Avoidance of Model Re-Induction in SVM-based Feature Selection for Text Categorization

Aleksander Kołcz
Microsoft Research and Live Labs
ark@microsoft.com

Abdur Chowdhury
Illinois Institute of Technology
abdur@ir.iit.edu

Abstract

Searching the feature space for a subset yielding optimum performance tends to be expensive, especially in applications where the cardinality of the feature space is high (e.g., text categorization). This is particularly true for massive datasets and learning algorithms with worse than linear scaling factors. Linear Support Vector Machines (SVMs) are among the top performers in the text classification domain and often work best with very rich feature representations. Even they however benefit from reducing the number of features, sometimes to a large extent. In this work we propose alternatives to exact re-induction of SVM models during the search for the optimum feature subset. The approximations offer substantial benefits in terms of computational efficiency. We are able to demonstrate that no significant compromises in terms of model quality are made and, moreover, in some cases gains in accuracy can be achieved.

1 Introduction

Linear Support Vector Machines (SVMs) [Vapnik, 1998] have been found to be among the best performers in tasks involving text categorization [Joachims, 1998][Lewis *et al.*, 2004]. Due to the richness of natural language, text categorization problems are characterized by very large numbers of features. Even with infrequent features removed, the dimensionality of the attribute space tends to be very high (e.g., ≈100,000), which for some learners poses computational challenges and/or leads to overfitting the training data. In the case of linear SVMs good performance is in many cases is achieved with little or no feature selection [Joachims, 1998][Lewis *et al.*, 2004][Mladenic *et al.*, 2004], although best performance using all features is by no means guaranteed. It has been shown [Gabrilovich and Markovitch, 2004] that it is relatively easy to identify text classification problems where best accuracy is achieved with aggressive feature selection.

Even if optimum performance is achieved with no selection, the dependence between the classification accuracy and the number of features used often exhibits saturation whereby the improvement in accuracy due to increasing the number of features beyond a certain count is very small. It is therefore important to be able to estimate at which point the performance curve of the classifier measured against the number of most informative features either achieves a maximum or "levels off". Unfortunately, the search for optimum feature settings can be time-consuming due to repetitive model retraining.

In this work we investigate alternatives to SVM model re-induction during feature selection. We are able to demonstrate that the proposed techniques not only result in substantial gains in terms computational efficiency but may actually lead to more accurate solutions and typically lead to equivalent results. As for the feature selection criterion we focus on the feature ranking induced by the SVM itself, since it was found that in the text categorization domain it compares favorably [Mladenic *et al.*, 2004] with the mainstream feature selection criteria such as Information Gain [Rogati and Yang, 2002].

2 Model-driven feature selection for SVMs

A linear SVM creates a classification model by attempting to separate elements of two different classes by a maximum margin [Vapnik, 1998]. For problems that are linearly separable this results in identifying the subsets of both positive and negative examples that lie exactly at the margin — these are called support vectors (SVs). It is quite common, however, that the problem is not linearly separable, in which case the support vector set is enriched by those training examples that cannot be classified by the model correctly (the soft-margin SVM [Cortes and Vapnik, 1995] balances the margin with the total loss over the training data). In either case, the weight vector of the SVM solution (derived via convex optimization) is given by a linear combination of the SVs, and the output of a the SVM to an input vector x is expressed as

$$f(x) = \sum_i w_i x_i + b = \sum_i \left(\sum_j y^j \alpha_j d_i^j \right) x_i + b \qquad (1)$$

where d_i^j is the value corresponding to feature i for the training example j, w_i is the weight assigned to the i-th feature by the SVM and α_j is the Lagrange multiplier associated with d^j (the value of α_j is 0 unless d^j is a support vector - otherwise $\alpha_j > 0$; for soft-margin SVMs the Lagrange multipliers must satisfy $0 < \alpha_j \leq C$); $y^j \in \{-1,+1\}$ is the class label associated with d_i^j.

SVMs have proven to be quite robust in dealing with large numbers of features and overfitting tends to be less of an issue compared to other learners applied in the text classification domain. When feature selection needs to be applied SVMs have been reported to perform well with established feature selection approaches such as IG, χ^2 and BNS [Forman, 2003].

