

# Instance-Based Spam Filtering Using SVM Nearest Neighbor Classifier

**Enrico Blanzieri**

University of Trento, Italy

blanzier@dit.unitn.it

**Anton Bryl**

University of Trento, Italy;

Create-Net, Italy

abryl@dit.unitn.it

## Abstract

In this paper we evaluate an instance-based spam filter based on the SVM nearest neighbor (SVM-NN) classifier, which combines the ideas of SVM and  $k$ -nearest neighbor. To label a message the classifier first finds  $k$  nearest labeled messages, and then an SVM model is trained on these  $k$  samples and used to label the unknown sample. Here we present preliminary results of the comparison of SVM-NN with SVM and  $k$ -NN.

## Introduction

Unsolicited bulk email, also called spam, is a serious problem of today's Internet, and one of the popular ways of anti-spam protection is filtering. The Support Vector Machine (SVM) classifier was proposed for spam filtering by Drucker, Wu, & Vapnik (1999), and proved to show good results. A filter based on the  $k$ -nearest neighbor ( $k$ -NN) algorithm was introduced by Androulatsopoulos *et al.* (2000b) and is shown to have relatively low accuracy (Zhang, Zhu, & Yao 2004). For a more detailed overview of spam filtering approaches see the survey by Blanzieri and Bryl (2006).

In this paper we evaluate the performance of the SVM nearest neighbor classifier (SVM-NN) (Blanzieri & Melgani 2006) on spam filtering. SVM-NN is an instance-based learning technique that combines the  $k$ -NN approach with an SVM-based decision rule. The combination of SVM and  $k$ -NN achieves a smaller generalization error bound produced by the combination of a bigger margin and a smaller ball containing the points. The motivation for applying this method to spam filtering is the following. Spam is known to be not uniform, but rather to consist of messages that vary seriously, in particular in topic (Hulten *et al.* 2004). The same can be said of legitimate mail as well. This suggests that a classifier which works on the local level can be meaningful here. As a quality baseline we used the SVM classifier, reproducing the experiment of Zhang, Zhu, & Yao (2004). Also the  $k$ -NN classifier was involved in the comparison.

## SVM Nearest Neighbor Spam Filter

The SVM nearest neighbor classifier (Blanzieri & Melgani 2006) combines the ideas of the SVM (Cortes & Vapnik

1995) and the  $k$ -NN classifiers. It works in the following way: given a sample to classify, the filter determines  $k$  nearest samples in the training data, and an SVM classifier is trained on these samples. Then, the trained SVM classifier is used to classify the unknown sample. More formally, the algorithm can be defined as follows:

### Algorithm: SVM Nearest Neighbor

**Inputs:** sample  $x$  to classify; training set  $T = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ ; number of nearest neighbors  $k$ ; threshold  $t$ .

**Output:** decision  $y \in \{-1, 1\}$

Find  $k$  samples  $(x_i, y_i)$  with minimal values of  $K(x_i, x_i) - 2 * K(x_i, x)$

Train SVM model on the  $k$  selected samples.

Classify  $x$  using SVM, get the result in the form of a real number.

Make the decision using the threshold  $t$ .

In our experiments we use the Euclidean metric for determining nearest neighbors and the linear kernel for SVM. Feature selection is performed as in the work of Zhang, Zhu, & Yao (2004). Each message is considered as an unordered set of strings (tokens) separated by spaces. Presence of a certain token in a certain part of a message, namely in the body or in a field of the header, is considered a binary feature of this message. Then, the  $d$  most frequent features in the training data are selected and used. Thus, each message is represented by a vector of  $d$  binary features.

In order to evaluate the performance of SVM-NN filter we run an experimental comparison. To have a verifiable baseline we decided to reproduce one of the experiments with the SVM classifier by Zhang et al. (Zhang, Zhu, & Yao 2004), namely ten-fold cross-validation on the SpamAssassin corpus<sup>1</sup>. The corpus contains 4150 legitimate and 1897 spam messages. The performance measure used for parameter optimization is Total Cost Ratio (TCR) (Androulatsopoulos *et al.* 2000a). It is defined as follows:  $TCR = n_S / (\lambda \cdot n_{L \rightarrow S} + n_{S \rightarrow L})$ , where  $n_S$  is the total number of spam messages,  $n_{L \rightarrow S}$  is the number of legitimate messages classified as spam,  $n_{S \rightarrow L}$  is the number of spam messages classified as legitimate, and  $\lambda$  is the relative cost of misclassification of a legitimate message and misclassification of a spam message. As Zhang et al., we set  $\lambda$  equal to 9. The same random splittings of the data were used in all runs. Two variants of SVM-NN were involved in the

