Combining feature selection and feature reduction for protein classification

Ricco Rakotomalala
ERIC - University of Lyon 2
F-69676 Bron
France

Faouzi Mhamdi
URPAH - University of Tunis
Tunis
Tunisia

Abstract: We use the n-grams descriptors for a protein classification task. As they are automatically generated, we obtain many irrelevant and/or redundant descriptors. In this paper, we evaluate various strategies of feature selection and feature reduction. First, we evaluate separately the efficiency of a filtering feature selection algorithm and a feature reduction on the basis of a singular value decomposition process (SVD). Then, we evaluate the combination of the two approaches i.e. we propose to use a very tolerant filter to select on a univariate basis which attributes to include in the subsequent SVD. We expect that the features extracted from relevant descriptors should allow to build a better classifier. We experiment the various approaches on two non-linear classifiers: a 3-nearest neighbor which is very sensitive to high dimensionality, and a SVM with a RBF kernel function which is well regularized. The results show that the behavior of the approaches depends mainly on the supervised learning characteristics.

Key–Words: Bioinformatics, Supervised learning, Feature Extraction, Feature Selection, Singular Value Decomposition

1 Introduction

Data preprocessing is a crucial step when we have to analyze an unstructured dataset. Indeed, it is not possible to handle directly the native description of data to run a machine learning algorithm when we treat images, text, or in our case, when we want to predict proteins families from their primary structure. The learning process is thus preceded by two data preprocessing operations: extract descriptors from the native format of data in order to build an attribute value table; build features from these descriptors in order to produce an efficient classifier[1].

The direct use of all descriptors extracted from the unstructured representation as features for the learning algorithm is in general not a good strategy. Their number is very high, which induces drawbacks: the computing time is very high and the quality of the learning classifier is often poor because we have a scattered dataset, it is difficult to estimate in a reliable way the probability distribution (”The curse of Dimensionality Problem”). In a protein discrimination process from their primary structures [1], the native description of a protein is a succession of characters representing amino acids. It is not possible to run directly a learning algorithm. We then generated a Boolean attribute-value table by checking the presence or absence of 3-grams (a sequence of 3 consecutive characters) for each protein. Because there are 20 kinds of characters (amino acids), we can produce 8000 descriptors for 100 examples, the quality of the classifier on all descriptors is often bad.

To solve these disadvantages, we are interested in the creation of intermediate features from the descriptors. The first strategy is to perform a feature selection. We can mainly use a filter approach because we have many descriptors, the wrapper algorithms seems inappropriate [2]. The second strategy is feature reduction. The goal is to produce a reduced representation space which preserves the properties of the initial space, in particular by preserving the proximity between the examples. A few new features will be provided to the learning algorithm. The singular value decomposition (SVD) seems to answer in an adequate way these specifications. Indeed it aims to transform raw data to a new co-ordinate system, where the axes of the new space represent “factors” or “latent variables” which reveal the underlying structure of the dataset. This approach, very popular in high dimensional data processing, presents nevertheless a draw-

[1] In this paper, we call ”descriptors” the attributes which are extracted from the native data format, i.e., n-grams in our context; we call ”features” the attributes which are presented to the supervised learning algorithm.

back in the context of supervised learning: a lot of initial descriptors are irrelevant for the supervised learning task. To take them into account in the construction of features (factors) considerably reduced the relevance of these features.

We propose to combine these strategies in the protein discrimination process. We insert a phase of descriptor selection before building the latent variables with the singular value decomposition. This phase of selection must only take account of the relevance of the descriptors and not of their redundancy, it must be rather permissive so that information necessary to discriminate is preserved. Only the selected descriptors will then be presented to the SVD, thus making it possible to produce an effective reduced space of representation for discrimination. In our protein discrimination context, the results show that it is sufficient to keep a few factors. Another advantage, although that was not our first goal in this work, is that the reduction of the number of descriptors presented to the SVD algorithm allows one to reduce dramatically the computing time.

