Unsupervised dimensionality reduction via gradient-based matrix factorization with two adaptive learning rates

Vladimir Nikulin VNIKULIN.UQ@GMAIL.COM
Tian-Hsiang Huang HUANGTX@GMAIL.COM
Department of Mathematics, University of Queensland, Australia

Editor: I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver

Abstract

The high dimensionality of the data, the expressions of thousands of features in a much smaller number of samples, presents challenges that affect applicability of the analytical results. In principle, it would be better to describe the data in terms of a small number of meta-features, derived as a result of matrix factorization, which could reduce noise while still capturing the essential features of the data. Three novel and mutually relevant methods are presented in this paper: 1) gradient-based matrix factorization with two adaptive learning rates (in accordance with the number of factor matrices) and their automatic updates; 2) nonparametric criterion for the selection of the number of factors; and 3) nonnegative version of the gradient-based matrix factorization which doesn't require any extra computational costs in difference to the existing methods. We demonstrate effectiveness of the proposed methods to the supervised classification of gene expression data.

Keywords: matrix factorization, unsupervised learning, clustering, nonparametric criterion, nonnegativity, bioinformatics, leave-one-out, classification

1. Introduction

The analysis of gene expression data using matrix factorization has an important role to play in the discovery, validation, and understanding of various classes and subclasses of cancer. One feature of microarray studies is the fact that the number of samples collected is relatively small compared to the number of genes per sample which are usually in the thousands. In statistical terms this very large number of predictors compared to a small number of samples or observations makes the classification problem difficult.

Many standard classification algorithms have difficulties in handling high dimensional data and due to a relatively low number of training samples, they tend to overfit. Moreover, usually only a small subset of examined genes is relevant in the context of a given task. For these reasons, feature selection methods are an inevitable part of any successful microarray data classification algorithm.

Another approach to reduce the overfitting is to describe the data in terms of metagenes as a linear combinations of the original genes.

Recently, models such as independent component analysis and nonnegative matrix factorization (NMF) have become popular research topics due to their obvious usefulness (Oja *et al.*, 2010). All these blind latent variable models in the sense that no prior knowledge on the variables is used, except some broad properties like gaussianity, statistical independence, or nonnegativity.

As pointed out by Tamayo *et al.* (2007), the metagene factors are a small number of gene combinations that can distinguish expression patterns of subclasses in a data set. In many cases, these linear combinations of the genes are more useful in capturing the invariant biological features of the data. In general terms, matrix factorization, an unsupervised learning method, is widely used to study the structure of the data when no specific response variable is specified.

Examples of successful matrix factorization methods are singular value decomposition and nonnegative matrix factorization (Lee and Seung, 1999). In addition, we developed a novel and very fast algorithm for gradient-based matrix factorization (GMF), which was introduced in our previous study (Nikulin and McLachlan, 2009).

The main subject of this paper is a more advanced version of the GMF. We call this algorithm as GMF with two learning rates and their automatic updates (A2GMF). Details about this algorithm are given in Section 2.1. The main features of the A2GMF are two flexible (adaptive) learning rates in accordance to the number of factor matrices. By the definition, the learning rates will be updated during learning process. The explicit update formulas are given in Section 2.1.

Clearly, the number of factors is the most important input parameter for any matrix factorization algorithm. In Section 2.2 we are proposing a novel unsupervised method for the selection of the number of factors. This method is absolutely general, and we show in Section 4.5 one possible extension to the clustering methodology, see Figure 3. Using supervised classification with leave-one-out (LOO) evaluation criterion, we can develop another approach to select the numbers of factors. Classification results, which are based on five well-known and publicly available datasets, and presented in Section 4 demonstrate correspondence between the outcomes of both methods. However, speed is the main advantage of the proposed here nonparametric criterion.

The third proposed novelty is a nonnegative version of GMF (NN-GMF), see Section 2.3. Essentially, an implementation of the NN-GMF doesn't require any extra computational costs. Consequently, the NN-GMF is as fast as GMF, and may be particularly useful for a wide range of the professionals who are working with real data directly and who are interested to find out an interpretation of the data in terms of meta-variables.

2. Methods

Let $(\mathbf{x}_j, y_j)\,, j = 1, \ldots, n$, be a training sample of observations where $\mathbf{x}_j \in \mathbb{R}^p$ is p-dimensional vector of features, and y_j is a multiclass label, which will not be used in this section. Boldface letters denote vector-columns, whose components are labelled using a normal typeface. Let us denote by $\mathbf{X} = \{x_{ij}, i = 1, \ldots, p, j = 1, \ldots, n\}$ the matrix containing the observed values on the n samples.

For gene expression studies, the number p of genes is typically in the thousands, and the number n of experiments is typically less than 100. The data are represented by an expression matrix $\mathbf{X}$ of size $p \times n$, whose rows contain the expression levels of the p genes in the n samples. Our goal is to find a small number $k \ll p$ of metagenes or factors. We can then approximate the gene expression patterns of samples as a linear combinations of these metagenes. Mathematically, this corresponds to factoring matrix $\mathbf{X}$ into two matrices

$$\mathbf{X} \sim \mathbf{AB}, \tag{1}$$

where weight matrix $\mathbf{A}$ has size $p \times k$, and the matrix of metagenes $\mathbf{B}$ has size $k \times n$, with each of k rows representing the metagene expression pattern of the corresponding sample.

