CS410 – Technology Review [Ritik Kulkarni (rk30)]

Overview of Beta Gamma Thresholding and possible improvements

Introduction:

Generic support vector machines (SVMs) provide excellent performance on a variety of learning problems including: handwritten character recognition, face detection and most recently text categorization. However, generic SVMs, when applied to text classification, lead to SVMs with excellent precision, but poor recall. Several attempts have been made to improve the recall of SVMs while not adversely affecting precision in a text classification context like uneven margin-based learning, cost-based learning, post-processing or thresholding the output value (or margin/score) of the learnt SVM. While the first two methods are applied during the learning step, the thresholding method is applied after the learning step, on the learnt SVM.

The paper being reviewed gives an overview of the third approach, i.e., beta gamma thresholding and presents a novel modification to this approach.

Body:

Brief overview of SVMs:

A learnt SVM model can be Geometrically (for linear support vector machines), seen as a hyperplane that separates a set of positive examples (belonging to the positive class) from a set of negative examples (negative class). Mathematically a hyperplane can be represented as follows: $(_{i=1}\sum^{n} w_i x_i) + b = 0$

This can be written more succinctly in vector format as <W, X>+b =0. Here W is known as a weight vector and corresponds to the normal vector to the separating hyperplane, H, and X is an input vector or document. b denotes the perpendicular distance from the hyperplane to the origin. n represents the number of input variables, in the case of text, this can be viewed as the number of words (or phrases, etc.) that are used to describe a document. The classification rule for an unlabelled document, X, using a support vector machine with separating hyperplane (W, b), is as follows: Class (X) = Sign (<W, X> + b).

The distance from the hyperplane to the nearest positive or negative examples is known as the margin of the SVM. Learning a linear SVM can be simply thought of as searching for a hyperplane (i.e., the weights and bias values) that separates the data with the largest margin. As a result, learning for linearly separable data can be viewed as the following optimization problem:

$$\text{minimize}\left(\frac{\|w\|^2}{2}\right) \quad \text{subject to}: \; y_i\left(\langle W, X_i\rangle + b\right) \geq 1 \quad \forall\, i = 1,\ldots,n$$

where,

Xi is a training example with label y_i, and ||W|| is the L^2 norm of the weight vector (i.e., $\sqrt{(\Sigma^{n}{}_{i=1}(wi * wi))}$.

Beta gamma thresholding:

In beta gamma thresholding, the critical step is in determining the threshold value, usually above which a document is classified as positive, and below it as negative. This step is independent of the learning step, and while inexpensive, optimizing this threshold is challenging. The main challenge arises from a lack of labelled training data. The limited amount of training data available is generally required for training the base model, thereby, resulting in a situation where it is rare to have an independent sample solely for threshold optimization. Due to this limited data, standard approaches to information retrieval use the same data for both learning the model and threshold optimization. Consequently, this often biases the threshold to high precision, i.e., overfits the training data.

In traditional text classification systems, such as content-based recommendation systems, the system is required to make a decision for each document, such as, if it belongs to a given class or not, or whether it is similar to what items a user likes or not. Most situations, require the decision to be

made as soon as the item arrives, and each decision is made independently. Therefore, creating a ranked list of items/documents is not a viable scenario. In such systems, the degree of satisfaction of a user may be expressed by a utility measure, and consequently, the goal of the classification system is to optimize this measure. The paper being reviewed adapts on the Linear Utility Measure; Linear Utility = 2R+ - N+, where R+, N+, R-, and N- are true positives, false positives, false negatives and true negatives respectively. This measure is also known as T10U. T10U provides a means of modelling a user in an information retrieval setting, whereby, each document with an actual label of C that was accepted by the model M, i.e., each true-positive document, denoted as R+, receives a utility of two points, whereas each negative document that is accepted by the model, i.e., a false-positive document, denoted as N+, receives a utility of minus one. It also shows empirical results, that optimizing for linear utility, with most of these utilities being asymmetric, leads to boosted performance of traditional information retrieval measures such as recall, precision and Fbeta. Thresholding strongly affects effectiveness, with no single threshold satisfying all utility measures.

