

Learning to Filter Junk E-Mail from Positive and Unlabeled Examples

Karl-Michael Schneider

Department of General Linguistics

University of Passau

`schneide@phil.uni-passau.de`

Abstract

We study the applicability of partially supervised text classification to junk mail filtering, where a given set of junk messages serve as positive examples while the messages received by a user are unlabeled examples, but there are no negative examples. Supplying a junk mail filter with a large set of junk mails could result in an algorithm that learns to filter junk mail without user intervention and thus would significantly improve the usability of an e-mail client. We study several learning algorithms that take care of the unlabeled examples in different ways and present experimental results.

1 Introduction

Junk mail filtering is the problem of detecting junk messages (spam, UCE, unrequested mass e-mailings of commercial, pornographic or otherwise inappropriate content) in a stream of e-mails received by a user. Under the assumption that junk mail can be distinguished from legitimate (non junk) mail in terms of style and vocabulary, the problem can be rephrased as a text classification problem, and machine learning techniques can be employed to learn a classifier from examples (Sahami et al., 1998; Drucker et al., 1999).

A supervised learning algorithm needs examples of junk mail and legitimate mail to learn the difference in the distribution of words between the two classes. It cannot be pretrained by a

software vendor, because legitimate mail depends very much on the individual user. Rather, examples of legitimate and junk mail must be given to the learner by the user.

From a user's perspective it would be preferable to have a junk filter that requires as little intervention from the user as possible, in order to learn to filter junk mail. It is long known that unlabeled data can reduce the amount of labeled data that is required to learn a classifier (Dempster et al., 1977), and recently it has been shown how unlabeled data can be exploited in text classification (Nigam et al., 2000). Moreover, one can even learn text classifiers from labeled examples of one class alone, when these are augmented with unlabeled examples. This type of learning to classify text is called *partially supervised text classification* (Denis et al., 2002; Liu et al., 2002).

The class for which a partially supervised learner has access to labeled examples is called the positive class. In the context of junk mail filtering, junk mails constitute the positive class, while the e-mails received by the user are unlabeled examples. Since junk e-mail can be collected automatically quite easily in large quantities and with high purity, it can be given to the learner as positive examples by the vendor of a junk filter instead by the user.

In this paper we study the applicability of learning from positive and unlabeled examples for junk mail filtering. An algorithm that learns to filter junk mail from the user's incoming e-mail when supplied with a large collection of junk mail, without user intervention, could significantly improve the usability of an e-mail client.

In typical applications of partially supervised text classification, e.g. the identification of interesting websites for a user, the positive examples usually depend on the user, while there is a large set of unlabeled examples. In contrast, in junk mail filtering, unlabeled examples (e-mails received by a user) are not available instantly in large quantities but must be accumulated over time, while positive examples (junk e-mails) do not depend on a particular user and are readily available in large quantities.

In this paper, we study the situation where a learner is initially given a sufficiently large set of positive examples (junk e-mails), and receives unlabeled examples from a stream (messages received by the user). The learner's task is to label each unlabeled example as positive (junk) or negative (legitimate) when it is received from the stream. At the same time, the learner must employ some strategy to exploit the information about the negative class contained in the examples from the stream, as the stream is the only source of negative examples. We study several learner architectures with different strategies and compare their learning success rates.

The paper proceeds as follows. In Sect. 2 we discuss related work, especially in the framework of partially supervised text classification. In Sect. 3 we define the statistical framework used in this paper. In Sect. 4 we describe the different learner architectures studied in this paper. Sect. 5 describes some experiments and discusses the results. Finally, in Sect. 6 we draw some conclusions.

2 Related Work

That learning from positive examples alone with the help of unlabeled examples is possible was shown by Denis (1998) in a theoretical study in the context of PAC learning. Liu et al. (2002) proved an upper bound on the expected error of a classifier that is selected such that all positive examples are classified correctly, and the number of unlabeled examples classified as positive is minimized. In the case where the class labels are noisy or the hypotheses space is not powerful enough to learn the true classification, the expected error can still be bounded by choosing a classifier that correctly classifies a fraction r of the positive ex-

amples.