Algorithm 1 A2GMF.

```
1: Input: $\mathbf{X}$ - microarray matrix.
2: Select $m$ - number of global iterations; $k$ - number of factors; $\lambda_a, \lambda_b > 0$ - initial learning rates,
   $0 < \tau(\ell) \le 1$ is a scaling function of the global iteration $\ell$.
3: Initial matrices $\mathbf{A}$ and $\mathbf{B}$ are generated randomly.
4: Global cycle: for $\ell = 1$ to $m$ repeat steps 5 - 16:
5: genes-cycle: for $i = 1$ to $p$ repeat steps 6 - 15:
6: samples-cycle: for $j = 1$ to $n$ repeat steps 7 - 15:
7: compute prediction $S_{ij} = \sum_{f=1}^{k} a_{if} b_{fj}$;
8: compute error of prediction: $E_{ij} = x_{ij} - S_{ij}$;
9: internal factors-cycle: for $f = 1$ to $k$ repeat steps 10 - 15:
10: compute $\alpha = a_{if} b_{fj}$;
11: update $a_{if} \Leftarrow a_{if} - \tau(\ell) \cdot \lambda_a \cdot g_{ifj}$;
12: $E_{ij} \Leftarrow E_{ij} + \alpha - a_{if} b_{fj}$;
13: compute $\alpha = a_{if} b_{fj}$;
14: update $b_{fj} \Leftarrow b_{fj} - \tau(\ell) \cdot \lambda_b \cdot h_{ifj}$;
15: $E_{ij} \Leftarrow E_{ij} + \alpha - a_{if} b_{fj}$;
16: update $\lambda_a$ and $\lambda_b$ according to (5) and (7);
17: Output: $\mathbf{A}$ and $\mathbf{B}$ - matrices of metagenes or latent factors.
```

2.1. A2GMF algorithm under the stochastic gradient descent framework

Let us consider the following loss function

$$L(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{p} \sum_{j=1}^{n} E_{ij}^2, \tag{2}$$

where $E_{ij} = x_{ij} - \sum_{f=1}^{k} a_{if} \cdot b_{fj}$.

The above target function (2) includes in total $k(p+n)$ regulation parameters and may be unstable if we minimise it without taking into account the mutual dependence between elements of the matrices $\mathbf{A}$ and $\mathbf{B}$.

Derivatives of the function E are given below:

$$g_{ifj} = \frac{\partial E_{ij}^2}{\partial a_{if}} = -2 \cdot E_{ij} \cdot b_{fj}, \tag{3a}$$

$$h_{ifj} = \frac{\partial E_{ij}^2}{\partial b_{fj}} = -2 \cdot E_{ij} \cdot a_{if}. \tag{3b}$$

Considering step 11 of Algorithm 1, we can replace in (2) a_{if} by its update $a_{if} - \lambda_a g_{ifj}$, assuming that $\tau(\ell) = 1$, where scaling function τ is defined in Remark 2.1. After that, we can rewrite (2) in the following way

$$L(\mathbf{A}, \mathbf{B}, \lambda_a) = \sum_{i=1}^{p} \sum_{j=1}^{n} (E_{ij} + \lambda_a U_{ij})^2, \tag{4}$$

where

$$U_{ij} = \sum_{f=1}^{k} g_{ifj} \cdot b_{fj} = -2E_{ij} \sum_{f=1}^{k} b_{fj}^2.$$

Minimising (4) as a function λ_a, we shall find

$$\lambda_a = -\frac{\sum_{i=1}^{p} \sum_{j=1}^{n} E_{ij} U_{ij}}{\sum_{i=1}^{p} \sum_{j=1}^{n} U_{ij}^2}$$

$$= \frac{1}{2} \frac{\sum_{j=1}^{n} \sum_{f=1}^{k} b_{fj}^2 \sum_{i=1}^{p} E_{ij}^2}{\sum_{j=1}^{n} (\sum_{f=1}^{k} b_{fj}^2)^2 \sum_{i=1}^{p} E_{ij}^2} = \frac{1}{2} \frac{\sum_{j=1}^{n} b_j \phi_j}{\sum_{j=1}^{n} b_j^2 \phi_j}, \tag{5}$$

where

$$b_j = \sum_{f=1}^{k} b_{fj}^2, \;\; \phi_j = \sum_{i=1}^{p} E_{ij}^2.$$

Considering step 14 of Algorithm 1, we can replace in (2) b_{fj} by its update $b_{fj} - \lambda_b h_{ifj}$, assuming that $\tau(\ell) = 1$. After that, we can re-write (2) in the following way

$$L(\mathbf{A}, \mathbf{B}, \lambda_b) = \sum_{i=1}^{p} \sum_{j=1}^{n} (E_{ij} + \lambda_b V_{ij})^2, \tag{6}$$

where

$$V_{ij} = \sum_{f=1}^{k} h_{ifj} \cdot a_{if} = -2E_{ij} \sum_{f=1}^{k} a_{if}^2.$$

