Tensor Decompositions for Probabilistic Classification

Marcel van Gerven
Department of Knowledge and Information Systems
Radboud University Nijmegen
Toernooiveld 1, 6525 ED, Nijmegen, The Netherlands
marcelge@cs.ru.nl

Abstract

Tensor decompositions are introduced as a novel approach to probabilistic classification and can be interpreted as a particular kind of mixture model. Since many problems in medicine and biology can be described as a classification problem, the approach is seen as a useful tool for biomedical data analysis. The approach is validated by means of a clinical database consisting of data about 1002 patients that suffer from hepatic disease. It is shown that the approach performs comparably to state-of-the-art results that have been obtained using a naive Bayes classifier.

1 Introduction

Classification is an important concept in current medical practice. The (differential) diagnosis of disease, the selection of appropriate treatment, and the prediction of patient survival can all be cast in a framework that selects the correct class from a set of possible classes given observed patient data. In case of probabilistic classification, each class has an associated posterior probability that represents the belief in that particular class.

In this paper, we present a novel probabilistic classification technique which is based on the decomposition of a multiway array, also known as a *tensor* [1]. Components of the decomposition are given by a set of vectors that allow for a compact representation of the original tensor. We call classifiers that use this technique *decomposed tensor classifiers*, and test their performance by means of a database that contains data about 1002 patients that present with hepatic disease. The goal is to diagnose the correct disease for each of the patients from a set of four distinct diseases. Classification performance of the technique is analyzed and compared with the performance of the naive Bayes classifier [2].

We proceed as follows. In Sections 2, 3 and 4 the theoretical background of tensors, tensor decompositions, and their interpretation in terms of mixture models is described. Subsequently, in Section 5, we address how tensor decompositions can be used for probabilistic classification. The clinical database and the techniques used to evaluate classification performance are described in Section 6. We analyse the experimental results in Section 7 and we end with some concluding remarks in Section 8.

2 Tensors

A tensor is a concept taken from multilinear algebra which generalizes the concepts of vectors and matrices, and is defined as follows.

Definition 1. *Let $I_1, \ldots, I_N \in \mathbb{N}$ denote index upper bounds. A* tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *is an N-way array where elements $a_{i_1 \cdots i_n}$ are indexed by $i_j \in \{1, \ldots, I_j\}$ for $1 \leq j \leq N$.*

We call N the *order* of a tensor, such that a tensor of order one denotes a vector $\mathbf{a} \in \mathbb{R}^{I_1}$, and a tensor of order two denotes a matrix $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$. The nth *mode* of a tensor refers to the nth dimension of a tensor. A tensor can be expressed in terms of a matrix using the concept of a matrix unfolding.

Definition 2. *Given an Nth order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, the* matrix unfolding $\mathbf{A}_{(j)} \in \mathbb{R}^{I_j \times (I_{j+1} I_{j+2} \cdots I_N I_1 I_2 \cdots I_{j-1})}$ *of $\mathcal{A}$ is the matrix that has element $a_{i_1 \cdots i_N}$ at row number i_j and column number*

$$1 + \sum_{\substack{1 \leq k \leq N \\ k \neq j}} (i_k - 1) \prod_{\substack{k+1 \leq m \leq N \\ m \neq j}} I_m \,.$$

Example 1. The matrix unfolding $\mathbf{A}_{(2)}$ of a third-order tensor

$$\mathcal{A} = \begin{pmatrix} (a,b)^T & (c,d)^T \\ (e,f)^T & (g,h)^T \end{pmatrix}$$

is given by

$$\mathbf{A}_{(2)} = \begin{pmatrix} a & b & e & f \\ c & d & g & h \end{pmatrix} .$$

A tensor may be multiplied by a matrix by means of the *n-mode product*.

