.8259	0.8752	0.9213
SVMCLN	None	$[\mathcal{Y}]^2$	0.8011	0.8257	0.8751	0.9194
SVMCLN	Logical	$\mathcal{Y}^2$	0.8024	0.8271	0.8769	0.9210
SVMCLN	Logical	$[\mathcal{Y}]^2$	0.8376	0.8631	0.9142	0.9159
SVMCLN	Expected	$\mathcal{Y}^2$	0.8303	0.8557	0.9092	0.9159
SVMCLN	Expected	$[\mathcal{Y}]^2$	0.8307	0.8558	0.9084	0.9171

off which were predicted incorrectly. Furthermore, there is no non-diagonal element in the confusion matrix with more than 10 mistakes. These observations suggest that many of the SVM baseline's mistakes come from infrequent labels rather than systematic conflations between certain label pairs, and further that all of the incorrect prediction classes in the cost learning model will be extremely small to begin with. As a result, it's unlikely that SVMCLN would be able to re-scale the costs to greatly shrink any incorrect prediction classes. In hindsight, this provides a good example of a good example of how analysis of the errors made by the standard SVM can determine whether cost learning will be beneficial.

In contrast, the fourth column of Table 1 shows that every SVMCLN variation improves the accuracy over the SVM on the 20 Newsgroups data. Unsurprisingly, the **Logical** and **Expected** normalized versions perform better than the non-normalized versions since they should give better estimates of the 'easiness' measure and cost function. Also, each $\mathcal{S}_{[\mathcal{Y}]^2}$ version performs slightly better than each $\mathcal{S}_{[\mathcal{Y}]^2}$ version, which makes sense given that the ease of resolving mistakes in $S_{\mathbf{y},\mathbf{y}'}$ should be the same as the ease of resolving mistakes in $S_{\mathbf{y}',\mathbf{y}}$, and so the ordering on $\mathbf{y}$ and $\mathbf{y}'$ gives the model unnecessary extra parameters. We also expected **Expected** to perform better than **Logical** since **Logical** over-estimates the maximum value of each incorrect prediction class, but this expectation was not met with $\mathcal{S}_{[\mathcal{Y}]^2}$ prediction classes, possibly because the **Expected** normalizers tend to under-estimate.

The hierarchical structure of the 20 Newsgroups topics encodes a notion of distance between topics as distance within the hierarchy. The fifth and sixth columns of Table 1 show the accuracies computed when nearby topics in the hierarchy are are collapsed into single topics–the fifth column shows the accuracies computed when all topics that are the same to two levels deep in the hierarchy are collapsed into a single topic, and the sixth column shows the accuracies computed when all topics that are the same at the first level of the hierarchy are collapsed into a single topic.

[I THINK THERE ARE WAYS TO MAKE THE FOLLOWING RESULT CLEARER IF WE COMPUTE OTHER NUMBERS... BUT I DON'T KNOW IF WE HAVE TIME FOR THAT. THIS MAY BE A BIT CONFUSING I THINK THOUGH –BM]

The cost learning should learn a notion of distance between topics approximated by the cost function, and we expect this notion of distance to approximate the notion of distance encoded by the hierarchy. The approximated notion of distance encoded in the cost function should cause SVMCLN to improve at distinguishing topics that it estimates to be far apart from each other. So if SVMCLN's learned distances approximate the distances through the hierarchy, then we should expect the SVMCLN's collapsed accuracy improvements to be nearly as great as the uncollapsed accuracy improvements. This is shown by the [FILL] accuracy improvements in the fourth, fifth, and sixth columns of Table 1 by the **Logical** $\mathcal{S}_{[\mathcal{Y}]^2}$ version of SVMCLN.

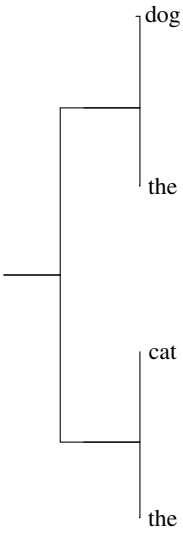
[FIX THE ABOVE PARAGRAPH WHEN GET NEW NUMBERS –BM]

[CITE HIERARCHICAL CLUSTERING. BE SPECIFIC ABOUT WHICH TYPE OF HIERARCHICAL CLUSTERING IS USED. –BM]

[ADD HIERARCHY FIGURE. –BM]

We also directly evaluated the extent to which the cost function approximates the newsgroup hierarchy by constructing a hierarchy from the 'easiness' approximations and comparing it to the newsgroup hierarchy. In order to construct the hierarchy, we ran hierarchical clustering on the newsgroups where each approximated 'easiness' value acts as a distance measure. The approximate hierarchy for the **Logical** $\mathcal{S}_{[\mathcal{Y}]^2}$ SVMCLN is shown in figure [FILL], and the true Newsgroup hierarchy is shown in figure [FILL].