In order to evaluate these various data preprocessing strategies, we organize the experiments as follows. In the first time, we deeply study these strategies with a 3-NN (Nearest Neighbor) classifier. It is a non-linear classifier with a weak regularization, it is very sensitive to high dimensionality. In the second time, we compare these results with the results of a Support Vector Machine classifier with a Radial Basis kernel function (SVM-RBF). It is also a non-linear classifier but it has a strong regularization property, it is known to be robust against dimensionality problem [3].

Section 2 introduces the SVD process and our improvement in the context of supervised learning in high dimensional dataset. The protein discrimination problem and results of experiments are presented in Section 3. Section 4 describes some further experiments which allows us to better evaluate the behavior of our approach. We conclude in Section 5.

2 Combining feature selection and singular value decomposition

2.1 The singular value decomposition process

SVD produces a new representation space of the observations starting from the initial descriptors by preserving the proximity between the examples. These new features known as ”factors” or ”latent variables” have several very advantageous properties: (a) their interpretation very often allows to detect patterns in the initial space; (b) a very reduced number of factors allows to restore information contained in the data; (c) the new features form an orthogonal basis, learning algorithms such as linear discriminant analysis work well [4]. This process is often used in microarray data analysis [5] or text retrieval [3], fields where the initial number of descriptors is very high and where the dimensionality reduction is crucial before data analysis.

There are numerous theoretical presentations of the SVD. Roughly speaking, we produce from an initial description space $\aleph = \{X_1, \ldots, X_J\}$ of J descriptors (and n examples), a new space of J features $\Psi = \{F_1, \ldots, F_J\}$ with the following constraints: Ψ is an orthogonal basis; the factor F_1 is built from a projection vector P_1 ($\|P_1\| = 1$) so as to maximize the variance of F_1, $v_1 = Var(F_1)$; the second factor F_2 is built from a projection vector P_2 ($\|P_2\| = 1$) so as to maximize the variance $v_2 = Var(F_2)$, and F_2 must be independent (perpendicular) to F_1, etc. In the two spaces, the proximity between two individuals is preserved, and more interesting, in the subspace p ($p < J$) of Ψ, the distance between two examples is roughly respected, the quality of the approximation can be measured using the sum of variance of p first selected factors ($S_p = \sum_{j=1}^{p} v_j$).

There is a mathematical relation between SVD and PCA (Principal Component Analysis) when the descriptors are standardized. If $\aleph'$ is the transpose of $\aleph$, the square matrix $(\aleph'\aleph)$ is a correlation matrix: v_1 is its first eigenvalue and P_1 is the associated eigenvector. Thus, the sum of variance of the first p selected factors is the proportion of explained variance with these factors ($E_p = \frac{S_p}{J}$).

In addition to the dimensionality reduction which improves the efficiency of the supervised learning algorithm, this process allows to detect and extract the true patterns in the data, the last factors express the noisy information in the dataset. From this point of view, the SVD is an effective data cleaning process, by selecting the p best factors, we reject negligible information contained in the data. Thus, it is possible to reconstruct an approximate version of original data from the selected factors and projection vectors.

About the implementation, the challenge was considerable. It was not possible to use diagonalization techniques from the 8000×8000 correlation matrix in order to extract eigenvalue and eigenvectors. It was thus necessary to consider the direct extraction of the singular values from the standardized matrix $\aleph$ with a powerful algorithm, the computing time and the memory requirement are major constraints. We used the NIPALS implementation [6] which interprets the singular value extraction as successive orthogonal regressions: the first one produces the first factor F_1, using the residuals of this regression, we perform a new regression in order to produce the second factor F_2, etc. This approach allows to reduce computations considerably since we can stop calculations as soon as the first p factors were generated. In our experiments, from a $n = 100$ examples and $J = 7000$ descriptors, the first 15 factors are generated in 10 seconds on a standard personal computer running under Windows (Pentium IV – 3.8 Ghz – 2 GB RAM). We use the TANAGRA [7], an open source data mining software, source code is available on the website of the authors (http:$\backslash\backslash eric.univ - lyon2.fr\backslash\tilde{}ricco\backslash tanagra)$.

