Learning when to try hard at learning

Anonymous Author(s)
Affiliation
Address
email

[PROBABLY NEED A BETTER TITLE. –BM]

[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]

Abstract

1 Introduction

[GENERAL IDEAS ABOUT COST LEARNING ABSTRACTED AWAY FROM THE SVM DETAILS IN THE NEXT SECTION...? –BM]

2 Background

[I INTRODUCED SVMS THROUGH THE QUADRATIC PROGRAMMING FORMULATION BECAUSE IT SEEMED EASIER TO SUMMARIZE MARGIN MAXIMIZATION ACROSS ALL THE RELEVANT SVM VARIANTS THAT WAY WHILE STAYING TRUE TO THE LITERATURE... BUT MAYBE I SHOULD HAVE JUST KEPT IT IN THE FORM OF AN UNCONSTRAINED OPTIMIZATION PROBLEM FOR CONTINUITY WITH THE REST OF THIS PAPER? –BM]

[CITE SOME PAPER ON USING BINARY SVMS FOR MULTICLASS –BM]

[CITE MULTICLASS SVM –BM]

[CITE SOMETHING FOR MARGIN MAXIMIZATION –BM]

[CITE STRUCTURED SVM –BM]

[CITE EXAMPLE OF UNRELIABLE OUTPUT LABELS –BM]

[ALSO CITE TASKAR ET AL FOR 'MARGIN RESCALING'? –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]

For the multiclass classification problem, we are given a set of m labelled training data instances $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\} \subset \mathcal{X} \times \mathcal{Y}$ where each example $\mathbf{x}_i$ is assigned label $\mathbf{y}_i$, and we want to learn a function $f : \mathcal{X} \to \mathcal{Y}$ which gives the correct label for any input taken from $\mathcal{X}$. Past work has developed a multiclass support vector machine (SVM) which generalizes the concept of margin maximization employed by classical SVMs for binary classification tasks. In particular, assuming

that $\mathbf{g} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^k$ computes a featurized representation of an input-output pair, the multiclass SVM gives a score to a labelled instance $(\mathbf{x}, \mathbf{y})$ as:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{g}(\mathbf{x}, \mathbf{y}) \tag{1}$$

And makes predictions according to:

$$\hat{f}(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y}' \in \mathcal{Y}}{\operatorname{argmax}} F(\mathbf{x}, \mathbf{y}'; \mathbf{w}) \tag{2}$$

Where the feature weights $\mathbf{w}$ are given by the solution to the following soft-margin maximizing quadratic program:

$$\begin{aligned} &\min_{\xi \geq 0, \mathbf{w}} \left(\frac{\lambda_2}{2} \|\mathbf{w}\|_2^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \\ &\text{s.t. } \forall i : \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i : F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}) \geq 1 - \xi_i \end{aligned} \tag{3}$$

This quadratic program chooses weights that minimize the number of misclassified instances while simultaneously trying to increase the margin between the scores of correct and incorrect labels, and the λ_2 hyper-parameter determines the relative importance of these two goals.

Tsochantaridis et al further generalized the margin maximization to make sense for domains where some prediction mistakes are more costly than others (especially domains where $\mathcal{Y}$ contains a large number of structured outputs). They introduce a function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ that determines the cost $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ of predicting label $\hat{\mathbf{y}}$ when then correct label is $\mathbf{y}$. The $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ is larger for more inaccurate $\hat{\mathbf{y}}$ predictions, and $\Delta(\mathbf{y}, \mathbf{y}) = 0$, giving no cost for a correct prediction. Tsiochantaridis et al incorporate the cost function into the SVM learning by modifying the multiclass SVM quadratic program through 'margin re-scaling' as:[1]

$$\begin{aligned} &\min_{\xi \geq 0, \mathbf{w}} \left(\frac{\lambda_2}{2} \|\mathbf{w}\|_2^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \\ &\text{s.t. } \forall i : \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i : F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}) \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i \end{aligned} \tag{4}$$

Intuitively, this margin re-scaling quadratic program allows the model to choose margins between between scores of $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ proportional to the prediction cost $\Delta(\mathbf{y}_i, \hat{\mathbf{y}})$. This choice biases the model toward making low cost predictions over high cost predictions.

Whereas previous work assumes that we hand-select domain-specific cost functions that seem like good measures of the distances between labels, we propose learning this function from the data jointly with the prediction function f. Similarly to the hand-selected cost functions, the learned cost functions should reflect some notion of the distance between two labels, but as seen from the perspective of the model and its features rather than from the perspective of the person engineering the model. In other words, the value of $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ should be proportional to the ease with which the model discriminates between $\mathbf{y}$ and $\hat{\mathbf{y}}$ given its features. This choice of cost gives the model the freedom to shift its weights so that there are wider margins between labels which it can easily discriminate, and smaller margins between labels which it has difficulty discriminating. Such a policy is particularly useful when there are classes of incorrect predictions that are difficult or impossible for the model to systematically resolve due to unreliably annotated output labels or a choice of features that is insufficient for the task.

There are several ways that we might quantify the ease with which the model discriminates between two labels, but for now, we will keep the notion of 'easiness' fuzzy while establishing its relation to the cost function. Basically, we propose that an incorrect prediction $\hat{\mathbf{y}}$ where the actual label is

[1] Tsiochantaridis et al also introduce an alternative 'slack re-scaling' technique, but the present paper mainly builds off of the 'margin re-scaling' approach.