Minimising (6) as a function λ_b, we shall find

$$\lambda_b = -\frac{\sum_{i=1}^{p} \sum_{j=1}^{n} E_{ij} V_{ij}}{\sum_{i=1}^{p} \sum_{j=1}^{n} V_{ij}^2}$$

$$= \frac{1}{2} \frac{\sum_{i=1}^{p} \sum_{f=1}^{k} a_{if}^2 \sum_{j=1}^{n} E_{ij}^2}{\sum_{i=1}^{p} (\sum_{f=1}^{k} a_{if}^2)^2 \sum_{j=1}^{n} E_{ij}^2} = \frac{1}{2} \frac{\sum_{i=1}^{p} a_i \psi_i}{\sum_{i=1}^{p} a_i^2 \psi_i}, \tag{7}$$

where

$$a_i = \sum_{f=1}^{k} a_{if}^2, \;\; \psi_i = \sum_{j=1}^{n} E_{ij}^2.$$

Figure 1(a-e) illustrates clear advantage of the A2GMF compared to the GMF algorithm. We did not try to optimize performance of the A2GMF, and used the same regulation parameters against all five datasets described in Section 3.

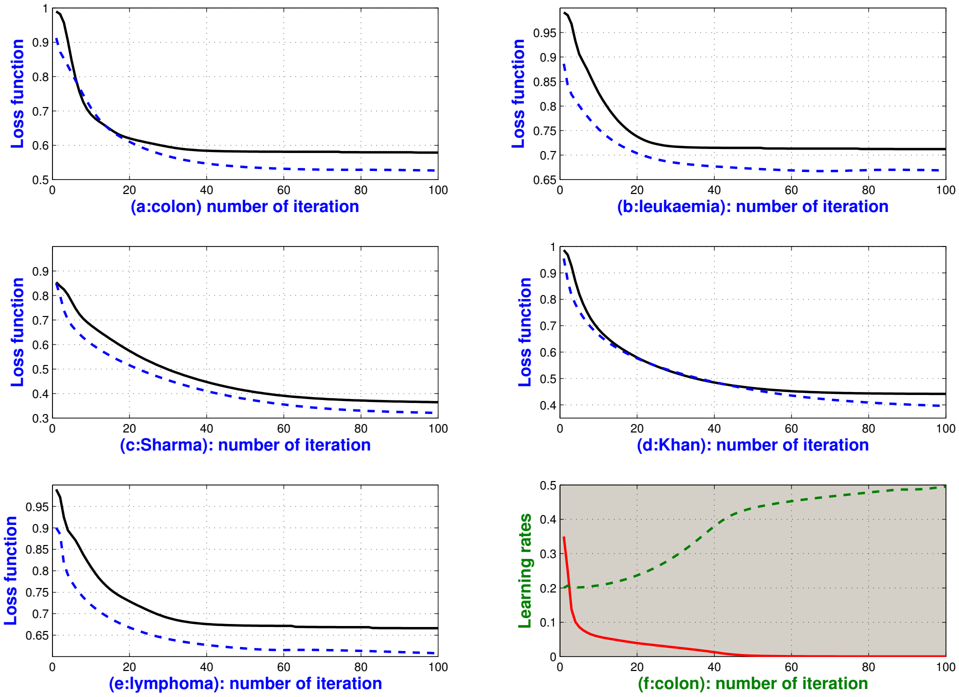

Figure 1: Convergence of the GMF (black line) and A2GMF (dashed blue line, Algorithm 1) algorithms in the cases of (a) colon, (b) leukaemia,(c) Sharma, (d) Khan and (e) lymphoma sets. The last window (f) illustrates the difference in behavior between learning rates λ_a (red) and λ_b (dashed green), see Algorithm 1.

Remark 1 *Scaling function τ is a very important in order to ensure stability of Algorithm 1. Assuming that the initial values of matrices A and B were generated randomly, we have to use smaller values of τ during the first few global iterations. The following structure of the function $\tau(\ell)$ was used for the illustrations given in Figure 1:*

$$\tau(\ell) = 0.005(1 - r) + 0.2r, r = \sqrt{\frac{\ell}{m}},$$

where parameters ℓ and m are defined in the body of Algorithm 1.

