Chapter 22

Information Gain, Correlation and Support Vector Machines

Danny Roobaert, Grigoris Karakoulas, and Nitesh V. Chawla

Customer Behavior Analytics
Retail Risk Management
Canadian Imperial Bank of Commerce (CIBC)
Toronto, Canada
{danny.roobaert,grigoris.karakoulas,nitesh.chawla}@cibc.ca

Summary. We report on our approach, CBAmethod3E, which was submitted to the NIPS 2003 Feature Selection Challenge on Dec. 8, 2003. Our approach consists of combining filtering techniques for variable selection, information gain and feature correlation, with Support Vector Machines for induction. We ranked 13th overall and ranked 6th as a group. It is worth pointing out that our feature selection method was very successful in selecting the second smallest set of features among the top-20 submissions, and in identifying almost all probes in the datasets, resulting in the challenge's best performance on the latter benchmark.

22.1 Introduction

Various machine learning applications, such as our case of financial analytics, are usually overwhelmed with a large number of features. The task of feature selection in these applications is to improve a performance criterion such as accuracy, but often also to minimize the cost associated in producing the features. The NIPS 2003 Feature Selection Challenge offered a great testbed for evaluating feature selection algorithms on datasets with a very large number of features as well as relatively few training examples.

Due to the large number of the features in the competition datasets, we followed the filtering approach to feature selection: selecting features in a single pass first and then applying an inductive algorithm independently. We chose a filtering approach instead of a wrapper one because of the huge computational costs the latter approach would entail for the datasets under study. More specifically, we used information gain (Mitchell, 1997) and analysis of the feature correlation matrix to select features, and applied Support Vector Machines (SVM) (Boser et al., 1992, Cortes and Vapnik, 1995) as the classification algorithm. Our hypothesis was that by combining those filtering techniques with SVM we would be able to prune non-relevant features and

D. Roobaert et al.: *Information Gain, Correlation and Support Vector Machines*, StudFuzz **207**, 463–470 (2006)
www.springerlink.com

learn an SVM classifier that performs at least as good as an SVM classifier learnt on the whole feature set, albeit with a much smaller feature set. The overall method is described in Section 2. Section 3 presents the results and provides empirical evidence for the above hypothesis. Section 4 refers to a few of the alternative techniques for feature selection and induction that we tried. Section 5 concludes the paper with a discussion on lessons learned and future work.

22.2 Description of Approach

We first describe the performance criterion that we aimed to optimize while searching for the best feature subset or parameter tuning in SVM induction. We then present the two filtering techniques for feature selection and briefly describe the specifics of our SVM approach. We report on the approach submitted on Dec. 8. For the Dec. 1 submission, we used an alternative approach (see Section 4) that was abandoned after the Dec. 1 submission because we obtained better performance with the approach described in the following.

22.2.1 Optimization Criterion

For choosing among several algorithms and a range of hyper-parameter settings, the following optimization criterion was followed: Balanced error rate (BER) using random ten-fold cross-validation before Dec. 1 (when the validation labels were not available), and BER on the validation set after Dec. 1 (when the validation labels were available). BER was calculated in the same way as used by the challenge organizers: $BER = \frac{1}{2}\left[\frac{fp}{tn+fp} + \frac{fn}{tp+fn}\right]$, with fp = false positives, tn = true negatives, fn = false negatives and tp = true positives.

22.2.2 Feature Selection

Information Gain

Information gain (IG) measures the amount of information in bits about the class prediction, if the only information available is the presence of a feature and the corresponding class distribution. Concretely, it measures the expected reduction in entropy (uncertainty associated with a random feature) (Mitchell, 1997). Given S_X the set of training examples, $\mathbf{x}_i$ the vector of i^{th} variables in this set, $|S_{\mathbf{x}_i=v}|/|S_X|$ the fraction of examples of the i^{th} variable having value v:

$$IG(S_X, \mathbf{x}_i) = H(S_X) - \sum_{v=values(\mathbf{x}_i)}^{\frac{|S_{\mathbf{x_i}=v}|}{|S_X|}} H(S_{\mathbf{x}_i=v}) \text{ with entropy:}$$

$$H(S) = -p_+(S)\log_2 p_+(S) - p_-(S)\log_2 p_-(S)$$

$p_\pm(S)$ is the probability of a training example in the set S to be of the positive/negative class. We discretized continuous features using information theoretic binning (Fayyad and Irani, 1993).