In recent work [Mladenic *et al.*, 2004] investigated the efficacy of using the feature ranking imposed by a trained SVM itself. It was found that ranking according to absolute feature weight values outperforms other established alternatives. Its use is justified [Mladenic *et al.*, 2004] by the fact that the sensitivity of the margin separating the two classes (as determined by a linear SVM) to changes in the j-th feature is directly dependent on the sensitivity of the norm of the weight vector to such changes. This can be expressed as [Mladenic *et al.*, 2004]

$$\sum_{i \in trn\ set} \left| \frac{\delta}{\delta x_i^j} \|w\|^2 \right| \propto |w_i|$$

Therefore, features receiving low absolute weights can be expected to have little effect on the orientation of the optimum hyperplane.

[Mladenic *et al.*, 2004] proposed to apply this criterion in the filter fashion [John *et al.*, 1994], whereby the ranking is obtained just once and subsequent attempts to identify the optimum subset use the top-N features according to the initial ordering. [Guyon *et al.*, 2002] followed an alternative wrapper approach [John *et al.*, 1994] where, after each subset of the least relevant features is removed, the SVM model trained with the remaining ones provides the new ranking, which is then used to identify the next subset of features to remove. As discussed in [Hardin *et al.*, 2004] an SVM may assign low weights to features that are redundant given the presence of other relevant features. The recursive approach of [Guyon *et al.*, 2002] is thus likely to be more successful in compensating for this effect, especially if it is desired to identify a fairly small set of the most relevant features.

In the text domain, where the optimum number of features tends to be large, the differences in quality between models utilizing the recursive and non-recursive approaches are rather small [Li and Yang, 2005][Kalousis *et al.*, 2005]. In this work we focus exclusively on the filter variant of SVM based feature selection proposed in [Mladenic *et al.*, 2004].

3 Avoiding model re-induction in SVM-based feature selection

Traditionally, investigation of the efficacy of different feature subsets relies on inducing a model with each particular choice of the subset and comparing the results achieved. Depending on the type of the learner, repetitive model re-induction may carry a more or less significant cost in terms of the computational time required. In the case of linear SVMs the cost is quadratic in terms of the number of training instances and linear in terms of the number of features used. For massive datasets and rich feature representations, both of which are common in text categorization applications, identifying the optimum can be thus quite expensive.

Published results on SVM-based feature selection indicate that SVM is good in feature ranking in text applications. One interesting question however is: *"how stable is the original solution subject to subsequent feature selection?"*. Put differently, we are interested if the original solution can be significantly reoptimized in the restricted domain of using the top-N features.

For linear SVMs, the quality of the solution is primarily determined by the orientation of the hyperplane normal. Given the direction of this vector, the final position of the hyperplane (controlled by the bias term b in (1)) can be adjusted so as to satisfy a particular utility function (e.g., misclassification cost, precision or recall).

For any feature subset one can obtain a projection of the original hyperplane normal onto the reduced feature space, which is obtained by ignoring or "masking" the coordinates corresponding to the removed features. Our contribution is the address the following questions:

- Is the direction of the projected hyperplane normal substantially different from the direction of the normal induced over the reduced feature representation?
- How does the classification performance of the solution associated with the projected hyperplane normal compare with the solution induced over the reduced feature representation?
- How is the membership of the support vector set affected by reducing the dimensionality of the training data.

3.1 Normal vector feature masking

Let us consider a simple procedure which uses the original SVM model to derive one that operates within the reduced space of top-N features. In this approximation the original model is unchanged, except that only those weight vector components which intersect with the top-N feature subset are retained. A corresponding *feature masking* transformation is applied to the documents. The output of the masked model can be expressed as:

$$f_{mask}(x) = \sum_i m_i w_i x_i + b = \sum_i \left(\sum_{j \in \mathcal{SV}} \alpha_j y^j m_i d_i^j \right) x_i + b$$

where

$$m_i = \begin{cases} 1 \text{ if } i \in \mathcal{M} \\ 0 \text{ if } i \notin \mathcal{M} \end{cases}$$

and $\mathcal{M}$ is the set of features to be masked.

