October 3, 2014

Background
- Text classification has an enormous number of practical applications, from spam filtering [2], to triaging helpdesk requests, to recommender systems (using review text) [**?**]. An exciting recent application is the inference of latent attributes about a user based on their posts in social media [**?**, **?**, **?**].

- There is is a rich body of literature that has explored many aspects of the text classification problem. Chosing the set of features to represent documents is critical to accurate classification, and various options have been investigated [1, 2]. Features may consist of frequency counds, boolean variables that only indicate whether a word is present (but don't show its frequency) [1]. But often, before feautures are calculated, some pre-processing of the documents is often done. The most frequent and most rare words are sometimes ignored because they tend to be less descriminating [1]. A list of stopwords can be used, which is a set of common words deemed to be not useful for text classification purposes [1]. The words in the documents may also be simplified by stemming or lemmatization, for example, to convert `running` and `ran` both to the more basic infinitive form `run` [1]. Once Feature vectors are obtained, they can be further treated by multiplying the feature vectors (component wise) by a weighting scheme [1]. The application of weights has a dual purpose. Primarily, it regulates the degree of influence that given features have over the decision boundary to be learned, which ideally should be made to reflect the degree to which given features are descriminative [1]. Secondarily, it can be used to normalize the vectors to unit length according to some metric (usually the euclidean distance or L2-norm) [1]. Normalization helps overcome the fact that longer documents have higher frequencies overall, which otherwise can be a source of bias [1].

In addition to choosing how to represent the data, one must choose the learning algorithm. The performance of learning algorithms have been scrutinized in a wide variety of contexts. Roughly ordered in order of increasing accuracy, the algorithms that have been found useful in text classification include: naive Bayes (NB), k-nearest-neighbor (kNN), latent Dirichlet allocation (LDA), and support vector machines (SVMs).

Choice of inference method (learner)

Preprocessing
- lematization offers little or no benefit when using SVM [1]

Normalization
- l2-normilization is helpful for SVM [1]

Choice of term weighting scheme
- The choice of weighting scheme is more important than the choice of kernel [1]
- redundancy does better than idf [1]

SVM - the rbf kernal appear beats linear and polynomial [1]

References

[1] Edda Leopold and Jörg Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1-3):423–444, 2002.

[2] Deqing Wang and Hui Zhang. Inverse-category-frequency based supervised term weighting scheme for text categorization. *Journal of Information Science and Engineering*, 29:209–225, 2013.