Definition 3. *The* n-mode product $\mathcal{A} \times_n \mathbf{B}$ *of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a matrix $\mathbf{B} \in \mathbb{R}^{J_N \times I_N}$, is a tensor $\mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ with elements:*

$$c_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} a_{i_1 \cdots i_N} b_{j_n i_n} \,.$$

Example 2. Let $\mathcal{A}$ be a third-order tensor as in example 1 and let $\mathbf{B}$ denote a square matrix with $b_{11} = u$, $b_{12} = v$, $b_{21} = w$, $b_{22} = x$. The 2-mode product $\mathcal{A} \times_2 \mathbf{B}$ is then given by

$$\begin{pmatrix} (a(u+v), b(u+v))^T & (c(w+x), d(w+x))^T \\ (e(u+v), f(u+v))^T & (g(w+x), h(w+x))^T \end{pmatrix} .$$

We define for tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, the *inner product*

$$\langle \mathcal{A}, \mathcal{B} \rangle \equiv \sum_{i_1,\ldots,i_N} a_{i_1 \cdots i_N} b_{i_1 \cdots i_N}$$

and *Frobenius norm* $||\mathcal{A}|| \equiv \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The *outer product* $\mathcal{A} \circ \mathcal{B}$ of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_n}$ is defined as the tensor $\mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times J_1 \times \cdots \times J_n}$ such that $c_{i_1 \cdots i_n j_1 \cdots j_n} = a_{i_1 \cdots i_n} \cdot b_{j_1 \cdots j_n}$ for all elements of $\mathcal{C}$. The rank of a tensor is then defined as follows [3].

Definition 4. *A tensor of order N has* rank *one if it can be written as an outer product $\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)}$ of vectors. The rank of a tensor $\mathcal{A}$ is defined as the minimal number of tensors $\mathcal{A}_1, \ldots, \mathcal{A}_K$ of rank one such that*

$$\mathcal{A} = \sum_{k=1}^{K} \mathcal{A}_k \,. \tag{1}$$

Example 3. The third-order tensor

$$\mathcal{A} = \begin{pmatrix} (\ \ 6, -3)^T & (\ \ 8, -4)^T \\ (-12, \ \ 6)^T & (-16, \ \ 8)^T \end{pmatrix}$$

has rank one since it can be written as the outer product of vectors $(1, -2)^T$, $(3, 4)^T$, and $(2, -1)^T$.

3 Tensor Decompositions

Equation (1) is known as a *rank-K decomposition* of $\mathcal{A}$. A more general kind of decomposition is the *Tucker decomposition* [4], which can be interpreted as a multilinear formulation of the singular value decomposition [5]:

$$T_{\mathbf{J}}(\mathcal{A}) = \mathcal{C} \times_1 \mathbf{B}^{(1)} \times_2 \cdots \times_N \mathbf{B}^{(N)} \tag{2}$$

with $\mathbf{J} = (J_1, \ldots, J_N)$, *core tensor* $\mathcal{C} = (c_{j_1 \cdots j_N})$ and matrices $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times J_n}$. Elements of $\mathcal{A}$ are then computed as follows:

$$a_{i_1 \cdots i_N} = \left(\sum_{j_1,\ldots,j_N} c_{j_1 \cdots j_N} \cdot b^{(1)}_{i_1 j_1} \cdots b^{(N)}_{i_N j_N} \right) + r_{i_1 \cdots i_N} \,, \tag{3}$$

where $(r_{i_1 \cdots i_N})$ denotes a residual tensor $\mathcal{R}$. A special case of the Tucker decomposition is obtained when one assumes that the core tensor $\mathcal{C}$ is a superdiagonal tensor with $c_{j_1 \cdots j_N} = 0$ if there are $u, v \in \{1, \ldots, N\}$ such that $j_u \neq j_v$, and $c_{j_1 \cdots j_N} = 1$ otherwise. Hence, we obtain:

$$a_{i_1 \cdots i_N} = \left(\sum_{k=1}^{K} \lambda_k \cdot b^{(1)}_{i_1 k} \cdots b^{(N)}_{i_N k} \right) + r_{i_1 \cdots i_N} \tag{4}$$

for some suitably chosen K. Equation (4) is known as the *canonical decomposition* [6], or *parallel factors decomposition* [7]. In general, the decomposition of (4) is not necessarily minimal nor exact, and can be interpreted as a sum of rank-1 approximations. One way of finding a rank-1 approximation is by means of the *higher-order power method* (HOPM) [8], as shown in Algorithm 1.