2.2 SVD and irrelevant descriptors

If the SVD is a very interesting process for dimensionality reduction by controlling the loss of information, it has a major drawback in a protein classification framework: the SVD is an unsupervised process. In fact, to build the factors, it used all the descriptors, including the irrelevant one for a supervised learning task.

To illustrate this drawback, we show the same situation on an artificial two-dimensional dataset (Figure 1). On the unlabeled dataset (Figure 1.a), the first extracted factor F_1 seems appropriate, but on the labeled dataset (Figure 1.b), we see that the descriptor X_2 is irrelevant for the learning task, however the SVD extracts the same factor F_1.

2.3 Feature selection and SVD

In this paper, we propose to perform first a descriptor selection before building the factors with SVD. We call this combination FS-SVD (Feature Selection - Singular Value Decomposition). The goal of the selection is not to produce the most powerful subspace for the prediction like in classical feature selection process [2] but rather to eliminate the irrelevant descriptors before the SVD process. In this point of view, we use a very simple filter algorithm: we rank the descriptors according to the correlation coefficient criterion and keep the 150 best for SVD (the correlation coefficient computed on 0/1 attribute is similar to χ^2 criterion on Boolean true/false attribute) [8]. Of course, some selected descriptors are redundant but it is does not matter because the features obtained with the SVD are orthogonal.

We propose the following general framework for protein classification:

1. Extract descriptors from native format of proteins sequences;
2. Select the 150 most correlated descriptors with the class-attribute (protein family);
3. Compute the 15 first features with NIPALS;
4. Select the most significant one among the extracted features. They can be individually evaluated because they are orthogonal. We use a ANOVA (Analysis of Variance) test with a p-level of 0.01. This process allow to avoid irrelevant features which corresponds to artifacts of the learning set;
5. Use features in a supervised learning algorithm.

From this general framework, we can define various strategies:

- **ALL** is a combination of (1) and (5), we use all descriptors as features for the learning algorithm;
- **FS** is a combination of (1), (2) and (5), we use the selected descriptors as features for the learning algorithm;
- **SVD** is a combination of (1), (3), (4) and (5), we extract factors from all descriptors and use them as features for the learning algorithm;
- **FS-SVD** is a combination of (1), (2), (3), (4) and (5), we extract factors from selected descriptors and use them as features for the learning algorithm.

The chosen parameters (150 correlated descriptors and 15 features) are defined in an approximate way and are appropriate for all situations that we treated. We preferred to make a simplified choice and avoid fine tuning parameters which is always problem dependent and a source of overfitting, especially when we use a cross validation error rate estimate.

3 Experiments on a protein classification problem

3.1 The protein classification problem

In this paper, we use the text mining framework for a protein classification problem from their primary structures. The analogy with text classification is relevant in our case, indeed the original description of the dataset is very similar. A protein is described by a series of characters which represents amino acids. There are 20 possible amino acids. We show an example of a file describing a few proteins (Figure 2).

However, unlike the text classification, there is no ”natural” separation in the character sequences, it is not possible to extract ”words” for which we can easily attach semantics properties. Therefore, we have used the n-grams, a sequence of n consecutive characters, in order to produce descriptors.