Remark 2 *We note an important fact: a significant difference in behavior of the learning rates λ_a and λ_b as functions of the global iteration, see Figure 1(f). This difference reflects, also, the main conceptual difference between A2GMF and GMF algorithms.*

Definition 3 *We define GMF algorithm as Algorithm 1 but without the scaling function τ and with constant and equal learning rates $\lambda_a = \lambda_b = 0.01$ (Nikulin and McLachlan, 2009).*

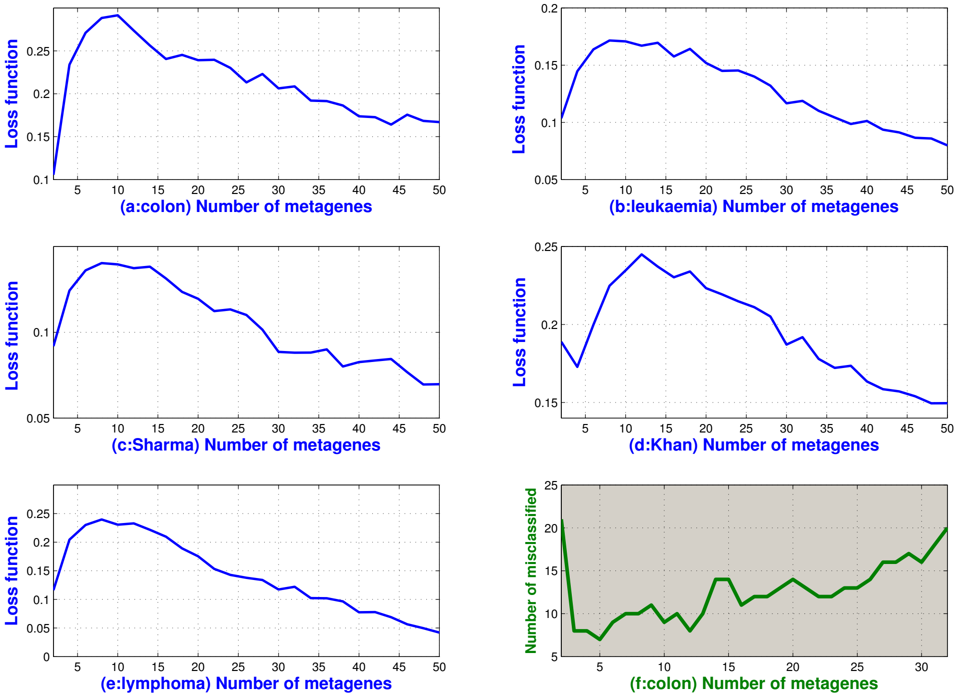

Figure 2: Nonparametric feature selection with criterion (8), where (a) colon, (b) leukaemia, (c) Sharma, (d) Khan and (e) lymphoma datasets. Window (f) illustrates the numbers of misclassified samples as a function of k - number of metagenes, where we used evaluation scheme $Alg.1 + LOO\{lm\}$ applied to the colon dataset.

The value 0.01 *was used in the comparison experiments Figure 1(a-e). However, the optimal settings may be different depending on the particular dataset.*

2.2. Nonparametric unsupervised criterion for selection of the number of meta-variables

The basic idea of nonparametric inference is to use data to infer an unknown quantity while making as few assumptions as possible (Wasserman, 2006). The criterion, which we propose in this section is the most general and may attract interest as a self-target in fundamental sense, but not only as an application for the selection of the number of meta-variables. It is very logical to assume that the microarray matrix contains some systematic dependencies, which will be discovered or revealed by the matrix factorization. We can easily destroy those hidden dependencies by randomly re-shuffling the elements within any column. Note, also, that all the columns will be re-shuffled independently.

Let us denote by $X^{(\gamma)}, \gamma = 1, \ldots, 3$ three random re-shuffling of the original matrix X. By $A^{(\gamma)}$ and $B^{(\gamma)}$ we shall denote the corresponding factor matrices. Figure 2 illustrates behavior of

$$\mathcal{D}_k = \frac{1}{3}\sum_{\gamma=1}^{3} \Phi_k^{(\gamma)} - \Phi_k, \tag{8}$$

as a function of k, where

$$\Phi_k = \sqrt{\frac{L(A_k, B_k)}{pn}}, \Phi_k^{(\gamma)} = \sqrt{\frac{L(A_k^{(\gamma)}, B_k^{(\gamma)})}{pn}}, \gamma = 1, \ldots, 3.$$

As expected, the values of $\mathcal{D}_k$ are always positive. Initially, and because of the small number of factors k, the values of $\mathcal{D}_k$ are low. This fact has very simple explanation, as far as the number of metagenes is small, those metagenes are not sufficient to contain the hidden dependencies and relations. Then, the values of $\mathcal{D}_k$ will grow to some higher level. The pick point appears to be an appropriate criterion for the selection of the optimal number of meta-variables. After that point, the values of $\mathcal{D}_k$ will decline because of the overfitting.

2.3. On the nonnegative modification of the GMF (NN-GMF)

Many real-world data are nonnegative and the corresponding hidden components have a physical meaning only when nonnegative (Cichocki *et al.*, 2009). Suppose that microarray matrix X is nonnegative (that means, all the elements of the matrix are nonnegative). Then, it is very logical to have factor matrices A and B in (1) to be nonnegative as well. In order to implement this, we can easily create a special modification of Algorithm 1.

The main idea behind this modification is a remarkably simple. Let us consider a generalization of (1)

$$\mathbf{X} \sim \xi(\mathbf{A})\xi(\mathbf{B}), \tag{9}$$

where ξ is a differentiable function, and $\xi(A)$ is a matrix with the same dimensions as matrix A and elements $\{\xi(a_{ij}), i = 1, \ldots, p, j = 1, \ldots, k\}$.