The higher-order power method finds a tensor $\hat{\mathcal{A}} = \lambda \cdot \mathbf{b}^{(1)} \circ \cdots \circ \mathbf{b}^{(N)}$, with scalar λ and unit-norm vectors $\mathbf{b}^{(n)}$, $1 \leq n \leq N$, that minimizes the least-squares cost function $C(\mathcal{A}, \hat{\mathcal{A}}) \equiv ||\mathcal{A} - \hat{\mathcal{A}}||^2$. A greedy approach to finding

input: $\mathcal{A}$
initialize $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(N)}$
repeat
  for $n = 1$ to N **do**
    $\tilde{\mathbf{b}}^{(n)} = \mathcal{A} \times_1 \mathbf{b}^{(1)^T} \times_2 \cdots \times_{n-1} \mathbf{b}^{(n-1)^T} \times_{n+1} \mathbf{b}^{(n+1)^T} \times_{n+2} \cdots \times_N \mathbf{b}^{(N)^T}$
    $\lambda_n = ||\tilde{\mathbf{b}}^{(n)}||$
    $\mathbf{b}^{(n)} = \tilde{\mathbf{b}}^{(n)} / \lambda_n$
  end for
until convergence
return $\hat{\mathcal{A}} = \lambda_N \cdot \mathbf{b}^{(1)} \circ \cdots \circ \mathbf{b}^{(N)}$

Algorithm 1: Higher-Order Power Method (HOPM).

the sum of rank-1 terms in (4) is to apply the higher-order power method to the residuals that remain after obtaining a rank-1 approximation; a technique which has been employed successfully in order to achieve high compression rates for image sequences [9]. By defining $\mathcal{A}^1 \equiv \mathcal{A}$ and $\mathcal{A}^k \equiv \mathcal{A}^{k-1} - \text{HOPM}(\mathcal{A}^{k-1})$ the following rank-K approximation of a tensor $\mathcal{A}$ is obtained:

$$R_K(\mathcal{A}) \equiv \sum_{k=1}^{K} \text{HOPM}(\mathcal{A}^k) \,. \tag{5}$$

In order to initialize matrices and vectors in Algorithm 1, various schemes can be used. One approach is to repeat the algorithm for several random initializations and to choose that decomposition which maximizes the fit between the original tensor and the approximation. Another approach, which has proven to work well in practice, is to choose the first dominant left singular vector of the matrix unfolding $\mathbf{A}_{(j)}$, as an initial estimate of $\mathbf{b}_j$ [8; 5]. The algorithm has converged when the increase in fit between the tensor and its approximation that is gained after one iteration drops below a small error criterion ϵ. In the following section, we will use decompositions of tensors $\mathcal{A} \in [0, 1]^{I_1 \times \cdots \times I_N}$ for the task of probabilistic classification.

4 Mixture Model Interpretation

As noted by [10; 11], we may interpret a rank-K approximation in terms of a mixture model. According to Eq. (4), the rank-K approximation of ψ can be written as:

$$R_K(\psi)_{x_1 \cdots x_N} = \sum_{h=1}^{K} \lambda_h \cdot b^{(1)}_{x_1 h} \cdots b^{(N)}_{x_N k} \,. \tag{6}$$

By defining functions $\phi_j(x_j, h) \equiv b^{(j)}_{x_j h}$ for $1 \leq j < n$ and absorbing λ into the function $\phi_n(x_N, h) \equiv \lambda_h \cdot b^{(N)}_{x_N h}$, we obtain:

$$\psi(x_1, \ldots, x_N) \approx \sum_h \prod_{j=1}^{N} \phi_j(x_j, h) \,, \tag{7}$$

which can be interpreted as marginalization over a hidden variable H with states h, as shown in Fig. 1. Note that, in case the decomposition (7) uses just one component, it reduces to:

$$\psi(x_1, \ldots, x_N) \approx \prod_{j=1}^{N} \phi_j(x_j) \,, \tag{8}$$

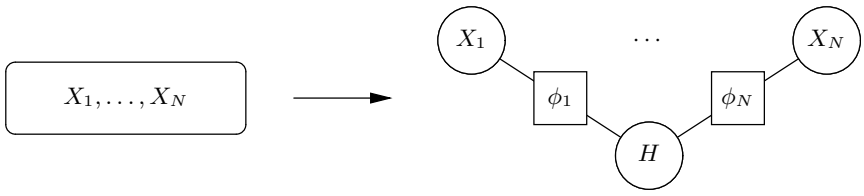

Figure 1: A function $\psi(x_1, \ldots, x_n)$ can be represented by a tensor rank-K approximation. This can be interpreted in terms of a mixture model, with real-valued functions ϕ_j and a hidden variable H taking values $h \in \{1, \ldots, K\}$.

which implies independence between X_i and X_j with $i, j \in \{1, \ldots, N\}, i \neq j$.