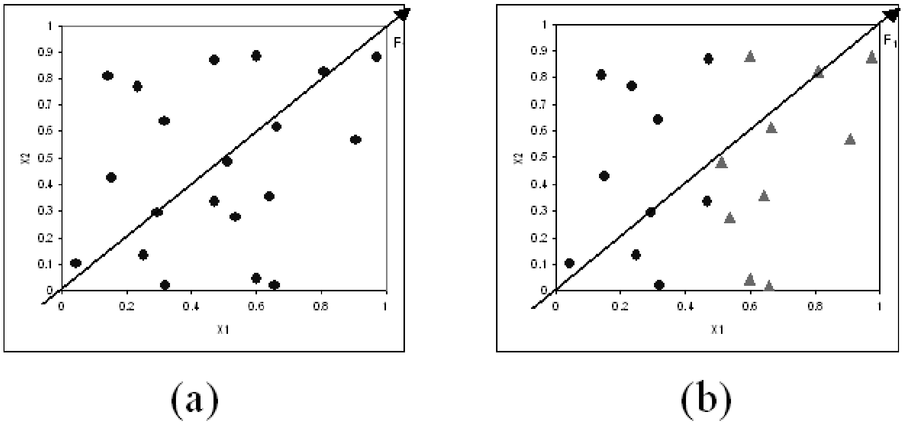
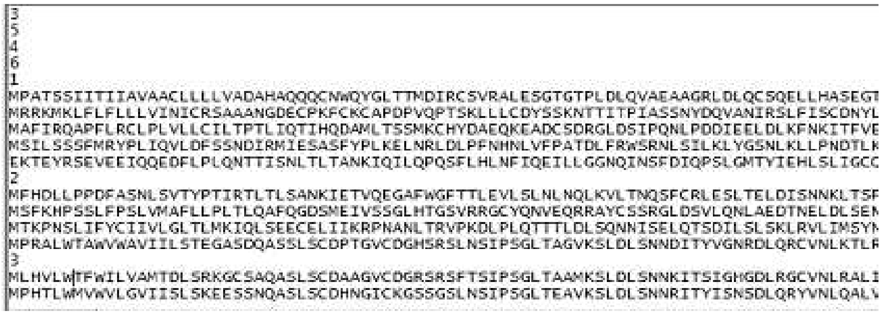


Figure 1: SVD on unsupervised (a) and supervised (b) tasks



```
3
5
4
6
1
MPATSSIITIIAVAACLLLLVADAHAQQQCNWQYGLTTMDIRCSVRALESGTGTPLDLQVAEAAGRLDLQCSQELLHASEGT
MRRKMKLFLFLLLVINICRSAAANGDECPKFCKCAPDPVQPTSKLLLCDYSSKNTTITPIASSNYDQVANIRSLFISCDNYL
MAFIRQAPFLRCLPLVLLCILTPTLIQTIHQDAMLTSSMKCHYDAEQKEADCSDRGLDSIPQNLPDDIEELDLKFNKITFVE
MSILSSSFMRYPLIQVLDFSSNDIRMIESASFYPLKELNRLDLPFNHNLVFPATDLFRWSRNLSILKLYGSNLKLLPNDTLK
EKTEYRSEVEEIQQEDFLPLQNTTISNLTLTANKIQILQPQSFLHLNFIQEILLGGNQINSFDIQPSLGMTYIEHLSLIGCC
2
MFHDLLPPDFASNLSVTYPTIRTLTLSANKIETVQEGAFWGFTTLEVLSLNLNQLKVLTNQSFCRLESLTELDISNNKLTSF
MSFKHPSSLFPSLVMAFLLPLTLQAFQGDSMEIVSSGLHTGSVRRGCYQNVEQRRAYCSSRGLDSVLQNLAEDTNELDLSEN
MTKPNSLIFYCIIVLGLTLMKIQLSEECELIIKRPNANLTRVPKDLPLQTTTLDLSQNNISELQTSDILSLSKLRVLIMSYN
MPRALWTAWVWAVIILSTEGASDQASSLSCDPTGVCDGHSRSLNSIPSGLTAGVKSLDLSNNDITYVGNRDLQRCVNLKTLR
3
MLHVLWTFWILVAMTDLSRKGCSAQASLSCDAAGVCDGRSRSFTSIPSGLTAAMKSLDLSNNKITSIGHGDLRGCVNLRALI
MPHTLWMVWVLGVIISLSKEESSNQASLSCDHNGICKGSSGSLNSIPSGLTEAVKSLDLSNNRITYISNSDLQRYVNLQALV
```