Taking into account the fact that an exponential function is always nonnegative, we can consider $\xi(\cdot) = \exp(\cdot)$. Equation (9) represents a flexible framework, where we can apply any suitable function. For example, we can create another modification of the NN-GMF with squared or logistic function.

According to Gao and Church (2005), it is obvious that dimension reduction to a much lower dimension (smaller than the number of observations) is appropriate. Principal component analysis (PCA) or singular value decomposition and partial least squares are two such methods that have been applied to cancer classification with satisfactory results. However, due to the holistic nature of PCA, the resulting components are global interpretations and lack intuitive meaning. To solve this problem, Lee and Seung (1999) demonstrated that NMF is able to learn localized features with obvious interpretation. Their work was applied elegantly to image and text analysis.

According to Fogel *et al.* (2007), singular value decomposition (SVD) attempts to separate mechanisms in an orthogonal way, although nature is all but orthogonal. As a consequence, SVD components are unlikely to match with real mechanisms and so are not easily interpreted. On the contrary, NMF appears to match each real mechanism with a particular component.

Clearly, we can extend above two statements to the NN-GMF algorithm.

As it was pointed by Lin (2007), there are many existing methods for NMF. Most of those method are presented in (Cichocki *et al.*, 2009). However, NN-GMF is an essentially novel algorithm (based on the different platform), and can not be regarded as a modification of NMF (Lee and Seung, 1999).

2.4. Sparsity and Regularization

A sparse representation of the data by a limited number of components is an important research problem. In machine learning, sparseness is closely related to feature selection and certain generalizations in learning algorithms, while nonnegativity relates to probability distributions (Cichocki *et al.*, 2009).

The most standard way to achieve sparsity is to include regularization term in (2), which penalises usage of the elements of the matrices **A** and **B**. We shall exploit here flexibility of Algorithm 1. The structure of the regularized GMF will be the same, but we have to make some changes in the formulas for derivatives (3a) and (3b), which may be found using standard techniques (Nikulin and McLachlan, 2010).

2.5. Handling missing values

Suppose that some elements of the matrix **X** have low level of confidence or missing. Based on the principles of the stochastic gradient descent technique, we are considering during any global iteration not the whole target function, but particular terms of the target function. Accordingly, we can easily exclude from an update process those terms, which have low level of confidence or missing. Using GMF algorithm we can make possible factorization of a huge matrices, for example, in the case of the well known Netflix Cup (marketing applications) we are dealing with hundreds of thousands of customers and tens of thousands of items, assuming that only about 5% or 10% of true relationships or preferences are available (Koren, 2009).

2.6. Boosting with GMF

Algorithm 1 represents a flexible framework, where we can easily insert suitable model as a component. For example, we can use the idea of boosting (Dettling and Buhlmann, 2003) in the following way. In the case if the element x_{ij} has been approximated poorly (step N8), we can increase the value of learning rate. On the other hand, in the case if an absolute value of E_{ij} is small enough, we can overpass the steps NN9-15. As a direct consequence, the algorithm will run faster and faster in line with general improvement of the quality of approximation.

3. Data

All experiments were conducted against 5 well known and publicly available microarray datasets named colon, leukaemia, Sharma, Khan and lymphoma. Some basic statistical characteristics of the datasets are given in Table 2, and more details are presented in (Nikulin *et al.*, 2011).

Table 1: Classification results (numbers of misclassified samples) in the case of the scheme: $SVD + LOO\{lm\}$.

k	colon	leukaemia	Sharma	Khan	lymphoma
2	22	8	24	48	6
4	10	5	22	27	2
6	9	2	18	15	1
8	**7**	**1**	14	5	**0**
10	8	1	**9**	6	0
12	8	1	10	2	0
14	8	2	9	**0**	0
16	10	2	10	0	0
18	9	2	10	0	0
20	10	2	11	0	0
22	11	5	10	0	0
24	13	3	13	0	0
26	13	1	15	1	1
28	11	1	16	1	1
30	14	2	16	1	2
32	13	3	16	3	1
34	9	2	18	2	1
36	12	3	18	3	1
38	14	4	18	3	2
40	15	6	20	3	3

4. Experiments

After decomposition of the original matrix $\mathbf{X}$, we used the leave-one-out (LOO) evaluation scheme, applied to the matrix of metagenes $\mathbf{B}$. This means that we set aside the ith observation and fit the classifier by considering remaining $(n-1)$ data points. We conducted experiments with lm function in R, which is a standard linear regression without any regulation parameters.

4.1. Evaluation scheme

We shall denote such a scheme (Nikulin and McLachlan, 2009) as

$$Alg.1 + LOO\{lm\}. \tag{10}$$

Above formula has very simple interpretation as a sequence of two steps: 1) compute the matrix of metagenes B using Algorithm 1 (dimensionality reduction), and 2) run LOO evaluation scheme with lm as a base classification model.