Unfortunately the number of unlabeled examples required to obtain a good classifier is quite large. A junk mail filter must collect unlabeled examples during operation and has no unlabeled examples at all when it gets operational for the first time. Standard methods for incorporating unlabeled examples, like the EM algorithm (Dempster et al., 1977; Nigam et al., 2000) may not work well in this situation. The question addressed in this paper is thus how good a learner can be when it has many positive examples but few unlabeled examples, and how fast it can improve as it collects more unlabeled examples.

The key problem in partially supervised classification is to identify negative examples in the unlabeled data to train a classifier. Li and Liu (2003) use the Rochio algorithm to find reliable negative examples among the unlabeled examples, and then apply a support vector machine (SVM) iteratively to find more negative examples, and to select a classifier. Yu et al. (2002) remove positive examples from the unlabeled data by identifying strong positive features, and use the remaining data as negative examples to train an SVM.

A different method to obtain a classifier from positive and unlabeled examples was proposed by Denis et al. (2002). Instead of finding negative examples, the *positive Naive Bayes* algorithm estimates the distribution of words in the negative class from the distributions in the positive and unlabeled examples. We present this algorithm in Sect. 3.2.

Another problem closely related to partially supervised text classification is the problem of describing a class of objects when only examples of that class are given. This is sometimes called *one-class classification* or *novelty detection* (Manevitz and Yousef, 2001). The one-class SVM is a variant of the traditional two-class SVM that identifies outliers in the set of examples and then uses traditional SVM to estimate the support of the target class. One-class classification does not require unlabeled examples.

The problem of learning during operation is also related to online learning, where the parameters of a classifier are constantly updated as new labeled examples are received (Chai et al., 2002). To the best of our knowledge, online classifiers

have not been applied to the situation where only positive labeled examples are available. In addition, whereas an online classifier is first trained from an initial training set and then updated during operation, this paper studies learning that begins when the first input message is received, with no prior training.

3 Bayesian Framework

3.1 Naive Bayes

We use a Naive Bayesian classifier to classify messages as junk or legitimate. The Naive Bayes classifier is based on a probabilistic model of text generation. It assumes that a message is generated by a mixture model by first choosing the category $c_j \in \mathcal{C} = \{\text{junk}, \text{legit}\}$ of the message according to a prior probability $P(c_j|\theta)$, where θ denotes the mixture parameters, and then generating the message according to some probabilistic model whose parameters depend on c_j . We use a multinomial model of text generation (McCallum and Nigam, 1998). In this model, a document d_i is generated by choosing its length $|d_i|$ according to some distribution $P(|d_i|)$ and then drawing $|d_i|$ words independently from a fixed vocabulary V according to the distribution $P(w_t|c_j; \theta)$. The probability of d_i in c_j is given by the multinomial distribution:

$$P(d_i|c_j; \theta) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j; \theta)^{N_{it}}}{N_{it}!} \quad (1)$$

where N_{it} is the number of occurrences of w_t in d_i . The likelihood of d_i is

$$P(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} P(c_j|\theta)P(d_i|c_j; \theta). \quad (2)$$

Using Bayes' rule, we can compute the posterior probability that d_i belongs to c_j :

$$P(c_j|d_i; \theta) = \frac{P(c_j|\theta)P(d_i|c_j; \theta)}{P(d_i|\theta)}. \quad (3)$$

For a given document d_i the Naive Bayes classifier selects the category c^* that maximizes (3).

Let $\mathcal{D} = \{d_1, \dots, d_m\}$ be a set of labeled training documents. The learning algorithm must estimate the parameters $P(c_j|\theta)$ and $P(w_t|c_j; \theta)$

from \mathcal{D} . When \mathcal{D} contains examples of both classes, this can be done using maximum likelihood estimates with Laplacean priors:

$$P(c_j|\hat{\theta}) = \frac{|c_j|}{|\mathcal{D}|} \quad (4)$$

and

$$P(w_t|c_j; \hat{\theta}) = \frac{1 + N(w_t, c_j)}{|V| + \sum_{s=1}^{|V|} N(w_s, c_j)} \quad (5)$$

where $|c_j|$ is the number of documents in c_j and $N(w_t, c_j)$ is the number of occurrences of w_t in c_j .