Figure 2: Native description of proteins

Previous works showed that the choice of $n = 3$ (3-grams) and boolean descriptors give a good compromise to produce accurate classifier [1]. We obtain a Boolean attribute - value dataset with several thousands of descriptors (Figure 3). The theoretical maximum number of 3-grams for a protein classification problem is $20^3 = 8000$. Of course, all 3-grams are not present in a dataset but experiments showed that we were close to this value. Many 3-grams are irrelevant, others also are redundant. The main challenge of the feature reduction is to build appropriate features for a supervised learning task. There are several reasons for this dimensionality reduction: (1) machine learning algorithms work badly when the dataset is too scattered; selecting a subset of relevant features often improves the classifier performance; (2) the complexity of the learning algorithms always depends on the number of input features; the elimination of useless attributes allows to a considerable improvement in computing time; (3) a reduced number of features provides a better understanding of the classifier.

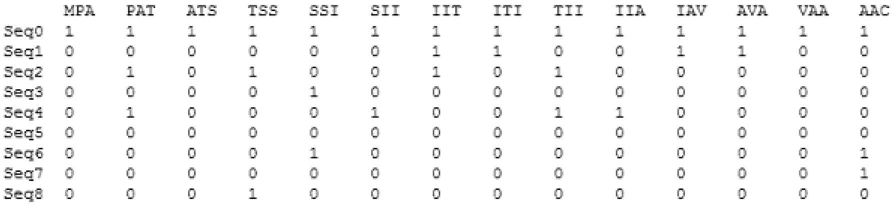
3.2 Learning methods characteristics

Because it is not feasible to test every possible types of classifier, we must restrict our hypothesis space in machine learning. Two types of biases are commonly used: the hypothesis space bias, the representation bias, that restricts the type of function used for prediction; and the preference bias that enables to choose the best one among the solutions.

About the representation bias, we distinguish mainly linear and non-linear classifier. *A priori*, a non-linear classifier is always preferable, it can also describe a linear concept. But, in practice, the "true" underlying concept between the target and the input attributes is unknown, perhaps it does not exist. The used function is a choice and, in this point of view, a linear representation impose a stronger restriction.

About the preference bias, we want especially put forward resistance to overfitting. Especially in cases where the dimensionality is high in relation to the dataset size, the learning algorithm may adjust to specific informations of the dataset. The generalization performance is bad. Support Vector Machines (SVM) for instance, with the maximum-margin principle, enables to produce very stable classifiers [9]. The dependency to the learning set is know also as the variance of the classifier.

Starting from these two criteria, we can position the methods used in our experimentation. We choose two non-linear classifier. The nearest-neighbor (3-



	MPA	PAT	ATS	TSS	SSI	SII	IIT	ITI	TII	IIA	IAV	AVA	VAA	AAC
Seq0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Seq1	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Seq2	0	1	0	1	0	0	1	0	1	0	0	0	0	0
Seq3	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Seq4	0	1	0	0	0	1	0	0	1	1	0	0	0	0
Seq5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Seq6	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Seq7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Seq8	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Figure 3: Boolean 3-grams attribute-value table from native description

NN) classifier allow to represent any type of concept between the target and the input features, but it is very sensitive to high dimensional feature space and has a high variance [4]. SVM with a RBF kernel function is also a non-linear classifier. It transforms the input space in an infinite dimension. But the maximum margin principle allows to produce stable classifier, it is well regularized, so the infinite dimension would not have spoil the results. We use the LIBSVM library in our experiments [10].