As a reference, we mention that similar evaluation model was employed in (Li and Ngom, 2010) to classify samples in NMF space using kNN algorithm. In accordance with the evaluation framework (10), we can express this model in the following way: $NMF + LOO\{kNN\}$.

Remark 4 *As far as Algorithm 1 is an unsupervised (labels are not used), we do not have to include it into the LOO internal loop in (10).*

Table 2: Comparison between the numbers of factors (metagenes) k_{NP} and k^*, where k_{NP} was selected according to the nonparametric criterion of Section 2.2 as a point on the horizontal axis corresponding to the maximum loss, see Figure 2(a-e); and k^* was selected according to the independent scheme $SVD + LOO\{lm\}$, see Table 2 (in supplementary material), where column "NM" indicates the numbers of corresponding misclassified samples.

Data	n	p	k^*	k_{NP}	NM
Colon	62	2000	8	10	7
Leukaemia	72	1896	8	8	1
Sharma	60	1368	10	8	9
Khan	83	2308	14	12	0
Lymphoma	62	4026	8	8	0

Table 3: Classification results in the case of the scheme $Alg.1 + LOO\{lm\}$, where we used 20 runs against colon dataset with random initial settings, columns *"min, mean, max, std"* represent the corresponding statistical characteristics for the numbers of misclassified samples. In the column "SVD" we presented the numbers of misclassified samples in the case of the scheme $Alg.1 + LOO\{lm\}$, where initial settings were generated using SVD method.

k	min	mean	max	std	SVD
2	16	21.89	28	4.50	21
4	7	15.22	19	3.53	10
6	7	10.56	18	3.39	9
8	5	8.50	11	2.09	7
10	6	8.92	12	1.72	6
12	6	9.33	12	2	7
14	8	9.22	11	1.09	9
16	7	9.89	12	1.54	10
18	8	10	11	1.12	10
20	8	10.33	13	1.58	10
22	11	12.56	15	1.59	10
24	11	12.78	15	1.20	13
26	12	12.56	14	0.73	12
28	10	12.67	16	2.06	12
30	10	13.22	16	1.64	14
32	12	14.33	17	1.58	14
34	12	15.11	18	1.90	12
36	11	14	17	2.18	14
38	13	14.78	17	1.39	14
40	13	15	17	1.22	14

4.2. Singular value decomposition (SVD)

An alternative factorisation may be created using *svd* function in Matlab or R:

$$X = UDV,$$

where U is $p \times n$ matrix of orthonormal eigenvectors, D is $n \times n$ diagonal matrix of non-negative eigenvalues, which are sorted in a decreasing order, and V is $n \times n$ matrix of orthonormal eigenvectors.

The absolute value of an eigenvalue indicates the significance of the corresponding eigenvector relative to the others. Accordingly, we shall use the matrix V_k with k first eigenvectors (columns) taken from the matrix V as a replacement to the matrix X. According to (10), we shall denote above model by $SVD + LOO\{lm\}$, where classification results are presented in Table 1.

It is a well known fact that principal components, or eigenvectors corresponding to the biggest eigenvalues, contain the most important information. However, the other eigenvectors, which may contain some valuable information will be ignored. In contrast, nothing will be missed by definition in the case of the GMF, because the required number of metavariables will be computed using the whole microarray matrix. The main weakness of the GMF algorithm is its dependence on the initial settings, see Table 3. By combining SVD and GMF algorithms together, we can easily overcome this weakness: it appears to be logical to use matrices V_k and U_k as an initial for the GMF algorithm, where U_k is defined according to U using the same method as in the case of V_k and V.

In the column “SVD”, Table 3, we presented classification results, which were calculated using the model $Alg.1 + LOO\{lm\}$ with initial setting produced by the SVD method. These results demonstrate some improvement compared to the “colon” column of Table 1. Similar improvements were observed in application to four remaining datasets.

4.3. Partial least squares and selection bias

Partial least squares (PLS) (Nguyen and Rocke, 2002) is an essentially different compared to GMF or to PCA, because it is a supervised method. Therefore, the evaluation scheme (10) with PLS as a first component will not be free of a selection bias. As an alternative, we can apply evaluation model

$$LOO\{PLS + lm\}, \tag{11}$$

which is about n times more expensive computationally compared to (10). According to (11), we have to exclude from the training process features of the test sample (that means we have to run PLS within any LOO evaluation loop), and this may lead to the overpessimistic results, because those features are normally available and may be used in the real life.

Very reasonably (Cawley and Talbot, 2006), the absence of the selection bias is regarded as a very significant advantage of the model, because, otherwise, we have to deal with nested cross-validation, which is regarded as an impractical (Jelizarow *et al.*, 2010, p.1996) in general terms.

Based on our own experience with the nested CV, this tool should not be used until it is absolutely necessary, because nested CV may generate secondary serious problems as a consequence of 1) the dealing with an intense computations, and 2) very complex software (and, consequently, high level of probability to make some mistakes) used for the implementation of the nested CV. Moreover, we do believe that in most of the cases scientific results produced with the nested CV are not reproducible (in the sense of an absolutely fresh data, which were not used prior). In any case, “low skills bias” could be much more damaging compared to the selection bias.