To gain some insight into the properties of masking let us start with the original model and assume that the feature to be masked has the index of N. Given that the original solution is optimal (for a linearly separable problem), it minimizes the Lagrangian

$$L(w,\alpha,b) = \frac{1}{2} w \cdot w - \sum_{j \in trn\ set} \alpha_j y^j \left(w \cdot d^j + b - 1 \right)$$

$$\text{subject to:} y^j \left(w \cdot d^j + b - 1 \right) \geq 0 \text{ for all } j \qquad (2)$$

and thus satisfies the first-order local minimum conditions:

$$\frac{\delta L}{\delta w} = w - \sum_{j \in trn\ set} \alpha_j y^j d^j = 0 \qquad (3)$$

$$\frac{\delta L}{\delta b} = \sum_{j \in trn\ set} \alpha_j y^j = 0$$

Let us now mask out the N-th feature of vector w and each of training vector d^j, while keeping the Lagrange multipliers unchanged, and consider the same optimization problem but in the $N-1$ dimensional space. Notice that the derivatives of the new Lagrangian in (3) with respect to w and b remain 0 and thus the original solution when projected on the lower-dimensional space meets the necessary optimality conditions (with respect to w and b) there as well. To be a true solution in the space in which feature N is ignored it also needs to maximize L with respect to α and meet to constraints (2) however. The constraints in (2) were met originally with equality for the SV set and with strong inequality by all other training vectors. For the sake of an argument let us assume that (2) will hold for points outside the SV set. For the true SVs the constraints will be violated only for those in which feature N was actually present (given the sparsity of text this may be only a small fraction of the SV set). The amount of the violation for each such SV will be

$$y^j \left(w \cdot d^j + b - 1 \right) - y^j \left(w^{-N} \cdot d^j + b - 1 \right) = y^j d_N^j w_N \qquad (4)$$

where it can be seen that a small value of $|w_N|$ leads to a small departure from the optimum (w^{-N} represents the vector with the N-th feature masked out).

Based on the above we can expect that by keeping the values of Lagrange multipliers fixed and masking a low weight feature we can achieve a solution that lies close to the optimum in the reduced-dimensionality space. The validity of such an assumption will increase for features that are infrequent (i.e., inequality constraints for fewer training points will be affected) and for features assigned low absolute weights (i.e., the departure from optimality will likely be small).

3.2 Feature masking and document normalization

In the text domain one often transforms the document feature vectors to unit norm (e.g., according to L2) to compensate for the document length variability [Joachims, 1998][Dumais *et al.*, 1998][Leopold and Kindermann, 2002]. Such transformation introduces feature weighting that is uniform for features within a document but varied across documents. The normal-based feature masking preserves the original feature weighting as the less relevant features are removed, which counters to some extent the original length normalization. An alternative that we consider here is to retain the set of SVs and the associated Lagrange multipliers but renormalize the training vectors after each set of features is removed. The output of such a model is thus given by (assuming L2 length normalization)

$$f_{sv}(x) = \sum_i \left(\sum_j \alpha_j \frac{d_i^j}{\sqrt{\sum_k m_k \cdot d_k^j \cdot d_k^j}} \right) m_i x_i + b$$

Note that such renormalization would typically be applied by default when re-inducing the SVM model using top-N features.

4 Experimental Setup

We used linear SVM induced with all features as the baseline and as the source of feature ranking. Given the original feature set, the methodology was to examine the quality of models using only top N features. To asses the extent and importance of these differences we compared the effectiveness of exact model re-induction and the proposed alternatives over the following document collections:

`TREC-AP`[1]: This dataset represents a collection of AP newswire articles for which the training/test split described in [Lewis *et al.*, 1996] was used. The collection consisted of 142,791 training and 66,992 test documents divided into 20 categories. Each document belonged to one category only.

`Reuters-RCV1`[2]: This collection represents of the most recent and largest corpus used in research involving text categorization. There are $23,149$ training documents and $781,265$ test documents As described in [Lewis *et al.*, 2004] there are several ways of grouping the documents and we restricted ourselves to the *Topic* ontology using the101 topic categories represented in the training portion of the data (out of the 103 categories total). A document may belong to more than one topic and in certain cases there is a hierarchical relationship where a topic may be considered a subtopic of another.

`TechTC-100`[3]: 100 two-class problems (based on the Open Directory Project[4]) generated to purposefully represent different challenges for SVMs as far as feature selection is concerned [Gabrilovich and Markovitch, 2004].