We will check these behaviors on our dataset. In the protein discrimination context where we have about 80 times more descriptors than example, we can expect that feature selection and regularization will be more efficient for learning classifier with a high variance.

3.3 Results of experiments for 3-NN classifier

Five protein families have been randomly extracted from the data bank SCOP [11], the aim being to discriminate each pair of proteins. We compare the performance on each task by using a 5×2-cross validation error rate estimate. Like these families are randomly extracted, we can consider that the result of each experiment is an estimation of the performance of the protein discrimination task. Thus we can also compute and compare the average performance of each data preprocessing strategy.

The results suggest some interesting comments (Table 1):

- Running the learning algorithm on all descriptors (ALL) is an inefficient approach, especially for 3-NN classifier. The high dimensionality deteriorates the results, because there are many irrelevant descriptors, but also because the nearest neighbor works poorly when we have a scattered dataset.
- FS and SVD improve the performances, but in a different way. FS removes irrelevant descriptors, perhaps there are again too descriptors, we discuss later the optimization of the number of descriptors, it is clear that 150 descriptors for about 100 examples is harmful to 3-NN. SVD reduces dramatically the number of features, the regularization is efficient. In our experiment, only a few factors are significant and used for the classification task. But too many irrelevant descriptors influence the construction of these factors.
- FS-SVD, descriptor selection before building the factors is an efficient way to improve the 3-NN classifier performance. Whatever the pair of families to discriminate, FS-SVD outperforms both FS and SVD. Removing irrelevant descriptors helps the singular value decomposition technique to build more relevant features (factors) for the learning algorithm.

Table 1: Estimated error rate on each protein discrimination problem using 3-NN classifier

Proteins pairs	**ALL**	**FS**	**SVD**	**FS-SVD**
F_{12}	0.365	0.188	0.060	**0.028**
F_{13}	0.323	0.162	0.185	**0.134**
F_{14}	0.308	0.067	0.095	**0.047**
F_{15}	0.361	0.106	0.133	**0.044**
F_{23}	0.240	0.088	0.140	**0.032**
F_{24}	0.131	0.094	0.055	**0.009**
F_{25}	0.189	0.128	0.081	**0.025**
F_{34}	0.287	0.127	0.115	**0.046**
F_{35}	0.190	0.092	0.187	**0.070**
F_{45}	0.180	0.069	0.074	**0.043**
$Average$	0.258	0.112	0.113	**0.048**

Even if that were not our first goal in this work, it nevertheless were interesting to compare the computing times between the two approaches (SVD and FS-SVD): we noted that, on average, the descriptor selection allows to reduce 15 times the execution time of the protein classification problems.

Table 2: Comparison of the average error rate between 3-NN and SVM-RBF classifiers

Proteins pairs	**ALL**	**FS**	**SVD**	**FS-SVD**
SVM-RBF	0.099	**0.045**	0.102	0.062
3-NN	0.258	0.112	0.113	**0.048**

3.4 Comparison with SVM-RBF

SVM with the maximum margin principle produces a stable classifier. By using a RBF kernel function, we have a non-linear classifier. So we obtain a powerful classifier with a low variance, it is very interesting to study the behavior of this type of classifier in relation to our data preprocessing strategy. The comparison of the average error rate of 3-NN is reported in the table 2, we obtain contrasted results.

- Using all descriptors allows to produce a classifier with a moderate error rate. Even if SVM is well regularized, our feature selection, which is very tolerant, improves significantly the performances of SVM-RBF. We see later that fine tuning the number of selected descriptors does not significantly improve the error rate.
- At the opposite, the regularization with the singular value decomposition is not powerful for SVM. Factors, which are linear transformations of input descriptors, deteriorates the behavior of SVM. Even if we have seen in the previous section that they are relevant for the classification task. SVD and maximum margin regularization are not compatible.

