# Fast Non-negative Dimensionality Reduction for Protein Fold Recognition

Oleg Okun[1], Helen Priisalu, and Alexessander Alves[2]

[1] Infotech Oulu, 4500, 90014 Oulu, Finland
[2] LIACC, Rua do Campo Alegre, 823, 4150 Porto, Portugal
FEUP, Rua Dr Roberto Frias, 4200-465 Porto, Portugal

**Abstract.** In this paper, dimensionality reduction via matrix factorization with nonnegativity constraints is studied. Because of these constraints, it stands apart from other linear dimensionality reduction methods. Here we explore nonnegative matrix factorization in combination with a classifier for protein fold recognition. Since typically matrix factorization is iteratively done, convergence can be slow. To alleviate this problem, a significantly faster (more than 11 times) algorithm is proposed.

## 1 Introduction

It is not uncommon that for certain data sets the number of attributes $m$ is greater than the number of examples $n$. In such cases, the effect referred to as curse of dimensionality occurs, which negatively influences on clustering and classification of a given data set. Dimensionality reduction is typically used to mitigate this effect. The simplest way to reduce dimensionality is to linearly transform the original data. Given the original, high-dimensional data gathered in an $n \times m$ matrix $\mathbf{V}$, a transformed matrix $\mathbf{H}$, composed of $m$ $r$-dimensional vectors ($r < n$ and often $r \ll n$), is obtained from $\mathbf{V}$ according to the following linear transformation $\mathbf{W}$: $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W}$ is an $n \times r$ (basis) matrix. It is said that $\mathbf{W}$ and $\mathbf{H}$ are the factorized matrices and $\mathbf{WH}$ is a factorization of $\mathbf{V}$. PCA and ICA are well-known techniques performing this operation.

Nonnegative matrix factorization (NMF) also belongs to this class of methods. Unlike the others, it is based on nonnegativity constraints on all matrices involved. Thanks to this fact, it can generate a part-based representation, since no subtractions are allowed. Due that, it is claimed that NMF is capable of decomposing the whole object into meaningful parts, and having such a decomposition can make object recognition easier and often more accurate.

Lee and Seung [1] proposed a simple iterative algorithm for NMF and proved its convergence. The factorized matrices are initialized with positive random numbers before starting matrix updates. It is well known that initialization is of importance for any iterative algorithm: properly initialized, an algorithm converges faster. However, this issue was not yet investigated in case of NMF. In

J. Gama et al. (Eds.): ECML 2005, LNAI 3720, pp. 665–672, 2005.

this paper, our contribution is *two modifications accelerating algorithm convergence*: 1) feature scaling prior to NMF and 2) combination of two techniques for mapping unseen data with theoretical proof of faster convergence.

Because of its straightforward implementation, NMF has been applied to pattern classification (faces, handwritten digits, documents) [2, 3, 4]. Here we extend the application of NMF to bioinformatics: NMF coupled with a classifier is applied to protein fold recognition. Our results show a dramatic acceleration of NMF convergence (greater than 11 times on average), compared to the conventional algorithm. Moreover, statistical analysis of the error rates demonstrates that dimensionality reduction done by NMF prior to the classification in reduced space does not cause significant accuracy degradations.

## 2 Nonnegative Matrix Factorization

Given the nonnegative matrices $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ whose sizes are $n \times m$, $n \times r$ and $r \times m$, respectively, we aim at such factorization that $\mathbf{V} \approx \mathbf{WH}$. The value of $r$ is selected according to the rule $r < \frac{nm}{n+m}$ in order to obtain dimensionality reduction. NMF provides the following simple learning rule guaranteeing monotonical convergence to a local maximum [1]:

$$W_{ia} \leftarrow W_{ia} \sum_{\mu} \frac{V_{i\mu}}{(WH)_{i\mu}} H_{a\mu} \ , \tag{1}$$

$$W_{ia} \leftarrow \frac{W_{ia}}{\sum_j W_{ja}} \ , \tag{2}$$

$$H_{a\mu} \leftarrow H_{a\mu} \sum_{i} W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}} \ . \tag{3}$$

The matrices $\mathbf{W}$ and $\mathbf{H}$ are initialized with positive random values. Eqs. (1-3) iterate until convergence to a local maximum of the following objective function:

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} (V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}) \ . \tag{4}$$

After learning the NMF basis functions, i.e. the matrix $\mathbf{W}$, unseen data in the matrix $\mathbf{H}_{new}$ are mapped to $r$-dimensional space by fixing $\mathbf{W}$ and using one of the following techniques:

1. randomly initializing $\mathbf{H}$ and iterating Eq. 3 until convergence,
2. initializing $\mathbf{H}_{new} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{V}_{new}$, since $\mathbf{V}_{new} = \mathbf{W}\mathbf{H}_{new}$, where $\mathbf{V}_{new}$ contains the new data.