4.1 Document representation

Documents were preprocessed by removing markup and punctuation, converting all characters to lower case and extracting feature tokens as sequences of alphanumerics delimited by whitespace. In the case of the `RCV1` and `TechTC-100` collections, we used the pre-tokenized feature files provided on the corpus websites. In-document term frequency was ignored and each feature received the weight of one if it appeared in a document at least once — otherwise its weight was zero. This binary representation in our experience performs as well as *TF-IDF* when coupled with document-length normalization. More importantly, in the context of this work, with the binary representation the magnitude of perturbations (4) to the conditions (2) depended primarily on the SVM weights and not on independently derived factors, e.g., *TF-IDF*. L2 length normalization was applied since it has been found to be beneficial in text classification [Joachims,

[1] http://www.daviddlewis.com/resources/testcollections/trecap

[2] http://www.daviddlewis.com/resources/testcollections/rcv1/

[3] http://techtc.cs.technion.ac.il/techtc100/techtc100.html

[4] http://www.dmoz.org

1998][Dumais *et al.*, 1998][Leopold and Kindermann, 2002]. Words which occurred just once in the training corpus were eliminated.

4.2 Experimental procedure

The multi-class categorization problems were broken into a series of one-against-the-rest tasks, where each class was treated in turn as the target with the remaining ones playing the role of the anti-target. The TechTC-100 dataset naturally consisted of 100 two-class tasks. Unlike TREC-AP and RCV1 however the two-class problems here were fairly well balanced, with comparable numbers of training and test examples available for the target and the anti-target.

Classification performance was measured in terms of the Area under the Receiver Operating Characteristic (ROC) curve (AUC) and the best attainable value of F1, as tuned via decision-threshold adjustment. Both metrics were estimated over the test data and are reported in terms of the macro-average (for best F1) and micro-average (for AUC) across the categories.

For each two-class task an SVM model was induced using all features, which were then ranked according to their absolute weight values. Features not represented in the SV set were removed and we then considered using just a fraction of the top ranking features (with respect to the non-zero weight ones) with values in $\{0.01, 0.05, 0.1, 0.2, ..., 0.9, 0.99\}$.

In addition to computing the average performance figures for each fraction we also provided the average of the best results on a per category basis, which acknowledges that the optimum number of features according to a given performance metric may change from one two-class problem to another.

In labeling the results the exact reinduction approach is denoted as *exact*, normal-based feature masking is denoted as *mask* and the weight re-computation using the masked and renormalized SV set is labeled as *sv set*.

5 Results

5.1 Accuracy effects

Table 1 one compares the average AUC and best F1 results for the TREC-AP and Reuters-RCV1 collections, where the best feature selection results were determined on a *per category* basis. It can be seen that the best results according these two metrics are numerically very close to each other within each dataset when using the three model feature selection approaches. According to the P-values the differences between the exact and approximate approaches can also be considered statistically insignificant (we used the macro t-test outlined in [Yang and Liu, 1999]), except for the best-F1 measure and the normal-based masking method for RCV1. Over the two collections, best AUC performance was achieved with all or almost all features, but the best F1 performance was more varied as illustrated in Figure 1 for the case of TREC-AP, where it is apparent that reducing the number of features can have a beneficial effect. This exemplifies the fact that optimality of feature selection is dependent on the performance criterion. Note that when a high fraction of top ranking features is retained ($> 50\%$) there is essentially no difference whether an exact or an approximate feature selection method is chosen. For lower feature counts the differences become more pronounced.

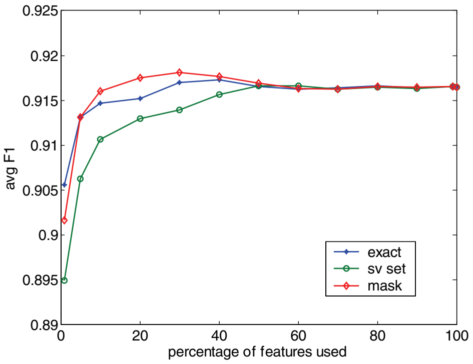

Figure 1: Average best-F1 as a function of top-*N* features for TREC-AP.

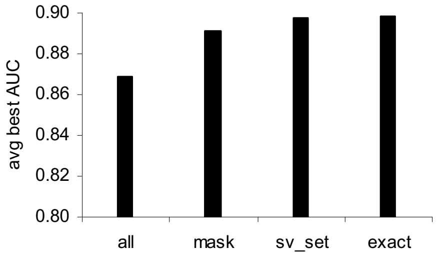

Figure 2: Average-best AUC across the 100 two-class problems for TechTC-100. According to the one-sided paired t-test the difference between the exact and masking approaches is significant (P-value $< 10^{-9}$), while the difference between the exact and SV set approaches is not (P-value= 0.0852).

Since the TechTC-100 collection consisted of problems with balanced numbers of positive and negative examples, in reporting classification accuracy we limited ourselves to the average AUC metric. Following [Gabrilovich and Markovitch, 2004], 4-fold cross-validation was used to estimate accuracy and one-sided paired t-test was applied to estimate significance of the differences in classification performance.