3.2 Positive Naive Bayes

When the learner has only labeled examples of one class, but has access to unlabeled examples, it must estimate the parameters for the other class from the combination of labeled and unlabeled examples. Let 0 and 1 denote the negative and positive class, respectively. Denis et al. (2002) have shown how $P(0|\theta)$ and $P(w_t|0; \theta)$ can be estimated from positive and unlabeled examples when $P(1|\theta)$ is known.¹ We set $P(0|\hat{\theta}) = 1 - P(1|\theta)$. Let \mathcal{P} and \mathcal{U} denote the sets of positive and unlabeled examples, respectively. $P(w_t|0; \theta)$ will be estimated by projecting \mathcal{U} to a set $\mathcal{P} \cup \mathcal{N}$ of positive and negative documents such that the positive and negative documents are distributed according to $P(c_j|\theta)$, and the distribution of words in $\mathcal{P} \cup \mathcal{N}$ is most similar to \mathcal{U} . Then $P(w_t|0; \theta)$ can be estimated by estimating the number of occurrences of w_t in \mathcal{N} .

The estimated number of negative documents is

$$|\mathcal{N}| = |\mathcal{P}| \cdot \frac{1 - P(1|\theta)}{P(1|\theta)}. \quad (6)$$

Let $N(\mathcal{X})$, $N(w)$ and $N(w, \mathcal{X})$ denote the number of word occurrences in \mathcal{X} , the number of occurrences of w in $\mathcal{P} \cup \mathcal{N}$ and the number of occurrences of w in \mathcal{X} , respectively. Assuming that documents in \mathcal{U} are generated according to the distribution given by θ , the estimated number of

¹Our presentation differs slightly from Denis et al. (2002).

word occurrences in $\mathcal{P} \cup \mathcal{N}$ is

$$\hat{N}(\mathcal{P} \cup \mathcal{N}) = \max \left\{ N(\mathcal{U}) \cdot \frac{|\mathcal{P}| + |\mathcal{N}|}{|\mathcal{U}|}, N(\mathcal{P}) \cdot \frac{2}{1 + P(1|\theta)} \right\}. \quad (7)$$

The second term is used as a lower bound to avoid documents of zero or negative length, especially when \mathcal{U} is small (Denis et al., 2002). The estimated number of word occurrences in \mathcal{N} is

$$\hat{N}(\mathcal{N}) = \hat{N}(\mathcal{P} \cup \mathcal{N}) - N(\mathcal{P}). \quad (8)$$

Similarly, the estimated number of occurrences of w in $\mathcal{P} \cup \mathcal{N}$ is

$$\hat{N}(w) = \max \left\{ \hat{N}(\mathcal{P} \cup \mathcal{N}) \cdot \frac{N(w, \mathcal{U})}{N(\mathcal{U})}, N(w, \mathcal{P}) \right\}. \quad (9)$$

Then

$$\hat{N}(w, \mathcal{N}) = \min \{ \hat{N}(w) - N(w, \mathcal{P}), \hat{N}(\mathcal{N}) \} \quad (10)$$

is the estimated number of occurrences of w in \mathcal{N} .

4 Learner Architectures

We study the following scenario: A learner receives messages from a stream (called input messages), one at a time. Its task is to build a classifier and classify each input message when it is received. The category of the input message (junk or legitimate) is not known to the learner. The learner can build a new classifier for each new input message. To this end, it can memorize the input messages it has received. In addition, it has access to a fixed set of junk messages.