SVM learning algorithms focus on finding the hyperplane that maximizes the margin since this criterion provides a good upper bound of the generalization error. Learning based on this criterion leads to models with very good ranking ability. However, the resulting separating hyperplane tends to be too conservative (high precision oriented). The natural threshold value for SVM learning and classification is zero.

The paper referenced in this review proposes to combine the powerful ranking ability of SVMs with the beta-gamma thresholding algorithm to reset the threshold of the learnt SVM in order to overcome this precision-oriented limitation. The powerful ranking ability of SVMs is only exploited for threshold adjustment, and is not used in classification (as each document is classified independently of each other). The beta-gamma thresholding algorithm relaxes the SVM threshold from zero, i.e., translates the SVM hyperplane towards the denser class (i.e., the class with more training data). Beta gamma thresholding adjusts the thresholds of generic SVMs by also incorporating a user utility model, which is an integral part of an information management system. By using thresholds based on utility models and the ranking properties of classifiers, it is possible to overcome the precision bias of SVMs and ensure robust performance in recall across a wide variety of topics, even when training data is sparse, while also preserving around the same level of precision.

This review provides an introduction to the beta gamma thresholding approach for setting the threshold of the learnt SVM and then reviews a novel technique for selecting the parameters of the threshold adjustment strategy automatically, based upon a combination of retrofitting and cross fold validation.

Basic beta-gamma thresholding strategy (1st Approach):

The core beta-gamma thresholding strategy uses as input a category label, C, a labelled dataset, T, of documents consisting of both positive and negative examples of C, a learnt SVM, M, that models the category C, β, the threshold adjustment parameter, and UtililtyMeasure, a utility measure that models the user's expectations.

The strategy consists of the following steps:

SetSVMThresholdUsingBetaGamma (C, T, M, β, UtililtyMeasure)

1) Rank the thresholding dataset, T, using the SVM, M, as scoring function, thereby yielding a ranked document list R consisting of tuples <Document, SVMScore>

2) Generate the cumulative utility curve for R, i.e., for each document in the ranked list R compute the cumulative utility using the utility measure UtililtyMeasure.

3) Determine the rank or indices of the maximum utility point on the cumulative curve and the first zero utility point following the maximum utility point. Denote these respectively as i_{Max}, and i_{Zero}. Assign the variables θ_{Max} and θ_{Zero} the output scores of the SVM, M, for the documents associated with the maximum and zero utility points respectively, i.e., the SVM scores of the documents at rank i_{Max}, and i_{Zero}.

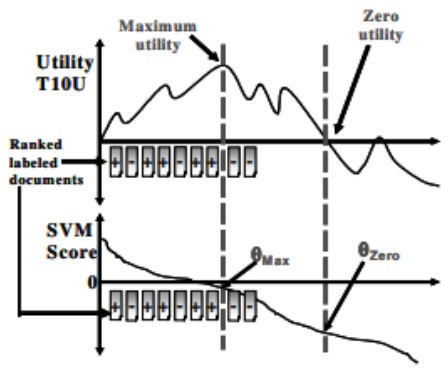
4) Return the threshold, θ, which is calculated as follows:

$\theta = (1-\beta)\,\theta_{Max} + \beta\theta_{Zero}$

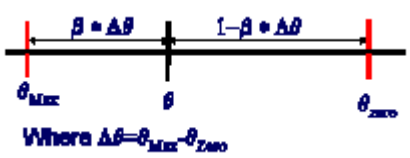
5)

$$\theta = \alpha\theta_{zero} + (1-\alpha)\theta_{Max}$$

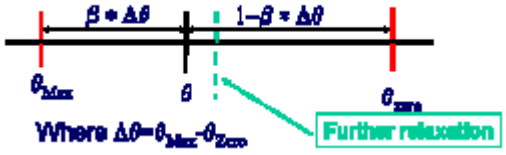
$$\alpha = \beta + (1-\beta)e^{-p\gamma}.$$



(a) Determining θ_{Max} and θ_{Zero} using a ranked list of training documents



(b) Beta relaxation of the threhsold using $\theta = (1-\beta)\,\theta_{Max} + \beta\theta_{Zero}$



(c) Beta relaxation of the threshold using equation in step 5

Step 4 presents the base version of the threshold adjustment algorithm, whereas step 5 presents an adjusted version of the algorithm which takes into account the number of positive training examples used in T.