Further we will call the first technique *iterative* while the second - *direct*, because the latter provides a straightforward non-iterative solution.

## 3 Our Contribution

We propose two modifications in order to accelerate convergence of the iterative NMF algorithm.

The first modification concerns feature scaling (normalization) linked to the initialization of the factorized matrices. Typically, these matrices are initialized with positive random numbers, say uniformly distributed between 0 and 1, in order to satisfy the nonnegativity constraints. Hence, elements of $\mathbf{V}$ (matrix of the original data) also need to be within the same range. Given that $V_j$ is an $n$-dimensional feature vector, where $j = 1, \ldots, m$, its components $V_{ij}$ are normalized as follows: $V_{ij}/V_{kj}$, where $k = \arg\max_l V_{lj}$. In other words, components of each feature vector are divided by the maximal value among them. As a result, feature vectors are composed of components whose nonnegative values do not exceed 1. Since all three matrices ($\mathbf{V}$, $\mathbf{W}$, $\mathbf{H}$) have now entries between 0 and 1, it takes much less time to perform matrix factorization $\mathbf{V} \approx \mathbf{WH}$ (values of the entries in the factorized matrices do not have to grow/decrease much in magnitude in order to satisfy the stopping criterion for the objective function $F$ in Eq. 4) than if $\mathbf{V}$ had the original (unnormalized) values. Given that the same iterative algorithm is used in both cases (unnormalized and normalized features), it takes less time to change from 0.5 to 0.7 (normalized feature) than to change from 0.5 to 10 (unnormalized feature), because on each step the convergence rate is the same. As additional benefit, MSE becomes much smaller, too, because a difference of the original ($V_{ij}$) and approximated ($(WH)_{ij}$) values becomes smaller, given that $mn$ is fixed. Though this modification is simple, it brings significant speed of convergence as will be shown below.

The second modification concerns initialization of NMF iterations for mapping unseen data (aka generalization), i.e. after the basis matrix $\mathbf{W}$ has been learned. Since such a mapping in NMF involves only the matrix $\mathbf{H}$ ($\mathbf{W}$ is kept fixed), its initialization is to be done. We propose to initially set $\mathbf{H}$ to $(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{V}_{new}$, i.e. to the solution provided by the direct mapping technique with zeroing negative values as in Section 2, because 1) it provides a better initial approximation for $\mathbf{H}_{new}$ than a random guess, and 2) it moves the start of iterations closer toward the final point, since the objective function $F$ in Eq. 4 is increasing [1], and the inequality $F^{direct} > F^{iter}$ always holds *at initialization* (theorem below proves this fact), where $F^{direct}$ and $F^{iter}$ stand for the values of $F$ when using the direct and iterative techniques, respectively.

**Theorem 1.** *Given $F^{direct}$ and $F^{iter}$ are values of the objective function when mapping unseen data with the direct and iterative techniques, respectively. Then $F^{direct} - F^{iter} > 0$ always holds at the start of iterations when using Eq. 3.*

*Proof.* By definition,

$$F^{iter} = \sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij}\log(WH)_{ij} - (WH)_{ij}) \ ,$$

$$F^{direct} = \sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij}\log V_{ij} - V_{ij}) \ .$$

The difference $F^{direct} - F^{iter}$ is equal to

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij}\log V_{ij} - V_{ij} - V_{ij}\log(WH)_{ij} + (WH)_{ij}) =$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left(V_{ij}(\log\frac{V_{ij}}{(WH)_{ij}} - 1) + (WH)_{ij}\right) =$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left(V_{ij}\log\frac{V_{ij}}{10(WH)_{ij}} + (WH)_{ij}\right) \ .$$

Given all three matrices involved are nonnegative, the last expression is positive if either condition is satisfied:

1. $\log\frac{V_{ij}}{10(WH)_{ij}} > 0$,
2. $(WH)_{ij} > V_{ij}\log\frac{10(WH)_{ij}}{V_{ij}}$.