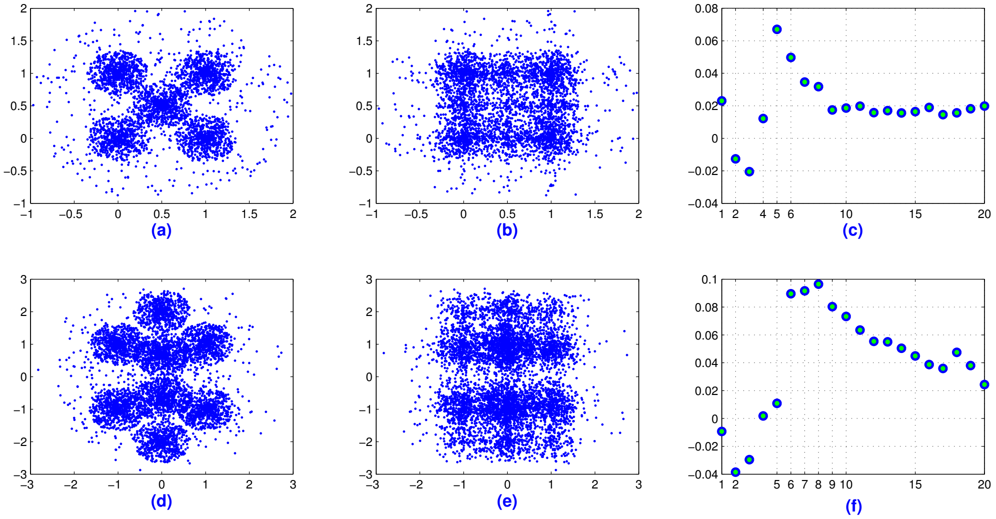

Figure 3: Illustrative examples based on two simulated datasets with known solutions (numbers of clusters) in support to the nonparametric criterion for the selection of the numbers of factors, Section 2.2. See for more details Section 4.5.

We fully agree with Jelizarow *et al.* (2010), p.1990, that the superiority of new algorithms should always be demonstrated on an independent validation data. In this sense, an importance of the data mining contests is unquestionable. The rapid popularity growth of the data mining challenges demonstrates with confidence that it is the best known way to evaluate different models and systems.

4.4. Classification results

As it was noticed in (Nikulin and McLachlan, 2009), the number of factors/metagenes must not be too large (in order to prevent overfitting), and must not be too small. In the latter case, the model will suffer because of the over-smoothing and loss of essential information as a consequence, see Figure 2(f), Tables 2 and 3 (in supplementary material).

As we reported in our previous studies (Nikulin and McLachlan, 2009, Nikulin *et al.*, 2010a), classification results observed with the model $GMF + LOO\{lm\}$ are competitive with those in (Dettling and Buhlmann, 2003, Hennig, 2007). Further improvements may be achieved using $A2GMF$ algorithm, because an adaptive learning rates make the factorization model more flexible. In addition, the results of Tables 2 and 3 (in supplementary material) may be easily improved if we shall apply instead of function lm more advanced classifier with parameter tuning depending on the particular dataset.

The numbers of factors k_{NP} and k^*, reported in Table 2, were computed using absolutely different methods, and we can see quite close correspondence between k_{NP} and k^*. In two cases (leukaemia and lymphoma), when $NM = 0$, we observed an exact correspondence.

This fact indicates that the nonparametric method of Section 2.2 is capable to discover some statistical fundamentals in microarray datasets.

Remark 5 *Additionally, we conducted extensive experiments with NMF and SVD matrix factorizations. The observed classification results were similar in terms of the numbers of misclassified. However, misclassified samples were not the same. This fact may be exploited by the suitable ensembling technique.*

4.5. Illustrative example

The goal of cluster analysis is to partition the observations into groups (clusters) so that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters (Hastie *et al.*., 2008).

Figure 3 illustrates two experiments with *kmeans* algorithm, which were conducted against simulated 2D data with known numbers of clusters, see Figure 3(a) - five clusters ($n = 10000$); Figure 3(d) - eight clusters ($n = 20000$).

As a next step we re-shuffled independently the coordinates (columns) of the data, see Figure 3(b); Figure 3(e).

Figures 3(c) and (f) illustrate behavior of

$$\widehat{\mathcal{D}}_k = \widehat{\mathcal{R}}_k - \mathcal{R}_k, \tag{12}$$

as a function of the number of the clusters k, where 1) $\widehat{\mathcal{R}}$ and 2) $\mathcal{R}$ are two averaged dissimilarity measures with squared loss functions corresponding to 1) the re-shuffled and to 2) the original data.

In both cases, we can see correct detection: Figures 3(c) - $k = 5$, and (f) - $k = 8$.

4.6. Computation time

A multiprocessor Linux workstation with speed 3.2GHz was used for most of the computations. The time for 100 global iterations (see, for example, trajectory of Figure 1) against colon set in the cases of the GMF and A2GMF algorithm was 6 and 8 sec. (expenses for the other sets were similar).