In the equation presented in step 5, p denotes the number of positive documents in the thresholding dataset, T. The β value is the critical parameter to threshold relaxation, while γ is more stable across topics and corpora and is set heuristically. The γ component of this threshold relaxation formulation provides a mechanism to further relax the threshold based entirely upon β. The parameter γ provides additional control allowing the threshold to be further relaxed or constrained (with a negative value of γ).

In this first approach, both parameters β and γ, are provided heuristically, whereas in the second modified method, both parameters β and γ, will be selected automatically.

Modified beta gamma thresholding strategy (2nd Approach):

The second approach, based upon a combination of retrofitting and cross fold validation, selects the parameters of the threshold adjustment strategy automatically and optimizes them empirically, thereby rendering the approach parameter free (i.e., the user does not have to specify thresholding parameters).

Thus, the modified strategy uses as input a category label, C, a labelled dataset, T, of documents consisting of both positive and negative examples of C (for example, T could be a subset or the complete training dataset), a learnt SVM, M, that models the category C, β, the threshold adjustment parameter, UtililtyMeasure, a utility measure that models the user's expectations, βs (valid values for β are positive or negative real numbers), the set of possible beta values, γs, the set of possible gamma values, and n, the number of folds that will be used in parameter selection.

The modified strategy consists of the following steps:

SelectOptimalSVMThreshold (C, T, M, UtililtyMeasure, βs, γs, n)

1) Partition the data into n non-overlapping subsets of the data ensuring that both positive and negative documents are present in each fold or subset.
2) Foreach each combination of β and γ values in βs and γs do steps 3 and 4
3) Foreach fold n
- Set T_n to the n-1 folds
- Set θ = SetSVMThresholdUsingBetaGamma (C, T_n, M, β, γ, UtililtyMeasure)
- Set Utilityβγ = Calculate the utility for M and the threshold, θ, over the fold n.

4) Compute the average utility as follows: Utilityβγ= Utilityβγ/n
5) End Foreach
6) Calculate the optimal threshold, θ_{Opt}, using the β and γ combination that has the highest average utility Utilityβγ as follows: SetSVMThresholdUsingBetaGamma (C, T, M, β, γ, UtililtyMeasure)
7) Return θ_{Opt}

The SVM classification rule is altered slightly as follows to accommodate the adjusted threshold:
Class (X) = Sign (<W, X> + b - θ_{Opt})

Experimentation:

The paper being reviewed uses an adapted version of the T10U linear utility measure (T11SU) for threshold optimization, as it provides an intuitive user utility model that generally leads to improved recall and precision when used as a cost function in learning

The experiment represents a document as a vector of terms that is derived as follows:
1) Replace all numerical and punctuation characters by spaces.
2) Eliminate stop-words such as articles and prepositions, etc.
3) Each term is associated with a TFxIDF weight, where TF denotes the frequency of a term in a document, and IDF is calculated based on the distribution of the term in the training corpus.

In all experiments the document vectors were normalized to unit length and the experiment examined several information retrieval performance measures.

An evaluation of the learning threshold adjusted SVM classifiers (TSVMs) was performed on the following classification corpora: Reuters21578 ModApte split collection and TREC2001 corpus.

Results:

The proposed thresholding approach was compared against the following approaches: baseline (unthresholded) SVMs, other threshold adjusting SVM approaches, asymmetric (misclassification costs) SVMs, and traditional IR approaches.