Let us introduce a new variable, $t$: $t = \frac{V_{ij}}{(WH)_{ij}}$. Then the above conditions can be written as

1. $\log\frac{t}{10} > 0$ or $logt > 1$,
2. $1 > t\log\frac{t}{10}$ or $\log t < \frac{t+1}{t}$.

The first condition is satisfied if $t > 10$ whereas the second if $t < t_0$ ($t_0 \approx 12$). Therefore either $t > 10$ or $t < 12$ should be satisfied for $F^{direct} > F^{iter}$. Since the union of both conditions covers the whole interval $[0, +\infty[$, it means that $F^{direct} > F^{iter}$, independently of $t$, i.e. of whether $V_{ij} > (WH)_{ij}$ or not. Q.E.D.

□

Because our approach combines both direct and iterative techniques for mapping unseen data, we will call it *iterative2*.

## 4 Summary of Our Algorithm

1. Scale both training and test data and randomly initialize the factorized matrices as described in Section 3. Choose $r$.
2. Iterate Eqs. 1-3 until convergence to obtain the NMF basis matrix $\mathbf{W}$ and to map training data to NMF (reduced) space.
3. Given $\mathbf{W}$, map test data by using the direct technique. Set to zero negative values in the resulting matrix $\mathbf{H}_{new}^{direct}$.
4. Fix the basis matrix and iterate Eq. 3 until convergence by using $\mathbf{H}_{new}^{direct}$ at initialization of iterations.

## 5 Experiments

Experiments with NMF involve estimation of the error rate when combining NMF and a classifier. Three techniques for generalization are used: direct, iterative, and iterative2. The training data are mapped into reduced space according to Eqs. 1-3 with simultaneous learning of the matrix **W**. Tests were repeated 10 times to collect statistics necessary for comparison of three generalization techniques. Each time, a different random initialization for learning the basis matrix **W** was used, but the same learned basis matrix was utilized in each run for all generalization techniques in order to create as fair comparison of three generalization techniques as possible.

Values of $r$ (dimensionality of reduced space) were set to 25, 50, 75, and 88 (max), which constitutes 20%, 40%, 60%, and 71.2% of the original dimensionality, respectively. In all reported statistical tests $\alpha = 0.05$. All algorithms were implemented in MATLAB running on a Pentium 4 (3 GHz CPU, 1GB RAM).

### 5.1 Data

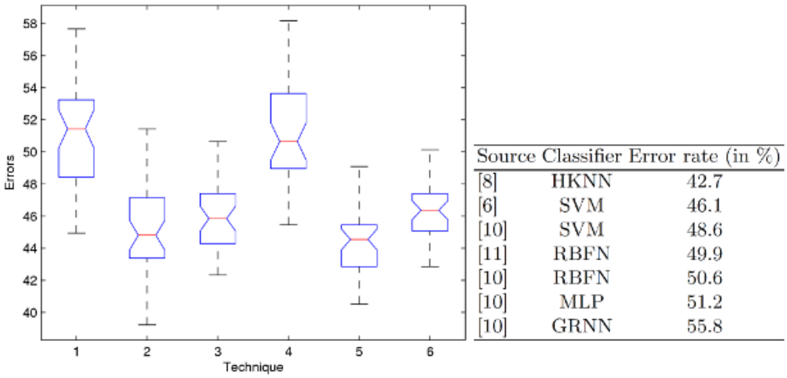
In bioinformatics, it is rather common to use a single data set in experiments, since many tasks in this field are much more difficult than those in general machine learning. A challenging data set [5] was used in experiments. The data set contains the 27 most populated folds represented by seven or more proteins. Ding and Dubchak already split it into the training and test sets, which we will use as other authors did. Feature vectors have 125 dimensions. The training set consists of 313 protein folds having no more than 35% of the sequence identity for aligned subsequences longer than 80 residues. The test set of 385 folds is composed of protein sequences of less than 40% identity with each other and less than 35% identity with the proteins of the first set. This, as well as multiple classes, many of which sparsely represented, render this task extremely difficult.

### 5.2 Classification Results

The K-Local Hyperplane Distance Nearest Neighbor (HKNN) [6] was selected as a classifier, since it demonstrated a competitive performance. Table 1 shows the error rates obtained with HKNN, SVM, and various neural networks when classifying protein folds in the original, 125-dimensional space.

