Explorations in Text Classification Using Support Vector Machines

Abdi-Hakin Dirie · Jason Tong
{abdihd, jktong}@mit.edu

I. Problem Statement

The project will recreate the SVM experiments performed in Joachims' paper [1]. The specific task is to classify news articles published by Reuters into multiple categories that represent various topics an article can address. Each article can be categorized into any number of topics, or none of them. The task at hand is a variant of the problem of text classification, and is a key problem in the field of information retrieval.

II. Components

A. Data Parsing and Feature Selection

The first step of this project will involve transforming the Reuters corpus into a representation amenable to manipulation with SVMs. [1] outlines some details regarding the choice of features, such as the prerequisite that a word occur at least thrice in the testing data, or that a word not be a stopwords. As the set of stopwords is left unspecified by [1], we will need to determine the set ourselves. It is plausible that we will have to return to this step to amend our feature representation after inspecting initial results, so the infrastructure should be designed with flexibility in mind.

B. Support Vector Machine Infrastructure

As described by [1], the multi-class classification problem is solved as distinct binary classification problems determining whether a particular text belongs in a given category. This allows for any text to be classified into multiple categories. [1] uses SVM^{light}, an SVM package implemented in C. We will be using our own implementation of SVM. In order to accommodate the possibility that the documents are linearly inseparable for a given class, we will implement soft SVM although this is not specified by [1]. To solve the quadratic programming problem of optimizing the SVM, we will using the `cvxopt` library.

III. Experiments

A. Models

In order to train our SVM models, we will have to choose the appropriate kernel function and the hyperparameters associated with that kernel. We will be using polynomial kernels of varying degrees and radial basis function (RBF) kernels of varying bandwidths. We will also do a search over the slack penalty factor that is used in soft SVM. To allow for comparison, candidate kernels, degrees, and bandwidths are taken from [1]. Choices for the slack penalty factor will be determined by us.

B. Evaluation

We are given 9,603 articles in the training set, and 3,299 articles in the test set. 8,676 articles are unused. This is the Modified Apte Split (ModApte) of the data, which is the split used in [1]. The chosen method of evaluation is the F_1 score, a common evaluation metric used in information retrieval. The F_1 score is the harmonic mean of the precision and recall of the system:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Precision and recall are defined as:

$$precision = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \qquad recall = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

We will have an F_1 score for each of the 10 most frequent topics used in [1]. We will also have a micro-averaged F_1 score over all the Reuters categories.

It must be noted that [1] used a metric called the precision-recall breakeven point for evaluation. The precision-recall breakeven point is the maximum precision such that precision is equal to recall, which is also the maximum recall under the same constraint. We will use the F_1 score instead of the precision-recall breakeven point because we believe it will be enlightening to evaluate the efficacy of SVMs along another metric. We hope that the results of experiments will corroborate the finding in [1] that SVMs are useful for text classification.

IV. Risks

As with any attempt to replicate the results of a paper, we expect to discover challenges not addressed in [1] that may pose significant obstacles to this project's progress. As these are unforeseeable, we have allocated additional time for the process of tuning our implementation of SVMs in the timeline.

Additionally, because we will be working with a high-dimensional feature space, we may run into memory limitations. We expect to be able to overcome challenges of this nature through careful allocation of memory and garbage collection. If necessary, we will also consider the option of working with a subset of the data.

V. Timeline

1. Nov 18 – Finish parsing data and transforming into features
2. Nov 22 – Finish SVM implementation
3. Nov 25 – Preliminary results for polynomial and RBF SVMs
4. Dec 3 – Final results for polynomial and RBF SVMs
5. Dec 8 – Paper Due Date

References

[1] T. Joachims. “Text categorization with support vector machines: Learning with many relevant features.” Springer Berlin Heidelberg, 1998.