$\mathbf{y}$ can fall into some number of 'incorrect prediction classes', and the overall cost of the incorrect prediction is the sum over the easiness of resolving (shrinking) each of these classes. More formally, let $\mathcal{S} \subseteq 2^{\mathcal{Y}^2}$ be a collection of incorrect prediction classes that collectively exhausts $\mathcal{Y}^2$. Assume that the 'easiness' with which a model of type $\mathcal{M}$ (e.g. SVM) given features $\mathbf{g}$ and data D resolves incorrect prediction class $S \in \mathcal{S}$ is given by $\mathcal{E}(S, \mathcal{M}, \mathbf{g}, D) \in \mathbb{R}$. Then, we assume that the cost is given by:

$$\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{S \in \mathcal{S}} \mathcal{E}(S, \mathcal{M}, \mathbf{g}, D) \mathbb{1}((\mathbf{y}, \hat{\mathbf{y}}) \in S) = \mathcal{E}^\top \mathcal{S}(\mathbf{y}, \hat{\mathbf{y}}) \tag{5}$$

There are many possible choices for prediction classes $\mathcal{S}$, but we focus on the following simplest ones:

$$\mathcal{S}_{[\mathcal{Y}]^2} = \{S_{\{\mathbf{y},\mathbf{y}'\}} | \mathbf{y}, \mathbf{y}' \in \mathcal{Y}\} \text{ where } S_{\{\mathbf{y},\mathbf{y}'\}} = \{(\mathbf{l}, \mathbf{l}') | (\mathbf{l} = \mathbf{y} \wedge \mathbf{l}' = \mathbf{y}') \vee (\mathbf{l} = \mathbf{y}' \wedge \mathbf{l} = \mathbf{y})\} \tag{6}$$

$$\mathcal{S}_{\mathcal{Y}^2} = \{S_{(\mathbf{y},\mathbf{y}')} | \mathbf{y}, \mathbf{y}' \in \mathcal{Y}\} \text{ where } S_{(\mathbf{y},\mathbf{y}')} = \{(\mathbf{y}, \mathbf{y}')\} \tag{7}$$

Given the above definitions, we desire a model which approximates $\mathcal{E}(S, \mathcal{M}, \mathbf{g}, D)$ (for some notion of 'easiness') in order to estimate the costs while simultaneous learning feature weights $\mathbf{w}$.

3 A Cost Learning Model

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

We propose a cost learning model that follows the assumptions in the previous section along with the intuition that the easiness $\mathcal{E}(S, \mathcal{M}, \mathbf{g}, D)$ of prediction class S for model class $\mathcal{M}$ is related to the size of the set:

$$S_{\mathcal{M},\mathbf{g},D} = \{i | (\mathbf{y}_i, \hat{f}^{\Delta}_{\mathcal{M},\mathbf{g},D}(\mathbf{x}_i; \mathbf{w}_{\mathcal{M},\mathbf{g},D})) \in S \text{ and } (\mathbf{x}_i, \mathbf{y}_i) \in D\} \tag{8}$$

Where $\mathbf{w}_{\mathcal{M},\mathbf{g},D}$ are the learned feature weights, and $\hat{f}^{\Delta}_{\mathcal{M},\mathbf{g},D}$ is the model's prediction function augmented with the cost:

$$\hat{f}^{\Delta}_{\mathcal{M},\mathbf{g},D}(\mathbf{x}_i; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \Big(F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}) + \Delta(\mathbf{y}_i, \mathbf{y}) \Big) \tag{9}$$

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 decrease 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 regardless 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. These observations suggest the following measure of easiness:

$$\mathcal{E}(S, \mathcal{M}, \mathbf{g}, D) = \max\left(0, 1 - \frac{|S_{\mathcal{M},\mathbf{g},D}|}{n_{\mathcal{M},\mathbf{g},D}}\right) \tag{10}$$

Where $n_{\mathcal{M},\mathbf{g},D}$ 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 is in $[0,1]$, and it has the property that if $|S_{\mathcal{M},\mathbf{g},D}| \geq n_{\mathcal{M},\mathbf{g},D}$, 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.

Given this definition of easiness, we modify the margin re-scaling SVM learning procedure given by quadratic program 4 to learn the cost function according to it. First, we transform this quadratic program into the equivalent unconstrained optimization problem which we find easier to work with:

$$\min_{\mathbf{w}} \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m}\left(-F(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}) + \max_{\mathbf{y}\in\mathcal{Y}}\Big(F(\mathbf{x}_i,\mathbf{y};\mathbf{w}) + \Delta(\mathbf{y}_i,\mathbf{y})\Big)\right) \tag{11}$$

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

$$\min_{\mathcal{E}\geq 0,\mathbf{w}} \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m}\left(-F(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}) + \max_{\mathbf{y}\in\mathcal{Y}}\Big(F(\mathbf{x}_i,\mathbf{y};\mathbf{w}) + \mathcal{E}^\top\mathcal{S}(\mathbf{y}_i,\mathbf{y})\Big)\right) - \mathcal{E}^\top\mathbf{n} + \|\mathcal{E}\|_{\mathbf{n}}^2 \tag{12}$$

4 Experiments

5 Discussion

5.1 Related Literature

5.2 Future Work

References