Last but not the least, we note that "FS-SVD + 3-NN" performs as well as SVM with a RBF kernel function ("FS + SVM-RBF"). It shows that in the context of the protein discrimination, the main characteristic which is relevant for the classification is the variance of the classifier. If we use a efficient regularization strategy, we obtain good results. Indeed, a 3-NN classifier is very powerful when we have an efficient representation space: a few features which are all relevant.

4 Discussion: further experiments

4.1 Perform a feature selection before a SVD

One of the main idea of this paper is to perform a feature selection before the singular value decomposition process. We think that the subsequent factors are more relevant for the protein classification. In this section, we show on the F_{23} discrimination task the new features space on the 2 first factors, using or not the feature selection.

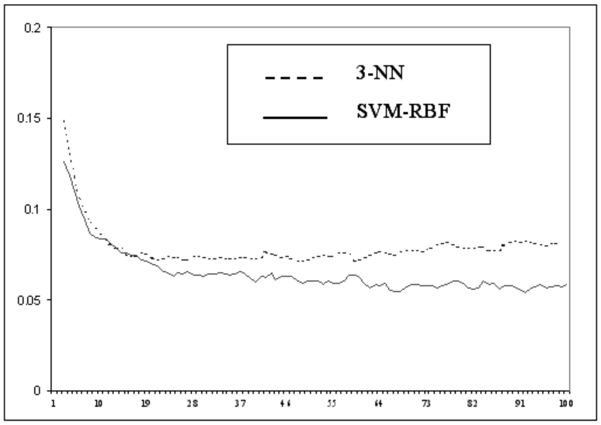


Figure 5: Error rate according the number of selected descriptors

We obtain very different representation space with or without feature selection (Figure 4.a and 4.b). We note also that the discrimination is not satisfying with SVD, some examples from different class values are overlapped (Figure 4.a, dotted circle). The error rate of the 3-NN and the SVM-RBF is about 0.14 in this situation. When we perform a feature selection before the SVD (Figure 4.b), we obtain a very different features space, and the discrimination between positive and negative class values is better. The error rate of our two learning algorithms is about 0.035 in this situation.

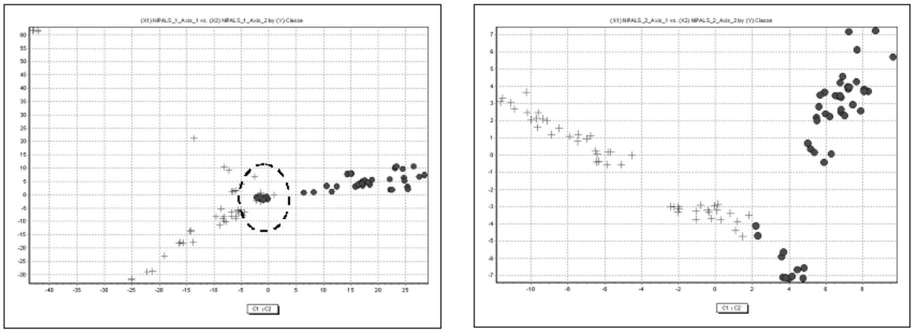
Of course, the improvement is not always too impressive. In some situations, F_{12} or F_{45} tasks for instance, the reduction of the error rate is real but not spectacular. This result shows anyway that the characteristics of the new features space produced by SVD relies heavily on the relevance of the input descriptors for the classification task.

4.2 More efficient feature selection

Select relevant features leads to improved classification accuracy. In this paper, the descriptor ranking allows to eliminate irrelevant descriptors, the SVD process allows to build a few features for the protein classification.

Another solution is to perform a more efficient feature selection, especially in order to determine the right number of relevant descriptors. In this section, we build candidate subsets of descriptors in increasing size. We evaluate them both the 3-NN and the SVM-RBF classifiers.