The *normalized* features were used since feature normalization prior to HKNN increases classification accuracy. The optimal values for two parameters of HKNN, $K$ and $\lambda$, determined via cross-validation, are 7 and 8, respectively.

Let us now turn to the error rates in dimensionally reduced space. For each value of $r$, NMF followed by HKNN were repeated 10 times. As a results, a 40x6 matrix containing the error rates was generated. This matrix is then subjected to the one-way analysis of variance (ANOVA) and multiple comparison tests in order to make statistically driven conclusions. Table 5.2 shows identifiers associated with the generalization techniques. Error bars for all generalization techniques are given in Fig. 1.



| Source | Classifier | Error rate (in %) |
|---|---|---|
| [8] | HKNN | 42.7 |
| [6] | SVM | 46.1 |
| [10] | SVM | 48.6 |
| [11] | RBFN | 49.9 |
| [10] | RBFN | 50.6 |
| [10] | MLP | 51.2 |
| [10] | GRNN | 55.8 |

**Fig. 1.** (a) Error bars resulting from NMF using six generalization techniques; (b) Classification errors on the original space of the protein folds dataset

The one-way ANOVA test is first utilized in order to find out whether the mean error rates of all six techniques are the same (null hypothesis $H_0 : \mu_1 = \mu_2 = \cdots = \mu_6$) or not. If the returned p-value is smaller than $\alpha = 0.05$, the null hypothesis is rejected, which implies that the mean error rates are not the same. The next step is to determine which pairs of means are significantly different, and which are not by means of the multiple comparison test.

Table 5.2 contains results of the multiple comparison test and it is seen that these results confirm that the direct technique stands apart from both iterative techniques. The main conclusions from Table 5.2 are $\mu_1 = \mu_4$ and $\mu_2 = \mu_3 = \mu_5 = \mu_6$, i.e. there are two groups of techniques, and the mean of the first group is larger than that of the second group.

The last column in Table 5.2 points to the very interesting result: whether feature normalization prior to NMF is applied or not, the standard deviation of the error rate of our technique is lower than that for the conventional one, which, in turn, is lower than the standard deviation for the direct technique. It implies that our modifications of NMF led to a *visible reduction in the deviation of classification error*! This reduction is caused by shrinking the search space of possible factorizations, and it is larger if normalization prior to NMF is used.

**Table 1.** Mean error for each generalization technique (standard error 0.4)

| Identifier | Technique | Scaling prior to NMF | Mean error | Std. deviation |
|---|---|---|---|---|
| 1 | Direct | No | 50.93 | 3.08 |
| 2 | Iterative | No | 44.94 | 2.52 |
| 3 | Iterative2 | No | 45.98 | 2.11 |
| 4 | Direct | Yes | 51.38 | 3.31 |
| 5 | Iterative | Yes | 44.52 | 2.13 |
| 6 | Iterative2 | Yes | 46.32 | 1.72 |

**Table 2.** Results of the multiple comparison test

| Identifier | Identifier 2 | Lower bound | Difference | Upper bound | Outcome |
|---|---|---|---|---|---|
| 4 | 5 | 5.25 | 6.87 | 8.48 | Reject $H_0 : \mu_4 \neq \mu_5$ |
| 4 | 2 | 4.83 | 6.45 | 8.07 | Reject $H_0 : \mu_4 \neq \mu_2$ |
| 4 | 3 | 3.79 | 5.41 | 7.03 | Reject $H_0 : \mu_4 \neq \mu_3$ |
| 4 | 6 | 3.45 | 5.07 | 6.69 | Reject $H_0 : \mu_4 \neq \mu_6$ |
| 4 | 1 | -1.16 | 0.46 | 2.07 | Accept $H_0 : \mu_4 = \mu_1$ |
| 1 | 5 | 4.79 | 6.41 | 8.03 | Reject $H_0 : \mu_1 \neq \mu_5$ |
| 1 | 2 | 4.38 | 5.99 | 7.61 | Reject $H_0 : \mu_1 \neq \mu_2$ |
| 1 | 3 | 3.34 | 4.96 | 6.57 | Reject $H_0 : \mu_1 \neq \mu_3$ |
| 1 | 6 | 3.00 | 4.61 | 6.23 | Reject $H_0 : \mu_1 \neq \mu_6$ |
| 6 | 5 | 0.18 | 1.80 | 3.42 | Reject $H_0 : \mu_6 \neq \mu_5$ |
| 6 | 2 | -0.24 | 1.38 | 3.00 | Accept $H_0 : \mu_6 = \mu_2$ |
| 3 | 5 | -0.16 | 1.46 | 3.07 | Accept $H_0 : \mu_3 = \mu_5$ |