We study four different learner architectures. Each learner memorizes all the input messages it receives and employs a particular strategy to use the junk messages (positive examples) and the input messages (unlabeled examples) to build a Naive Bayes classifier. The *static NB learner* (SNB) simply considers each input message as a negative example (i.e. legitimate message) and uses the standard Naive Bayes parameter estimation as described in Sect. 3.1. The *static PNB learner* (SPNB) considers the stored input messages as mixed messages (i.e. junk and legitimate)

and uses the positive Naive Bayes parameter estimation as described in Sect. 3.2. The *static EM learner* (SEM) uses the EM algorithm (Dempster et al., 1977; Nigam et al., 2000) to build a classifier. It builds an initial model by considering the stored input messages as negative examples, and then uses this model to assign probabilistic labels to the stored messages and iterates until the model converges or a maximum number of iterations is reached. The *static PEM learner* (SPEM) is like the static EM learner except that for the initial model, it considers the stored input messages as mixed examples and uses the positive Naive Bayes parameter estimation. We expect the model built by the static learners to improve over time as the number of negative examples in the memory increases.²

5 Experiments

We performed experiments with the four learners described in Sect. 4 on two different e-mail corpora. The first corpus consists of 1397 junk messages and 500 mixed messages from the SpamAssassin public mail corpus.³ We varied the amount of spam messages among the mixed messages between 15% and 50%. For the second corpus we used 2000 junk messages received by the author between January 22, 1998 and August 22, 2002, and 500 mixed e-mails received between October 16 and October 20, 2003 as input messages. 43% of the mixed messages were spam messages. We extracted the text of each message and removed everything else, including all e-mail headers. The text was then tokenized, counting alphabetic sequences, numbers and all other characters as tokens. We did not perform stemming, case normalization, removal of stop words or other preprocessing.

Each input message was fed to the learner, and the predicted label (junk or legitimate) was recorded. In order to reduce the size of the model and to avoid overfitting, we performed feature selection using mutual information (McCallum and Nigam, 1998). We used 5000 features for all

²Note that the static learners do not store the labels assigned to the input messages by the classifier. In future experiments, we plan to consider dynamic learners that memorize the labels assigned to the input messages and use the input messages as labeled training examples.

³Available from <http://spamassassin.org/publiccorpus/>

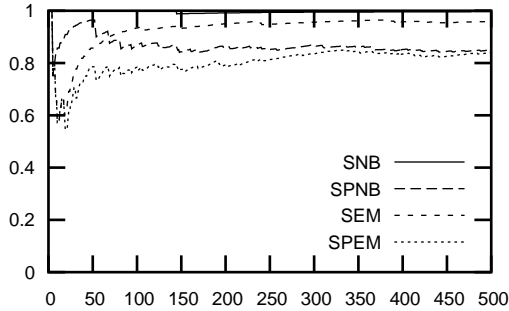


Figure 1: Accumulated legitimate recall.

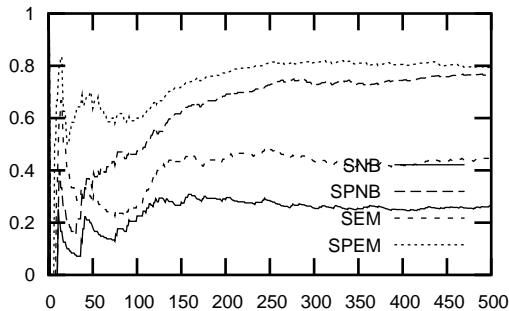


Figure 2: Accumulated spam recall.

learners except SNB, for which we found that varying the number of features linearly from 500 for the first input message to 5000 for the 500th input message increased the classification accuracy slightly.

For SEM and SPEM, the number of EM iterations was limited to 5. We found that allowing more iterations had no effect on the classification. For SPNB and SPEM, the prior probability of the positive examples was fixed in each experiment and was given to the learner.

Precision is the percentage of true legitimate messages among the predicted legitimate messages, and recall is the percentage of true legitimate messages that are classified as legitimate:

Figures. 1 and 2 show legitimate recall and spam recall (the fraction of legitimate/spam messages that are classified as legitimate/spam) for the SpamAssassin messages with a spam proportion (spam prior) of 40%, accumulated over the number of messages received, i.e. at any number of messages, the recall value for the messages received up to that point is shown. SNB and SEM have higher legitimate recall but much lower spam recall than SPNB and SPEM which consider the input messages as mixed messages for

Junk	SNB		SPNB	
30	0.9966	0.0146	0.7177	0.2816
60	0.9966	0.0194	0.6701	0.3010
125	0.9966	0.0291	0.5782	0.4466
500	0.9966	0.1408	0.6497	0.7282
1397	0.9966	0.2670	0.8469	0.7670

Table 2: Legitimate/spam recall for different amounts of junk messages.

training.