For each dataset we selected the subset of features with non-zero information gain. We used this filtering technique on all datasets, except the MADELON dataset. For the latter dataset we used a filtering technique based on feature correlation, defined in the next subsection.

Correlation

The feature selection algorithm used on the MADELON dataset starts from the correlation matrix M of the dataset's variables. There are 500 features in this dataset, and we treat the target (class) variable as the 501^{st} variable, such that we measure not only feature redundancy (intra-feature correlation), but also feature relevancy (feature-class correlation). In order to combine redundancy & relevancy information into a single measure, we consider the column-wise (or equivalently row-wise) average absolute correlation $\langle M \rangle_i = \frac{1}{n}\sum_j |M_{ij}|$ and the global average absolute correlation $\langle M \rangle = \frac{1}{n^2}\sum_{i,j} |M_{ij}|$. Plotting the number of column correlations that exceeds a multiple of the global average correlation ($\langle M \rangle_i > t \langle M \rangle$) at different thresholds t, yields Figure 22.1.

As can be observed from Figure 22.1, there is a discontinuity in correlation when varying the threshold t. Most variables have a low correlation, not exceeding about 5 times average correlation. In contrast, there are 20 features that have a high correlation with other features, exceeding 33 times the average correlation. We took these 20 features as input to our model.

The same correlation analysis was performed on the other datasets. However no such distinct discontinuity could be found (i.e. no particular correlation structure could be discovered) and hence we relied on information gain to select variables for those datasets. Note that Information Gain produced 13 features on the MADELON dataset, but the optimization criterion indicated worse generalization performance, and consequently the information gain approach was not pursued on this dataset.

22.2.3 Induction

As induction algorithm, we choose Support Vector Machines (Boser et al., 1992, Cortes and Vapnik, 1995). We used the implementation by Chang and Lin (2001) called LIBSVM. It implements an SVM based on quadratic optimization and an epsilon-insensitive linear loss function. This translates to the following optimization problem in dual variables α:

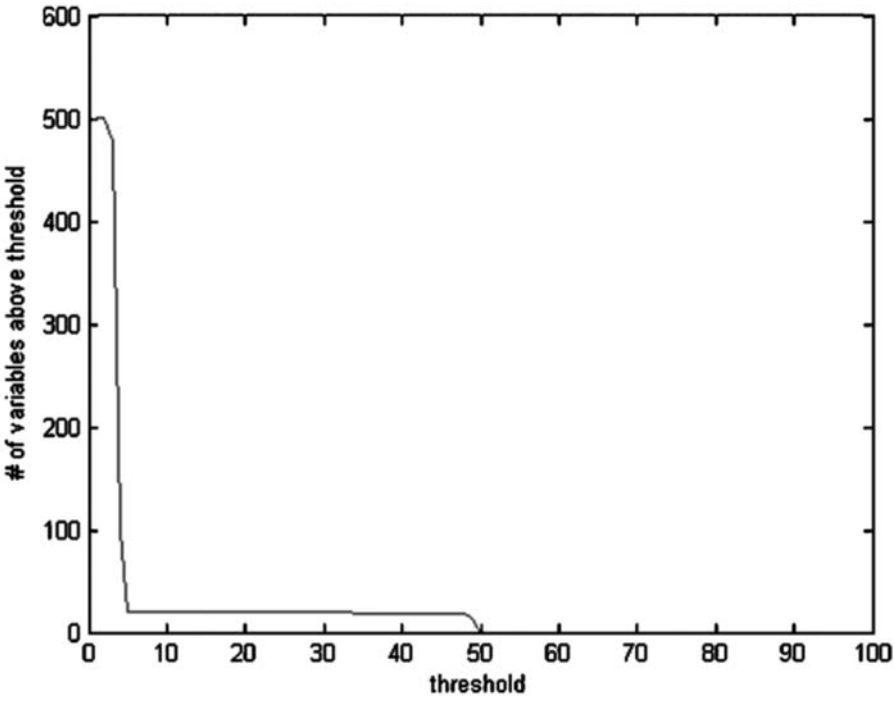

Fig. 22.1. The number of MADELON variables having a column correlation above the threshold