5 Classification with Tensor Decompositions

In this section, we focus on a multiset $\mathbf{A} = \{\mathbf{a}^1, \ldots, \mathbf{a}^n\}$ that represents our data, and where an instance $\mathbf{a}^i = (x_1^i, \ldots, x_N^i)$ consists of evidence $(x_1^i, \ldots, x_{N-1}^i)$ and a class label x_N^i. We assume that all variables are discrete and use I_j with $1 \leq j \leq N$ to denote the finite number of values x_j of a variable X_j. The basic idea is to obtain an approximation of a *incomplete* tensor $\mathcal{A}$ using a tensor decomposition. Let $\mathbf{x}$ denote the evidence and let $n(\mathbf{x}, x_N)$ stand for the number of times $(\mathbf{x}, x_N)$ occurs in $\mathbf{A}$. We transform $\mathbf{A}$ into an incompletely specified tensor $\mathcal{A} \in [0, 1]^{I_1 \times \cdots \times I_N}$, such that

$$a_{x_1 \cdots x_N} = \frac{1}{n} n(\mathbf{x}, x_N) \tag{9}$$

for all $(\mathbf{x}, x_N)$ for which some $(\mathbf{x}, j)$ with $1 \leq j \leq I_N$ occurs in $\mathbf{A}$. Hence, $a_{x_1 \cdots x_N}$ is undefined for unseen evidence $\mathbf{x}$ (as indicated by $*$), which implies that the tensor is incomplete. The element $a_{x_1 \cdots x_N}$ is used to represent an estimate of the joint probability $P(\mathbf{x}, x_N)$. For incomplete tensors, we interpret undefined elements as zero in Algorithm 1. Since zero elements have no contribution, we may use a sparse representation of tensors $\mathcal{A} \in [0, 1]^{I_1 \times \cdots \times I_N}$, where N may be large, provided that only some of the elements are defined.

Example 4. Consider a dataset $\mathbf{A} = \{(1, 1, 1), (1, 2, 1), (1, 2, 1), (1, 2, 2), (2, 1, 2), (1, 1, 2)\}$. By applying the transformation (9) to the example dataset, we obtain

$$\mathcal{A} = \begin{pmatrix} (\frac{1}{6}, \frac{1}{6})^T & (\frac{2}{6}, \frac{1}{6})^T \\ (\frac{0}{6}, \frac{1}{6})^T & (*, *)^T \end{pmatrix}.$$

In case of probabilistic classification, our interest is in computing $P(x_N \mid \mathbf{x})$ based on our estimate of $P(\mathbf{x}, x_N)$. Although $P(\mathbf{x}, x_N)$ is approximated by $R_K(\mathcal{A})_{x_1 \cdots x_N}$, we have no guarantee that the tensor approximation represents a proper probability distribution for unseen evidence (which is the goal of probabilistic classification), since the approximation may be unnormalized or even lying outside the unit interval. Therefore, we use the following transform when computing the conditional probability of X_N given $\mathbf{x}$:

$$P(x_N \mid \mathbf{x}) = \frac{R_K^+(\mathcal{A})_{x_1 \cdots x_N}}{\sum_{1 \leq j \leq I_N} R_K^+(\mathcal{A})_{x_1 \cdots x_{N-1} j}} \tag{10}$$

where $R_K^+(\mathcal{A})_{x_1 \cdots x_N}$ is defined as

$$R_K(\mathcal{A})_{x_1 \cdots x_N} - \min\left\{0, \min_j(R_K(\mathcal{A})_{x_1 \cdots x_{N-1}, j})\right\},$$

which ensures that we sum over positive terms by making (small) negative terms non-negative. Alternatively, a log transform along with a suitable prior may be used in order to guarantee that we obtain a proper conditional probability distribution. However, initial experiments in this direction led to less optimal classification results.