We compute the average error rate for the 100 first



(a)

(b)

Figure 4: Two first factors space for SVD (a) and FS-SVD (b)

descriptors for all discrimination task (Figure 5). We note that 3-NN is really sensitive to high dimensionality. The best error rate is achieved for about 30 descriptors. Using too many descriptors (upper than 60) deteriorates the classifier performance. Not because the additional descriptors are irrelevant, they seems relevant for SVM-RBF, but because we obtain mechanically a too scattered description of the dataset (too many descriptors in relation to the number of examples).

At the opposite, SVM-RBF use better the additional descriptors. Not reported in this chart, the decreasing of the error rate goes on until about 150 descriptors. When we use too many descriptors, in despite of the strong regularization of the maximum margin principle, the classifier becomes less accurate.

Of course, we can adjust the regularization parameter (the complexity cost parameter) of the SVM in order to obtain more or less smoothed decision boundary. In other experiments, when we determine the "best" complexity cost parameter by cross-validation on the whole descriptors (ALL), we achieved an error rate of 0.047, close to "'FS + SVM-RBF" scheme. But fine tuning a parameter is at the opposite of our philosophy in this study, we want a general framework which is suitable in the majority of the situations.

5 Conclusion

We use an alignment independent method for protein classification. Our approach is based on an analogy with the text mining. We use the n-grams descriptors which corresponds to amino acid sequences. The drawback of this approach is that a great number of descriptors are not relevant.

In this paper we show that elimination of irrelevant descriptors allows the singular value decomposition to produce more relevant factors for the protein classification context where we have a high dimensional boolean dataset. The approach, which one can be consider as a regularization process, is efficient especially for very instable classifier such as nearest neighbor. It is on the other hand ineffective when the learning algorithm relies already on a good regularization mechanism.

Anyway, our data preprocessing technique allows to obtain a classification scheme which is as accurate as SVM that is widely used in the bioinformatics domain. These results open new perspectives. Indeed, the singular value decomposition offers powerful tools for interpretation and visualization of results which make it possible for the expert to improve his domain knowledge. It is more suitable that *black box* classifiers which are accurate but do not give understandable results.

Another perspective is the utilization of "supervised" singular value decomposition algorithms such as PLS (Partial Least Square) regression [12] which uses explicitly the informations from the class attribute in order to build factors.

References:

[1] Mhamdi F., Elloumi M., and Rakotomalala R. Text-mining, feature selection and data-mining for proteins classification. In *Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications*. IEEE Press, 2004.

[2] Guyon I. and Elisseeff A. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, vol. 3, March 2003, pp. 1157–1182.

[3] Husbands P., Simon H., and Ding C. On the Use of the Singular Value Decomposition for Text

Retrieval. In *Proceedings of 1st SIAM Computational Information Retrieval Workshop*, 2000.

[4] Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[5] Wall M., Rechtsteiner A., and Rocha L. *A Practical Approach to Microarray Data Analysis*, chap. Singular Value Decomposition and Principal Component Analysis. Kluwer, 2003, pp. 91–109.

[6] Wold S., Esbensen K., and Geladi P. Principal Component Analysis. *Chemometrics and Intelligent Laboratory Systems*, vol. 2, 1987, pp. 37–52.

[7] Rakotomalala R. TANAGRA: une Plate-Forme d'Expérimentation pour la Fouille de Données. *Revue MODULAD*, (32), 2005, pp. 70–85.

[8] Duch W., Wieczorek T., Biesiada J., and Blachnik M. Comparison of feature ranking methods based on information entropy. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*. IEEE Press, 2004.

[9] Vapnik V. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.

[10] Chang C.C. and Lin C.J. *LIBSVM: a library for support vector machines*, 2001.

[11] Murzin A., Brenner S., Hubbard T., and Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, (247), 1995, pp. 536–540.

[12] Wold S., Martens H., and Wold H. The multivariate calibration problem in the chemistry solved by the PLS method. In *Proceedings of Conference Matrix Pencils*. Springer Verlag, 1983.