$$\max_{\alpha} \left(\sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k=1}^{m} \sum_{l=1}^{m} \alpha_k \alpha_l y_k y_l K(\mathbf{x}_k, \mathbf{x}_l) \right) \text{ s.t. } \begin{cases} 0 \leq \alpha_k \leq C, \forall k \\ \sum_{k=1}^{m} y_k \alpha_k = 0 \end{cases}$$

where C is the regularization hyper-parameter, $K(\mathbf{x}_k, \mathbf{x}_l)$ the kernel and $y_k$the target (class) variables. The implementation uses Sequential Minimal Optimization (Platt, 1999) and enhanced heuristics to solve the optimization problem in a fast way. As SVM kernel we used a linear kernel $K(\mathbf{x}_k, \mathbf{x}_l) = \mathbf{x}_k \cdot \mathbf{x}_l$ for all datasets, except for the Madelon dataset where we used an RBF-kernel $K(\mathbf{x}_k, \mathbf{x}_l) = e^{-\gamma \|\mathbf{x}_k - \mathbf{x}_l\|}$. The latter choices were made due to better optimization criterion results in our experiments.

For SVM hyper-parameter optimization (regularization hyper-parameters C and γ in the case of an RBF kernel), we used pattern search (Momma and Bennett, 2002). This technique performs iterative hyper-parameter optimization. Given an initial hyper-parameter setting, upon each iteration, the technique tries a few variant settings (in a certain pattern) of the current hyper-parameter settings and chooses the setting that best improves the performance criterion. If the criterion is not improved, the pattern is applied on a finer scale. If a pre-determined scale is reached, optimization stops.

For the imbalanced dataset DOROTHEA, we applied asymmetrically weighted regularization values C for the positive and the negative class. We used the following heuristic: the C of the minority class was always kept at a factor, $|majorityclass|/|minorityclass|$, higher than the C of the majority class.

22.3 Final Results

Our submission results are shown in Table 22.1. From a performance point of view, we have a performance that is not significantly different from the winner (using McNemar and 5% risk), on two datasets: ARCENE and MADELON. On average, we rank 13th considering individual submissions, and 6^th as a group.

Table 22.1. NIPS 2003 challenge results on December 8^{th}

Dec. 8^{th}	Our best challenge entry[1]					The winning challenge entry					
Dataset	Score	BER	AUC	Feat	Probe	Score	BER	AUC	Feat	Probe	Test
OVERALL	21.14	8.14	96.62	12.78	0.06	88.00	6.84	97.22	80.3	47.8	0.4
ARCENE	85.71	11.12	94.89	28.25	0.28	94.29	11.86	95.47	10.7	1.0	0
DEXTER	0	6.00	98.47	0.60	0.0	100.00	3.30	96.70	18.6	42.1	1
DOROTHEA	-28.57	15.26	92.34	0.57	0.0	97.14	8.61	95.92	100.0	50.0	1
GISETTE	-2.86	1.60	99.85	30.46	0.0	97.14	1.35	98.71	18.3	0.0	0
MADELON	51.43	8.14	96.62	12.78	0.0	94.29	7.11	96.95	1.6	0.0	1

From a feature selection point of view, we rank 2nd (within the 20 best submission) in minimizing the number of used features, using only 12.78% on average. However we are consistently 1st in identifying probes: on this benchmark, we are the best performer on all datasets.

To show the significance of feature selection in our results, we ran experiments where we ignored the feature selection process altogether, and applied SVMs directly on all features. In Table 22.2, we report the best BER on the validation set of each dataset. These results were obtained using linear SVMs, as in all experiments RBF-kernel SVMs using all features gave worse results compared to linear SVMs. As can be seen from the table, using all features always gave worse performance on the validation set, and hence feature selection was always used.

[1] Performance is not statistically different from the winner, using McNemar and 5% risk.