We use the term *decomposed tensor classifier* (DTC) to denote a classifier that uses the approximation $R_K(\mathcal{A})_{x_1 \cdots x_N}$ for the purpose of classification, as shown in Algorithm 2. In this paper, we use the rank-K approximation, although other tensor decompositions such as the Tucker decomposition could also be used. Furthermore, we require that variables are discrete and data is complete.

input: $\mathbf{A}_{\text{train}}$, $\mathbf{A}_{\text{test}}$, K
transform the dataset $\mathbf{A}_{\text{train}}$ into the tensor $\mathcal{A}_{\text{train}}$ using (9)
learn the approximation $R_K(\mathcal{A}_{\text{train}})$ using Algorithm 1
for all rows $(\mathbf{x}) \in \mathbf{A}_{\text{test}}$ **do**
  for $j = 1$ **to** I_N **do**
    compute $P(j \mid \mathbf{x})$ using (10)
  end for
  assign class label $\mathcal{L}(\mathbf{x}) = \arg\max_j\{P(j \mid \mathbf{x})\}$
end for
return class labels $\mathcal{L}$

Algorithm 2: Decomposed tensor classification.

6 Classifier Evaluation

In order to examine the performance of decomposed tensor classifiers, we have made use of the COMIK dataset, which was collected by the Copenhagen Computer Icterus (COMIK) group and consists of data on 1002 jaundiced patients that may be classified into one of four diagnostic categories: *acute non-obstructive*, *chronic non-obstructive*, *benign obstructive* and *malignant obstructive* given 21 evidence variables [12]. Earlier classification studies have shown that, typically, the correct diagnostic conclusion (in accordance with the diagnostic conclusion of expert clinicians) is found for about $75 - 77\%$ of jaundiced patients [13; 14]. As a preprocessing step, we have computed the mutual information between evidence variables and the class variable, and selected the eighteen evidence variables that show highest mutual information (MI) with the class variable as the basis for classification, since the three remaining evidence variables give relatively small contributions (Fig. 3).

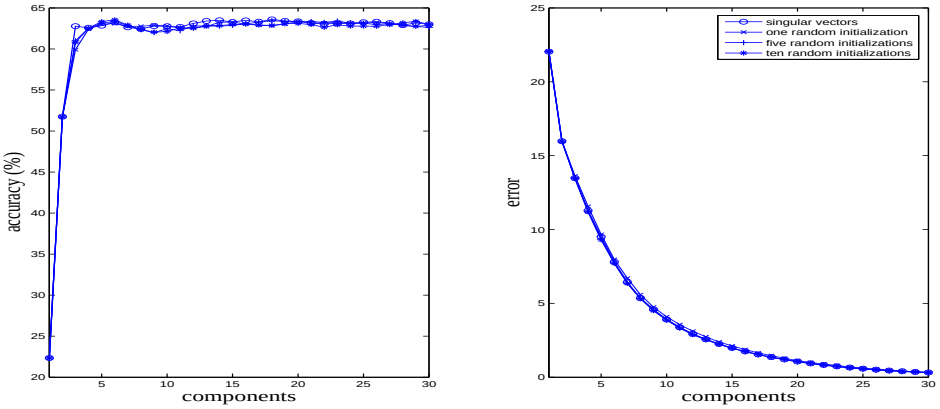

Figure 2: Average classification accuracy (left) and least squares error of the tensor approximation (right) based on five evidence variables with different initializations.

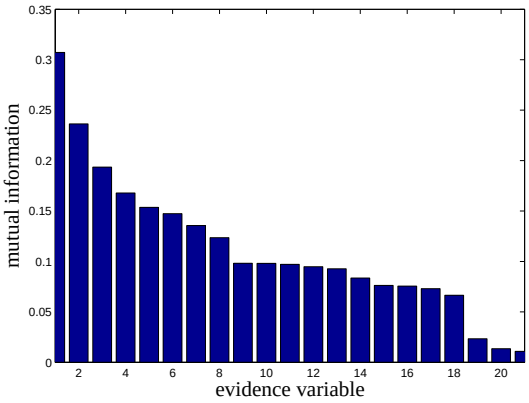

Figure 3: Mutual information between the class variable and evidence variables.