**Table 3.** Gains in time resulted from modifications of the conventional NMF algorithm

| | Gain due to scaling prior to NMF for | | | | | Gain due to initialization for | |
|---|---|---|---|---|---|---|---|
| | learning | generalization | | learning+generalization | | generalization | |
| $r$ | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | $R_7$ |
| 88 | 11.9 | 11.4 | 10.4 | 11.7 | 11.5 | 1.5 | 1.4 |
| 75 | 13.8 | 12.9 | 13.1 | 13.6 | 13.7 | 1.6 | 1.6 |
| 50 | 13.2 | 11.1 | 12.5 | 12.6 | 13.1 | 1.6 | 1.7 |
| 25 | 9.5 | 6.4 | 8.8 | 8.7 | 9.5 | 1.6 | 2.2 |
| Average | 12.1 | 10.4 | 11.2 | 11.6 | 11.9 | 1.6 | 1.8 |

That is, our initialization eliminates some potentially erroneous solutions before iterations even start and leads to more stable classification error.

One can say that the error rates in reduced space are larger than the error rate (42.7%) achieved in the original space. However, it is not, in general, uncommon to observe similar effects when doing classification after dimensionality reduction (see, e.g. [7]). Nevertheless, we observed that sometimes error in reduced space can be lower than 42.7: for example, the minimal error when applying the iterative technique with no scaling before NMF and $r = 50$ is 39.22, while the minimal error when using the iterative2 technique under the same conditions is 42.34. Varying errors can be attributed to the fact that NMF factorization of a given matrix may not be unique. Finally, even though NMF+HKNN led to the higher error rates than HKNN alone, the former was nevertheless superior (see Tables 1 and 5.2) to neural networks and comparable to SVMs, applied without NMF.

### 5.3 Time Results

Table 3 presents speed gains resulted from our modifications for different dimensionalities of reduced space. R1 stands for the speed gains due to scaling on the task of learning and mapping training data. R2 and R3 are the speed gains obtained due to scalling on the generalization task using iterative and iterative2. R4 and R5 are the same gains obtained on the task of training followed by generalization. R6 and R7 are the speed gains obtained due to applying iterative2 instead of iterative versus scaling. As a result, the average gain in time obtained with our modifications is more than 11 times.

## 6 Conclusion

The main contribution of this work is two modifications of the basic NMF algorithm and its practical application to a challenging real-world task, namely protein fold recognition. The first modification concerns feature scaling before NMF while the second modification combines two known generalization techniques, which we called direct and iterative; the former is used as a starting point for updates of the latter, thus leading to a new generalization technique. We proved (both theoretically and experimentally) that our technique converges faster than the ordinary iterative technique. On the data set studied, the average gain in convergence speed exceeds 11 times.

When combining the modified NMF with a classification algorithm, statistical analysis of the obtained results indicates that the mean error associated with the direct technique is higher than that related to either iterative technique while both iterative techniques lead to the statistically similar error rates. Since our technique provides a faster mapping of unseen data, it is advantageous to apply it instead of the ordinary one. In addition, our technique results in a smaller deviation of classification error, thus making classification more stable.

## References

1. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401 (1999) 788–791
2. Buciu, I.,Pitas, I.: Application of non-negative and local non-negative matrix factorization to facial expression recognition. In: Proceedings of the Seventeenth International Conference on Pattern Recognition, Cambridge, UK. (2004) 288-291
3. Chen, X., Gu, L., Li, S., Zhang, H.J.: Learning representative local features for face detection. In: Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HW. (2001) 1126-1131
4. Guillamet, D.e.a.: Introducing a weighted non-negative matrix factorization for image classification. Pattern Recognition Letters 24 (2003) 2447-2454
5. Ding, C.,Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17 (2001) 349-358
6. Vincent, P., Bengio, Y.: K-local hyperplane and convex distance nearest neighbor algorithms. In Dietterich, T., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems 14. MIT Press, Cambridge, MA (2002) 985-992
7. Pal, N., Chakraborty, D.: Some new features for protein fold recognition. In: LNCS - ICANN/ICONIP 2003. Volume 2714. Springer (2003) 1176-1183