In the next experiment, we investigated the effect of the spam proportion on the behaviour of the learners. Table 1 shows classification results for four different versions of the SpamAssassin (SA) corpus, where the spam proportion varies from 15% to 50%. In addition, the results for the private E-mail corpus is shown. In general, the learners that use the positive Naive Bayes parameter estimation have lower legitimate recall but higher spam recall, for all spam proportions, but the effect gets more pronounced for higher spam rates.

In another experiment, we varied the number of junk messages given to the learner. We show results for SNB and SPNB on the SpamAssassin corpus with 40% spam in Table 2. Both SNB and SPNB loose spam recall when provided with less junk messages, but for SNB the effect is much more pronounced.

6 Conclusions

We have defined the setting of learning to filter junk mail from the e-mails received by a user, without intervention of the user, by providing the learner with a large set of junk mails as positive examples. We have described several learner architectures for this setting. In particular, we compared two strategies to exploit the unlabeled messages. The first one considers all messages received by the user as legitimate (i.e. non junk) messages, while the second one uses the positive Naive Bayes algorithm of Denis et al. (2002) to estimate the distribution of the negative class from the positive and unlabeled examples.

Our experiments show that the positive Naive Bayes (PNB) algorithm achieves considerably higher spam recall, but at the cost of lower le-

	SA/15%		SA/25%		SA/40%		SA/50% ⁴		E-mail/43%	
SNB	0.9977	0.4595	0.9945	0.3060	0.9966	0.2670	1.0000	0.2120	0.8550	0.6429
SPNB	0.9742	0.6486	0.9590	0.7313	0.8469	0.7670	0.8800	0.8200	0.6298	0.8193
SEM	0.9765	0.5000	0.9809	0.4701	0.9558	0.4515	0.9880	0.3800	0.9084	0.5882
SPEM	0.7277	0.7568	0.9016	0.8507	0.8367	0.7961	0.9600	0.6240	0.6298	0.8361

Table 1: Legitimate recall and spam recall for different spam proportions.

gitimate recall. Since higher spam recall means better spam filtering and lower legitimate recall means more false positives (i.e. legitimate e-mails classified as spam), the PNB algorithm may be more appropriate in situations where false positives are associated with lower costs, or additional measures must be taken to avoid false positives. This indicates directions for future work.

Acknowledgements

The author would like to thank the anonymous reviewers for their helpful suggestions.

References

- Kian Ming Adam Chai, Hwee Tou Ng, and Hai Leong Chieu. 2002. Bayesian online classifiers for text classification and filtering. In *Proc. 25th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 97–104.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- François Denis, Rémi Gilleron, and Marc Tommasi. 2002. Text classification from positive and unlabeled examples. In *Proc. 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, pages 1927–1934.
- François Denis. 1998. PAC learning from positive statistical queries. In *Proc. 9th International Workshop on Algorithmic Learning Theory (ALT'98)*, LNAI 1501, pages 112–126. Springer-Verlag.
- Harris Drucker, Donghui Wu, and Vladimir N. Vapnik. 1999. Support vector machines for spam categorization. *IEEE Trans. on Neural Networks*, 10(5):1048–1054.
- Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico.
- Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Proc. 19th International Conference on Machine Learning (ICML-2002)*, pages 387–394.
- Larry M. Manevitz and Malik Yousef. 2001. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154.
- Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for Naive Bayes text classification. In *Learning for Text Categorization: Papers from the AAAI Workshop*, pages 41–48. AAAI Press. Technical Report WS-98-05.
- Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134.
- Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the AAAI Workshop*, pages 55–62, Madison Wisconsin. AAAI Press. Technical Report WS-98-05.
- Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. 2002. PEBL: Positive example based learning for web page classification using SVM. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD'02)*, pages 239–248, New York. ACM Press.

⁴SPEM on the SpamAssassin corpus with 50% spam used only 500 junk messages, therefore the results are not directly comparable.