Classification performance of the decomposed tensor classifiers is compared with that of a naive Bayes classifier using a ten-fold cross-validation scheme. Empirical estimates of the required probabilities are smoothed using Laplace smoothing. The naive Bayes classifier typically reaches high classification accuracies, and uses the (naive) assumption that evidence variables are independent given the class label, such that:

$$P(x_N \mid \mathbf{x}) \propto P(x_N) \prod_{j=1}^{N-1} P(x_j \mid x_N)\,.$$

Since the COMIK dataset contains missing values, and the decomposed tensor classifiers require complete data, we have used multiple imputation to create three complete datasets from the incomplete dataset. Since we have no knowledge about the missing data mechanism, we make the (admittedly unrealistic) assumption that data is missing completely at random, and use the prior probabilities of the evidence variables to determine the imputed values. This allows a comparison in terms of classification performance between the naive Bayes classifier and the decomposed tensor classifiers, where the performance is averaged over the ten folds and over the three complete datasets.

Classification performance is quantified by means of *classification accuracy* and *logarithmic score*. For a dataset consisting of m cases $(\mathbf{x}^i, x_N^i)$ where $\mathbf{x}^i$ denotes the evidence and x_N^i the class value for the ith case, the classification accuracy is defined as the percentage of correctly classified cases:

$$\frac{1}{m}\sum_{i=1}^{m} 1_{\arg\max_c\{P(X_N=c|\mathbf{x}^i)\}=x_N^i} \times 100\%,$$

where 1_X is the indicator function, which gives 1 if X is true and 0 if X is false. The logarithmic score [15] is a scoring rule which penalizes a probability model based on a database consisting of m instances $(\mathbf{x}^i, x_N^i)$ where $\mathbf{x}^i$ denotes the evidence and x_N^i denotes the class value. Assuming that instances are independently sampled and identically distributed, the logarithmic score is defined as:

$$S = -\sum_{i=1}^{m} \log P(x_N^i \mid \mathbf{x}^i)$$

which incurs a penalty if a low probability is assigned to events that actually occur. The logarithmic score of the decomposed tensor classifier is compared with that of the naive Bayes classifier in order to determine how well actual posterior probabilities are approximated.

7 Experimental Results

In order to use the rank-K approximation for classification, the first question is which initialization procedure to use in Algorithm 1. Therefore, we have conducted a preliminary experiment in order to compare different initialization schemes in terms of classification accuracy and least squares error. To this end, we have chosen the five most informative evidence variables as the basis for classification, and compared the performance on the test set of classifiers R_K, with $1 \leq K \leq 30$, for 1, 5, and 10 random initializations, and for the initialization with dominant left singular vectors, as described in Section 3. Figure 2 shows

the results, which indicate that there is not much difference in classification accuracy or least squares error for the different initialization schemes. Differences in standard deviations were also negligible (not shown). Therefore, we have chosen to use just one random initialization since this uses the fewest computational resources. Based on this initialization procedure, we have learnt decomposed tensor classifiers based on the eighteen most informative evidence variables for $1 \leq K \leq 30$ components.

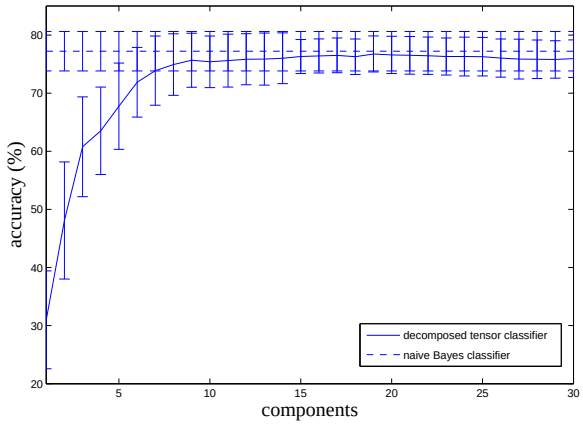

Figure 4: Average classification accuracy and standard deviations on the test set for the decomposed tensor classifier and the naive Bayes classifier.