Overall, the paper being reviewed, states that, adjusting the threshold using the beta-gamma procedure boosts the performance of the baseline SVM on all examined evaluation measures at a macro level for the Reuter-21578 corpus. When examining each topic from a T11SU perspective, it was noticed that the biggest improvement in performance comes from topics that have fewer than fifty positive training documents, topics that have traditionally been very difficult to model. Overall, 80% of the topics have improved or have not been adversely affected by this procedure.

Adjusting the threshold of the SVM for the TREC2001 topics has boosted recall and therefore led to over 20% improvement in terms of T11SU performance over baseline SVMs (linear SVM), while not effecting precision. This performance is comparable with the best performer for this text classification task that was generated using asymmetric SVMs. The following observations were made when comparing the evaluation measures for the threshold adjusted experiment and the asymmetric experiment run:

Due to the expensive cross fold validation required for determining the asymmetric costs of the SVM learning, training the asymmetric SVMs took two orders of magnitude more time to learn than the threshold adjusted SVMs (i.e., 500 hours for the asymmetric experiment versus 5 hours for the threshold experiment). The experiment with asymmetric SVMs provides 14% better precision than the threshold adjusted run. The threshold adjusted run provides 11% better recall than the asymmetric

run. This would seem to suggest that asymmetric SVMs and adjusting the threshold are addressing two independent aspects of the problem, which if combined could boost performance even further.

Conclusion:

SVMs are used for a variety of learning problems and provides state-of-the-art performance in dense data problems, but when applied to text categorization, where training data is sparse and the classes are unevenly distributed and poorly represented, these learning algorithms can lead to an over fitting of the more frequent class, which lead to learnt SVMs with good precision, but poor recall. To remedy this, various relaxation strategies are used to improve recall, without affecting precision in a text classification context. The paper being reviewed looks at one such relaxation strategy, called the threshold adjustment approach, which is applied to the SVM after the learning step, as a post-processing step.

While there are many adjustment algorithms, the approach that the paper reviews is the beta-gamma thresholding strategy and a novel modification to this strategy, which uses retrofitting and cross-validation to automatically determine the optimal parameters for the beta-gamma algorithm, which are subsequently used to relax the threshold of the class model.

The novel modification of automatic parameter selection for the beta gamma thresholding strategy is parameter free, relying on a process of retrofitting and cross validation to set algorithm parameters empirically, whereas the first approach required the specification of two parameters (beta and gamma). The proposed approach is more efficient, does not require the specification of any parameters, and similarly to the parameter-based approach, boosts the performance of baseline SVMs by at least 20% for standard information retrieval measures.

The modified approach is a continuous thresholding relaxation procedure as the value of beta (and gamma can be set to a constant given the nature of the beta) is determined using a process of retrofitting and cross validation. This is different from the core beta gamma thresholding algorithm, which is discrete in nature, requiring the user to provide a list of possible values for beta and gamma, which the system would empirically select through cross validation. This modified continuous thresholding relaxation procedure is also more efficient than the discrete approach since the value of β is determined using retrofitting, thereby, alleviating the need for a cross validation exploration of alternative β values.

The reviewed paper shows a gain in performance for examined TREC corpora of over 20% for standard information retrieval measures when compared to baseline SVMs.

This thresholding approach boosts the recall performance of baseline SVMs for text classification, while not adversely affecting precision. The extra cost of performing this threshold adjustment is small, in that it is a one-dimensional optimization problem. As stated before, threshold adjustment of SVMs is just one technique for boosting the performance of SVMs. Combining this adjustment algorithm with other techniques, such as asymmetric cost-based learning of SVMs, should lead to even better performance. In addition, since the thresholding approach is independent of the learnt SVM model, using it in conjunction with other types of models may also lead to interesting results.

References:

[1] Shanahan, James G. and Norbert Roma. “Improving SVM Text Classification Performance through Threshold Adjustment.” *ECML* (2003).

[2] Shanahan, James G. and Norbert Roma. “Boosting support vector machines for text classification through parameter-free threshold relaxation.” *CIKM '03* (2003).