The average best AUC results for using all features and the three feature selection methods are shown in Figure 2. Note that all approaches to optimize the feature set are substantially better than using all features, with the differences being statistically significant (P-values ≈ 0). The P-values show that the SV set approach was statistically equivalent to the exact one, although normal-based masking underperformed in this case. Given that TechTC-100 was specifically designed to illustrate the benefits of feature selection for SVM it is not surprising to see the big gains in AUC shown in Figure 2.

Correlation effects

Aside from measuring the impact of feature selection on the classification performance it is interesting to investigate the

Table 1: Average best AUC and F1 results for the exact and approximate methods for feature selection over the TREC-AP and Reuters-RCV1 collections. P-values for one-sided pairwise t-test (macro) are also given for determining at which point the differences between the two approximate variants and the exact approach can be considered significant.

	best AUC			best F1		
	exact	mask/P-val	sv set/P-val	exact	mask/P-val	sv set/P-val
TREC-AP	0.989499	0.989537/0.21	0.989525/0.23	0.920365	0.92107/0.36	0.919142/0.22
RCV1	0.974551	0.97452/0.117	0.97451/0.126	0.595599	0.594883/0.003	0.595328/0.086

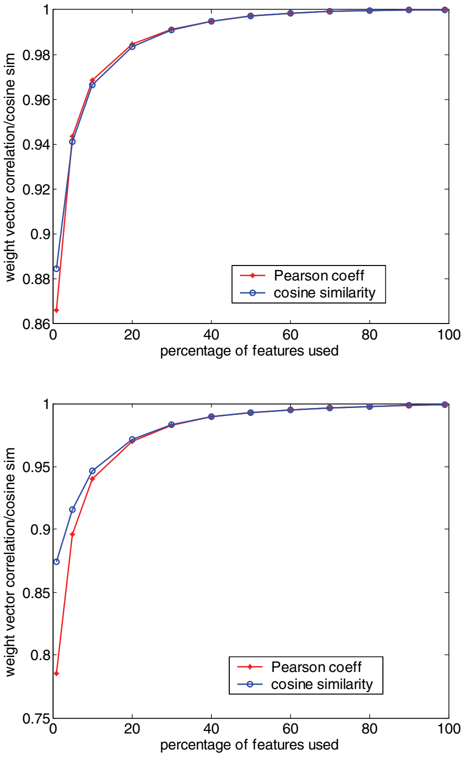

Figure 3: Pearson correlation coefficient and cosine similarity between the masked original weight vector and the weight vector obtained after SVM retraining using the top-N features for TREC-AP (top) and RCV1 (bottom).

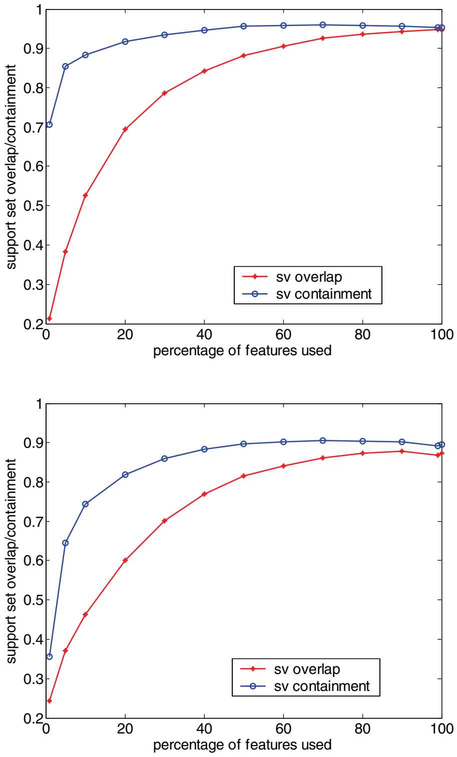

Figure 4: Overlap between the original set of support vectors and the set obtained when training with reduced feature representation for TREC-AP (top) and RCV1 (bottom). The fraction (containment) of SVs corresponding to the original SV set is also shown.

similarity between the weight vectors assigned by the original SVM and the SVM obtained using a reduced feature representation.