[ADD MEASUREMENTS OF SIMILARITY BETWEEN HIERARCHIES –BM]



5 Discussion

[MAYBE ADD SUMMARY OF CONCLUSIONS HERE OR AT THE VERY END...? –BM]

5.1 Related literature

[HERE ARE SOME THINGS TO POSSIBLY WRITE ABOUT: –BM]

Self-paced learning [8].

Curriculum Learning [9].

Confidence weighted learning [10].

Ed Hovy inter-annotator agreement cost [11]

Hidden variable learning by state splitting mentioned in `http://www.cs.cmu.edu/~nasmith/papers/career-proposal-2010.pdf` [12].

Finite state output encodings mentioned in `http://www.cs.cmu.edu/~nasmith/papers/career-proposal-2010.pdf` [13]

5.2 Future work

[IS THERE ANYTHING ELSE THAT I FORGOT? –BM]

We made several choices within the present work for which many alternatives might be interesting to pursue in future research. We proposed two choices for sets $\mathcal{S}$ of incorrect prediction classes which bucket predictions based on label pairs, but many others are possible. For example, we might consider construct prediction classes for varying frequencies of the labels in the training data, motivated by the idea that incorrect predictions involving frequently occurring labels are easier to resolve than incorrect predictions involving infrequently occurring labels.

Another idea is to change the cost function to incorporate the input $\mathbf{x}_i$ as $\Delta(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y})$, This would allow the incorrect prediction classes and their 'easiness' estimates to depend on features of the input–which would possibly be beneficial to the model for learning and to the model engineer for error analysis.

We chose to explore the straightforward **Logical** and **Expected** values for $\mathbf{n}$, but we might also consider using a more sophisticated model to estimate the irrelevant biases in the sizes of the incorrect prediction classes.

Our cost learning model was build for multiclass classification tasks, but it should be easy to extend to structured prediction tasks.

[NOAH, WOULD IT BE GOOD TO TALK ABOUT SOME OF THE LEARNING OUTPUT SPACES IDEAS FROM YOUR CAREER PROPOSAL IN RELATION TO THE STRUCTURED POSSIBILITY HERE? –BM]

In the more general theme of cost function learning, it's possible to choose alternative measures of 'easiness' that do not directly depend on the sizes of the prediction classes and could require changing the model to an entirely different form. For example, we might make easiness depend on some measure of linear separability between each label pair, or generalizability of predictions involving certain labels. The generalizability of certain predictions might be characterized by some measure like 'stability' from learning theory [14].

[THE COMMENT ABOUT STABILITY MIGHT BE ENTIRELY OFF BASE BECAUSE I DON'T KNOW ANYTHING ABOUT IT, BUT IT WAS SOMETHING I WANTED TO LOOK INTO. –BM]

There are also several unanswered theoretical questions related to this work. We have used scare-quotes around 'easiness' throughout this paper because we aren't using the term in an unambiguous way, and we don't have a single precise concept in mind. There are several concepts of 'easiness' related to, e.g., the reliability of the annotated labels, the distinctiveness of the labels given the choice of features, the distinctiveness of the labels given the choice of model, the learnability of each label given its frequency, or the similarity between the distribution of the training data and the true distribution. There are several ways we might model these notions of 'easiness' by the choice of model and the categorization of incorrect predictions into classes. It would be interesting to describe a taxonomy of different kinds of 'easiness' and ways in which they might be learned. Furthermore, it would be useful to find a systematic way to predict whether learning a given notion of 'easiness' on a data set will allow a model to improve its accuracy. Analyzing the confusion matrix for the Reuter's data helped us make some guesses about why the notion of 'easiness' captured by our model did not help improve its performance, but there might be ways of making our intuitions about this more systematic.

[POSSIBLY MOVE PARTS OF THE PREVIOUS PARAGRAPH TO THE BACKGROUND/INTRODUCTION SECTIONS. ALSO MIGHT WANT TO REWORD IT TO MAKE CLEARER, OR JUST TALK ABOUT IT IN A DIFFERENT WAY. AT LEAST MAKE IT SHORTER... –BM]

[THE 'REFERENCES' HEADING GIVEN BY THE BIBLIOGRAPHY COMMAND IS THE WRONG SIZE FONT. NEEDS TO BE THE SIZE OF A 'THIRD LEVEL HEADING'. HOW TO CHANGE THIS? –BM]

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

[7] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

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

[9] 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.

[10] 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.

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

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

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

[14] 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.