5. Concluding remarks

It seems natural to use different learning rates applied to two factor matrices. Also, the values of the learning rates must not be fixed and it is proposed to update them after any global iteration according to the given formulas. Based on our experimental results, the A2GMF algorithm presented in this paper is significantly faster. That means, less number of global iterations will be required in order to achieve the same quality of factorization. Besides, the final results have smaller value of the loss function. This fact indicates that the A2GMF algorithm is better not only technically in the sense of speed, but is better in principle: to achieve essentially better quality of factorization for the given number of factors.

The proposed criterion for the selection of the number of factors/metagenes is nonparameteric and general. By application of such criterion (as an alternative to an extensive

LOO experiments) it will be possible to save valuable computational time. Besides, we are confident that similar method is applicable elsewhere. An example with *kmeans* algorithm in given in Section 4.5.

An implementation of the third proposed novelty, a nonnegative modification of GMF, doesn't require any extra computational time. As an outcome, the NN-GMF is as fast as GMF algorithm itself. We can expect that practitioners in many fields will find this algorithm to be a competitive alternative to the well-known NMF.

Acknowledgments

This work was supported by a grant from the Australian Research Council. Also, we are grateful to three anonymous reviewers for the very helpful and constructive comments.

References

A. Cichocki, R. Zdunek, A. Phan and S. Amari. Nonnegative Matrix and Tensor Factorizations. *Wiley*, 2009.

M. Dettling and P. Buhlmann. Boosting for tumor classification with gene expression data. *Bioinformatics*, **19**(9), 1061-1069, 2003.

Q. Dong, X. Wang and L. Lin. Application of latent semantic analysis to protein remote homology detection, *Bioinformatics*, **22**, 285-290, 2006.

S. Dudoit, J. Fridlyand and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of American Statistical Association*, **97**(457), 77-87, 2002.

P. Fogel, S. Young, D. Hawkins and N. Ledirac. Inferential, robust nonnegative matrix factorization analysis of microarray data, *Bioinformatics*, **23**(1), 44-49, 2007.

Y. Gao and G. Church. Improving molecular cancer class discovery through sparse nonnegative matrix factorization, *Bioinformatics*, **21**(21), 3970-3975, 2005.

G. Cawley and N. Talbot Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics*, **22**(19), 2348-2355.

M. Jelizarow, V. Guillemot, A. Tenenhaus, K. Strimmer and A.-L. Boulesteix Over-optimism in bioinformatics: an illustration, *Bioinformatics*, **26**(16), 1990-1998.

T. Hastie, R. Tibshirani and J. Friedman. The Elements of Statistical Learning, *Springer-Verlag*, 2008.

C. Hennig. Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis*, **52**, 258-271, 2007.

Y. Koren. Collaborative filtering with temporal dynamics. *KDD, Paris*, 447-455, 2009.

D. Lee and H. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, **401**, 788-791, 1999.

Y. Li and A. Ngom. Nonnegative matrix and tensor factorization based classification of clinical microarray gene expression data. *In Proceedings of 2010 IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, 438-443, 2010.

C.-J. Lin. Projected gradient method for nonnegative matrix factorization. *Neural Computation*, **19**, 2756-2779, 2007.

D. Nguyen and D. Rocke Tumor classification by partial least squares using microarray gene expression data, *Bioinformatics*, **18**(1), 39-50.

V. Nikulin and G. J. McLachlan. Classification of imbalanced marketing data with balanced random sets. *JMLR: Workshop and Conference Proceedings*, 7, pp. 89-100, 2009.

V. Nikulin and G. J. McLachlan. On a general method for matrix factorization applied to supervised classification. *Proceedings of the 2009 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, Washington D.C., 44-49, 2009.

V. Nikulin and G. J. McLachlan. On the gradient-based algorithm for matrix factorisation applied to dimensionality reduction. *Proceedings of BIOINFORMATICS 2010, Edited by Ana Fred, Joaquim Filipe and Hugo Gamboa*, Valencia, Spain, 147-152, 2010.

V. Nikulin, T.-H. Huang and G.J. McLachlan. A Comparative Study of Two Matrix Factorization Methods Applied to the Classification of Gene Expression Data. *In Proceedings of 2010 IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, 618-621, 2010.

V. Nikulin, T.-H. Huangb, S.-K. Ng, S. Rathnayake, G.J. McLachlan. A very fast algorithm for matrix factorization. *Statistics and Probability Letters*, **81**, 773-782, 2011.

E. Oja, A. Ilin, J. Luttinen and Z. Yang. Linear expansions with nonlinear cost functions: modelling, representation, and partitioning. *Plenary and Invited Lectures, WCCI 2010, Edited by Joan Aranda and Sebastia Xambo*, Barcelona, Spain, 105-123, 2010.

P. Tamayo et al. Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences USA*, **104**(14) 5959-5964, 2007.

L. Wasserman. All of Nonparametric Statistics. *Springer*, 2006.