Figure 3 shows the dependence of the weight-vector similarity according to the Pearson correlation coefficient and the cosine similarity on the number of top-ranking features used for TREC-AP. and RCV1. For both datasets the weight vectors are very close to each other, even when the fraction of top-N features used is small. The direction of the hyperplane normal vector is thus only very weakly dependent on the less relevant features and by projecting the original weight vector onto the sub-space of the top ranking features one obtains a direction that is oriented very close the optimum.

Figure 4 shows set the average overlap between the original set of SVs and the one obtained when using top-N features for TREC-AP and RCV1. Since the overall number of SVs tends to decrease when fewer features are used, they also show the fraction of SVs corresponding to the original SV set that are part of the SV set obtained using the reduced feature representation. As can be seen, SV set overlap is quite high (exceeds 80%) as long as a sizable fraction of the original features is used (at least 50%). For smaller values of N, the overlap goes down, but this is mainly due to the decrease in the number of SVs in such cases, since the fraction of original SVs used remains consistently very high. The overlap and containment of SV sets do not approach 1 as the fraction of features used approaches 100%. This is because the fraction is defined with respect to the set of features that received non-zero weights in the original SVM run (i.e., after discarding the zero-weight ones). It is thus apparent that the use of features that were deemed irrelevant did in fact have an impact on the original SV set selection.

Given that correlation between masked original weight vectors and weight vectors obtained during reinduction is very high despite somewhat lower levels of overlap between the sets of support vectors, it appears that SVM training significantly alters the original Lagrange multipliers to compensate for the effect of vector length normalization (such normalization will increase the relative contribution of the non-eliminated features) and the use of fewer features. This seems to happen without substantially altering the orientation of the optimum hyperplane.

6 Conclusions

In this work we stipulated that given the stability and robustness of SVMs it is unnecessary to re-induce the SVM models when searching for optimum feature subset settings. We were able to demonstrate experimentally that feature masking produces results that are equivalent to the ones obtained using the traditional model reinduction while being extremely fast compared to SVM retraining. Our experiments showed that both orientation of the normal to the hyperplane and the support vector set remain quite stable as the set of active features is reduced by pruning the least relevant ones. This is particularly true for the orientation of the weight vector, which is the reason why feature masking is so effective and provides evidence that in text categorization the direction of the hyperplane normal will be mostly affected by the most relevant features, with the solution only marginally influenced by including the less relevant ones.

The SVM-based feature selection utilizing feature masking is practically very attractive and much faster (e.g., runtime cost of table-lookup for normal-based masking) than the commonly-used approaches, in which although the original feature ranking is typically quite fast (e.g., using Information Gain), the subsequent estimation of performance when using top-N features requires the more expensive model reinduction. By comparison, without model re-induction the search through the model space to identify the optimum feature count is orders of magnitude faster than the traditional approach and in fact this type of feature selection scales on par with some of the most scalable learners such as Naive Bayes.

References

[Cortes and Vapnik, 1995] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Dumais *et al.*, 1998] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 17th International Conference on Information and Knowledge Management*, pages 148–155, 1998.

[Forman, 2003] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[Gabrilovich and Markovitch, 2004] E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proceedings of The 21st International Conference on Machine Learning*, pages 321–328, 2004.

[Guyon *et al.*, 2002] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[Hardin *et al.*, 2004] D. Hardin, I. Tsamardinos, and C. Aliferis. A theoretical characterization of linear SVM-based feature selection. In *Proceedings of the International Conference on Machine Learning (ICML'04)*, 2004.

[Joachims, 1998] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML)*, 1998.

[John *et al.*, 1994] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994.

[Kalousis *et al.*, 2005] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms. In *Proceedings of the Fith IEEE International Conference on Data Mining (ICDM'05)*, 2005.

[Leopold and Kindermann, 2002] E. Leopold and J. Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46:423–444, 2002.

[Lewis *et al.*, 1996] D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[Lewis *et al.*, 2004] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[Li and Yang, 2005] F. Li and Y. Yang. An analysis of recursive feature elimination methods for statistical classification. In *Proceedings of the 28th Annual International ACM SIGIR Conference (SIGIR-2005)*, 2005.

[Mladenic *et al.*, 2004] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling. Feature selection using linear classifier weights: Interaction with classification models. In *Proceedings of the 27th Annual International ACM SIGIR Conference (SIGIR2004)*, 2004.

[Rogati and Yang, 2002] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the 11th ACM Interantional Conference on Information and Knowledge Management (CIKM)*, 2002.

[Vapnik, 1998] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[Yang and Liu, 1999] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference (SIGIR1999)*, pages 42–49, 1999.