The comparison of the classification accuracy of the decomposed tensor classifier with that of the naive Bayes classifier is shown in Fig. 4. The highest average accuracy for the decomposed tensor classifier is reached at nineteen components with an accuracy of 76.75%, whereas for the naive Bayes classifier, the average classification accuracy is 77.25%. At that point, the standard deviation of the classification accuracy of the decomposed threshold classifier is 3.24%, whereas that of the naive Bayes classifier is 3.40%. Although the naive Bayes classifier performs somewhat better than the decomposed tensor classifier in terms of classification accuracy, differences are negligible. Figure 5 depicts the average logarithmic scores for the decomposed tensor classifier and the naive Bayes classifier (where we have added a small term to (10) in order to prevent numerical problems). It shows that the logarithmic score of the decomposed tensor classifier decreases as more components are added and eventually becomes lower than that of the naive Bayes classifier. Note that, in practice, the appropriate number of components is selected by means of cross-validation on a hold-out set.

Figure 7 shows a Hinton diagram, depicting the contribution of each component for each of the four classes for a decomposed tensor classifier containing nineteen components. The large white block that can be found in each column indicates that each of the components improves the approximation by focusing mainly on one class. For the decomposed tensor classifier, the transform of (10) assigns distributions skewed towards zero for incorrect classes and skewed towards one for the correct class, although not as

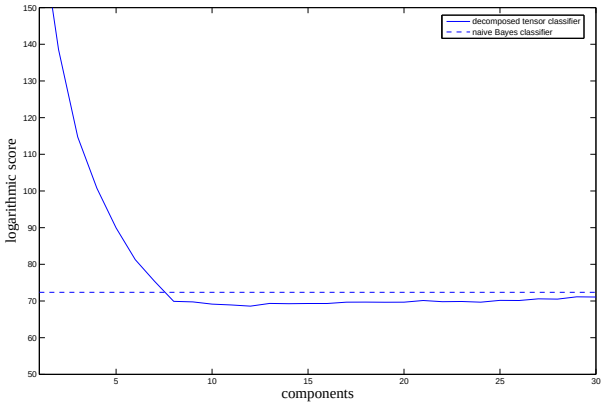

Figure 5: Average logarithmic score on the test set for the decomposed tensor classifier and the naive Bayes classifier.

well as the naive Bayes classifier, as shown in Fig. 6.

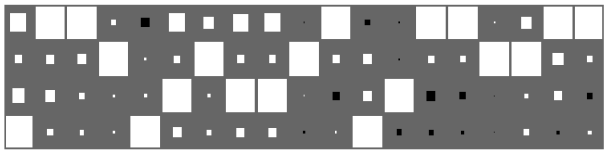

Figure 7: Hinton diagram, showing the magnitude of positive contributions (white blocks) and negative contributions (black blocks) of nineteen rank-1 components (horizontal axis) for the four classes (vertical axis).

Although the naive Bayes classifier and the decomposed tensor classifier operate differently, they perform comparably with respect to classification accuracy. If we inspect the classifications that were made by the classifiers then it is interesting to see that only 254 out of a total of 2955 cases (8.60%) have been classified differently by the two classifiers. Out of these 254 cases, the naive Bayes classifier assigned 107 cases to the correct class, whereas the decomposed tensor classifier assigned 93 cases to the correct class. Hence, the classifiers are able to classify different cases correctly, suggesting that there are certain problems for which the naive Bayes classifier is more suitable, and other problems for which the decomposed tensor classifier is more suitable.

8 Conclusion

In this paper, we have shown that tensor decompositions can be used for the purpose of probabilistic classification. The classification performance of this novel classification method on a problem in medical diagnosis is comparable to that of the naive Bayes classifier and other methods which have been specifically developed to solve this classification problem. The method is less suitable for obtaining accurate posterior probabilities (as is evident from Fig. 6), but the different mode of operation, together with the results concerning correctly classified cases, suggest that there may be particular problems for which this new technique performs

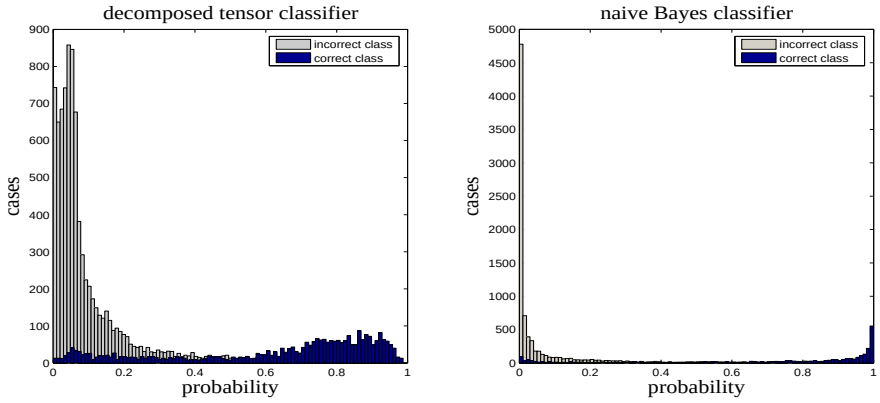

Figure 6: Distribution of posterior probabilities of correct and incorrect classes for the decomposed tensor classifier and the naive Bayes classifier.