<sup>1</sup> Available at: <http://spamassassin.apache.org/publiccorpus/>

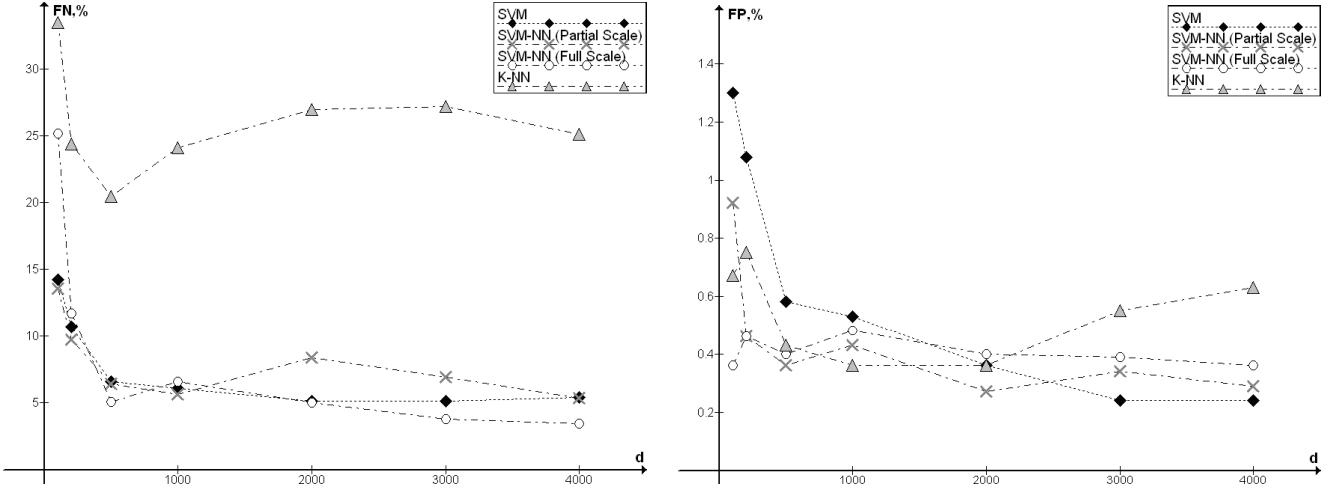


Figure 1: FN is false negative rate, FP is false positive rate,  $d$  is the number of features. SVM is an SVM classifier; the threshold was taken equal to 0.48 as proposed by Zhang et al. SVM-NN (**Partial Scale**) and SVM-NN (**Full Scale**) are SVM-NN classifiers. The optimal values of  $k$  and  $t$  are estimated using the training data. For SVM-NN (**Partial Scale**) the search for optimal  $k$  is performed only among the values up to about 25% of the whole data. K-NN is a  $k$ -NN classifier with a decision rule based on voting; the optimal  $k$  and the optimal relative cost of the votes from different classes were estimated using the training data. In all cases the optimal parameters were estimated separately for each of the ten folders of cross-validation.

comparison: SVM-NN (**Partial Scale**) and SVM-NN (**Full Scale**), the first of the two searching for the optimal value of  $k$  only among comparatively low values and therefore is faster but less accurate. We used a software realization of SVM, called SVMlight<sup>2</sup> (Joachims 1999), that provides not binary, but real number output, giving the possibility to apply different thresholds to the result.

On Figure 1 the performance of the methods in terms of error rates is presented. We can see that with low values of  $d$  (number of features) SVM-NN is able to outperform SVM. In particular, when  $d = 500$ , false negative rate for SVM-NN (**Full Scale**) is significantly lower, and false positive rate is insignificantly lower than for SVM (significance was determined by t-test with  $\alpha = 0.05$ ). With the higher values of  $d$  the advantage of SVM-NN is smaller, but it persists: according to the results of t-test, when  $d = 4000$  SVM-NN (**Full Scale**) is significantly better on false negatives, though insignificantly worse on false positives.

## Conclusion

In this paper we evaluated a learning-based spam filter based on the SVM nearest neighbor classifier and compared it with the SVM classifier and the k-NN classifier. Concerning the performance of the considered method the preliminary evaluation gives promising results. The SVM-NN classifier is shown able to outperform SVM significantly on the small dimensions of the feature space. On the higher dimensions the advantage is less clear; a possible reason for this is higher sensitivity to irrelevant features in comparison to SVM. The disadvantages of the method are comparatively low speed and high resource usage, especially with large values of  $k$ .

## References

- Androutsopoulos, I.; Koutsias, J.; Chandrinou, K.; and Spyropoulos, C. 2000a. An evaluation of naive bayesian anti-spam filtering. In *Proceedings of the Workshop on Machine Learning in the New Information Age, ECML 2000*, 9–17.
- Androutsopoulos, I.; Palouras, G.; Karkaletsis, V.; Sakkis, G.; Spyropoulos, C.; and Stamatopoulos, P. 2000b. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In *Proceedings of the Workshop on Machine Learning and Textual Information Access, PKDD 2000*, 1–13.
- Blanzieri, E., and Bryl, A. 2006. A survey of anti-spam techniques. Technical report #DIT-06-056. Under review.
- Blanzieri, E., and Melgani, F. 2006. An adaptive SVM nearest neighbor classifier for remotely sensed imagery. In *Proceedings of IEEE IGARSS 2006*.
- Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning* 20(3):273–297.
- Drucker, H.; Wu, D.; and Vapnik, V. 1999. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks* 10(5):1048–1054.
- Hulten, G.; Penta, A.; Seshadrinathan, G.; and Mishra, M. 2004. Trends in spam products and methods. In *Proceedings of CEAS'2004*.
- Joachims, T. 1999. *Making large-Scale SVM Learning Practical*. MIT-Press.
- Zhang, L.; Zhu, J.; and Yao, T. 2004. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)* 3(4):243–269.

<sup>2</sup>Available at: <http://svmlight.joachims.org/>