Table 22.2. BER performance on the validation set, using all features versus the described selected features

Dataset	All features	Selected features	Reduction in BER
Arcene	0.1575	0.1347	-16.87%
Dexter	0.0867	0.0700	-23.81%
Dorothea	0.3398	0.1156	-193.96%
Gisette	0.0200	0.0180	-11.11%
Madelon	0.4000	0.0700	-471.43%

22.4 Alternative Approaches Pursued

Several other approaches were pursued. All these approaches though gave worse performance (given the optimization criterion) and hence were not used in the final submission. We briefly discuss a few of these approaches, as we are restricted by paper size.

22.4.1 Alternative Feature Selection

We used a linear SVM to remove features. The approach is as follows: we first train a linear SVM (including hyper-parameter optimization) on the full feature set. Then we retain only the features that correspond with the largest weights in the linear function. Finally, we train the final SVM model using these selected features. We experimented with different feature fractions retained, as in general the approach does not specify how to choose the number of features to be retained (or the weight threshold). In Table 22.3, we show a performance comparison at half, the same and double of the size of the feature set finally submitted. We did not try a variant of the above approach called Recursive Feature Elimination (RFE), proposed by (Guyon et al., 2002) due to its prohibitive computational cost.

Table 22.3. BER performance on the validation set, comparing feature selected by LINSVM versus Infogain/Corr

Dataset	Feature	LIN SVM feature fraction			InfoGain /
	Final fraction	Half final	Final	Double final	Corr. feature
Arcene	0.2825	0.1802	0.1843	0.1664	0.1347
Dexter	0.0059	0.1000	0.1167	0.1200	0.0700
Dorothea	0.0057	0.2726	0.3267	0.3283	0.1156
Gisette	0.3046	0.0260	0.0310	0.0250	0.0180
Madelon	0.0400	0.1133	0.1651	0.2800	0.0700

22.4.2 Combining Feature Selection and Induction

We tried also a linear programming approach to SVM inspired by Bradley and Mangasarian (1998). Here SVM is formulated as a linear optimization problem instead of the typical SVM quadratic optimization. The resulting model only uses a selected number of non-zero weights and hence feature selection is embedded in the induction. Unfortunately the results were not encouraging.

22.5 Discussion and Conclusion

We showed how combining a filtering technique for feature selection with SVM leads to substantial improvement in generalization performance of the SVM models in the five classification datasets of the competition. The improvement is the highest for the datasets Madelon and Dorothea as shown in table 2 above. These results provide evidence that feature selection can help generalization performance of SVMs.

Another lesson learned from our submission is that there is no single best feature selection technique across all five datasets. We experimented with different feature selection techniques and picked the best one for each dataset. Of course, an open question still remains: why exactly these techniques worked well together with Support Vector Machines. A theoretical foundation for the latter is an interesting topic for future work.

Finally, it is worth pointing out that several of the top-20 submissions in the competition relied on using large feature sets for each dataset. This is partly due to the fact that the performance measure for evaluating the results, BER, is a classification performance measure that does not penalize for the number of features used. In most real-world applications (e.g. medical and engineering diagnosis, credit scoring etc.) there is a cost for observing the value of a feature. Hence, in tasks where feature selection is important, such as in this challenge, there is need for a performance measure that can reflect the trade-off of feature and misclassification cost (Turney, 2000, Karakoulas, 1995). In absence of such a measure, our selection of approaches was influenced by this bias. This resulted in the second smallest feature set in the top-20 and the most successful removal of probes in the challenge.

Acknowledgements

Our thanks to Andrew Brown for the Information Gain code and Brian Chambers and Ruslan Salakhutdinov for a helpful hand.

References

B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

P.S. Bradley and O.L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proc1̇5th IntĊonfMachine Learning*, pages 82–90, 1998.

C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

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

U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc1̇0th IntĊonfMachine Learning*, pages 194–201, 1993.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

G. Karakoulas. Cost-effective classification for credit scoring. In *Proc3̇rd IntĊonfÁI Applications on Wall Street*, 1995.

T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

M. Momma and K.P. Bennett. A pattern search method for model selection of support vector regression. In R. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *Proceedings of the Second SIAM International Conference on Data Mining*, pages 261–274. SIAM, 2002.

J. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*, chapter 12, pages 185–208. MIT Press, 1999.

P. Turney. Types of cost in inductive concept learning. In *Workshop cost-sensitive learning, Proc. 17th Int. Conf. Machine Learning*, pages 15–21, 2000.