better than the naive Bayes classifier. Current limitations of the technique are the requirements that data is discrete and complete, and the fact that learning the classifiers requires more computational resources than the (easy to learn) naive Bayes classifier.

In this paper, we have focused on the use of rank-K approximations as the basis for decomposed tensor classification. The more general Tucker decomposition may also be used for this purpose, and can be learnt using *higher-order orthogonal iteration* [8]. Preliminary results suggest that this is possible, albeit much harder, since we are now required to search for the optimal sizes of matrices $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, $1 \leq n \leq N$, as shown in (2).

Decomposed tensor classifiers are a new way of employing tensor decompositions, the usefulness of which was demonstrated in this research using a classification problem in medical diagnosis. Dealing with current limitations, validation of the technique by means other datasets, and an analysis of the use of Tucker decompositions for probabilistic classification, are directions for future research.

References

[1] de Lathauwer, L.: Signal Processing Based on Multilinear Algebra. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium (1997)

[2] Maron, M.E.: Automatic indexing: An experimental inquiry. Journal of the ACM **8**(3) (1961) 404–417

[3] Håstad, J.: Tensor rank is NP-complete. Journal of Algorithms **11** (1990) 644–654

[4] Tucker, L.R.: Some mathematical notes of three-mode factor analysis. Psychometrika **31** (1966) 279–311

[5] de Lathauwer, L., de Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J Matrix Anal Appl **21** (2000) 1253–1278

[6] Carroll, J.D., Chang, J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika **35** (1970) 283–319

[7] Harshman, R.A.: Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis. UCLA Working Papers in Phonetics **16** (1970) 1–84

[8] de Lathauwer, L., de Moor, B., Vandewalle, J.: On the best rank-1 and rank-($R_1, R_2, \ldots, R_N$) approximation of higher-order tensors. SIAM J Matrix Anal Appl **21**(4) (2000) 1324–1342

[9] Wang, H., Ahuja, N.: Compact representation of multidimensional data using tensor rank-one decomposition. In: International Conference on Pattern Recognition, IEEE (2004) 44–47

[10] Savický, P., Vomlel, J.: Tensor rank-one decomposition of probability tables. Technical Report DAR-UTIA 2005/26, Institute of Information Theory and Automation, Prague, Czech Republic (2005)

[11] Shashua, A., Hazan, T.: Non-negative tensor decompositions with applications to statistics and computer vision. In: Proceedings of the 22nd International Conference on Machine Learning. Volume 119 of ACM International Conference Proceeding Series., New York, NY, ACM Press (2005) 792–799

[12] Malchow-Møller, A., Thomson, C., Matzen, P., Mindeholm, L., Bjerregaard, B., Bryant, S., Hilden, J., Holst-Christensen, J., Johansen, T.S., Juhl, E.: Computer diagnosis in jaundice: Bayes' rule founded on 1002 consecutive cases. J Hepatol **3** (1986) 154–163

[13] Lindberg, G., Thomson, C., Malchow-Møller, A., Matzen, P., Hilden, J.: Differential diagnosis of jaundice: applicability of the Copenhagen pocket diagnostic chart proven in Stockholm patients. Liver **7** (1987) 43–49

[14] van Gerven, M.A.J., Lucas, P.J.F.: Employing maximum mutual information for Bayesian classification. In: Biological and Medical Data Analysis. Volume 3337 of Lecture Notes in Computer Science. Springer, Berlin, Germany (2004) 188–199

[15] Spiegelhalter, D.J., Dawid, A.P., Lauritzen, S.L., Cowell, R.G.: Bayesian analysis in expert systems. Stat Sci **8**(3) (1993) 219–283