The Minimum Transfer Cost Principle for Model-Order Selection

Mario Frank, Morteza Haghir Chehreghani, and Joachim M. Buhmann

Department of Computer Science, ETH Zurich, Switzerland

Abstract. The goal of model-order selection is to select a model variant that generalizes best from training data to unseen test data. In unsupervised learning without any labels, the computation of the generalization error of a solution poses a conceptual problem which we address in this paper. We formulate the principle of "*minimum transfer costs*" for model-order selection. This principle renders the concept of cross-validation applicable to unsupervised learning problems. As a substitute for labels, we introduce a mapping between objects of the training set to objects of the test set enabling the transfer of training solutions. Our method is explained and investigated by applying it to well-known problems such as singular-value decomposition, correlation clustering, Gaussian mixture-models, and k-means clustering. Our principle finds the optimal model complexity in controlled experiments and in real-world problems such as image denoising, role mining and detection of misconfigurations in access-control data.

Keywords: clustering, generalization error, transfer costs, cross-validation.

1 Introduction

Clustering and dimensionality reduction are highly valuable concepts for exploratory data analysis that are frequently used in many applications for pattern-recognition, vision, data mining, and other fields. Both problem domains require to specify the complexity of solutions. When partitioning a set of objects into clusters, we must select an appropriate number of clusters. Learning a low-dimensional representation of a set of objects, for example by learning a dictionary, involves choosing the number of atoms or codewords in the dictionary. More generally speaking, learning the parameters of a model given some measurements requires selecting the number of parameters, i.e. one must select the model-order.

In this paper we address the general issue of model-order selection for unsupervised learning problems and we develop and advocate the principle of *minimal transfer costs* (MTC). Our method generalizes classical cross-validation known from supervised learning. It is applicable to a broad class of model-order selection problems even when no labels or target values are given. In essence, MTC can be applied whenever a cost function is defined. The MTC principle can be easily explained in abstract terms: A good choice of the model-order based on a given dataset should also yield low costs on a second dataset from the same distribution. We learn models of various model-orders from a given dataset $\mathbf{X}^{(1)}$. These models with their respective parameters are then used to interpret a second data set $\mathbf{X}^{(2)}$, i.e., to compute its costs. The principle selects the

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 423–438, 2011.

model-order that achieves lowest transfer cost, i.e. the solution that generalizes best to the second dataset. Too simple models underfit and achieve high costs on both datasets; too complex models overfit to the fluctuations of $\mathbf{X}^{(1)}$ which results in high costs on $\mathbf{X}^{(2)}$ where the fluctuations are different.

The conceptually challenging part of this procedure is related to the transfer of the solution inferred from the objects of the first dataset to the objects of the second dataset. This transfer requires a mapping function which generalizes the conceptually straightforward assignments in supervised learning. For several applications, we demonstrate how to map two datasets to each other when no labels are given.

Our main contribution is to propose and describe the minimum transfer cost principle (MTC) and to demonstrate its broad applicability on a set of different applications. We select well-known methods such as singular-value decomposition (SVD), max. likelihood inference, k-means, Gaussian mixture models, and correlation clustering because the understandability of our principle should not be limited by long explanations of the complicated models it is applied to. In our real-world applications *image denoising*, *role mining*, and *error detection in access-control configurations* we pursue the goal to investigate the reliability of the model order selection scheme, i.e. whether for a predetermined method (such as SVD), our principle finds the model-order that performs best on a second test data set.

In the remainder of the paper we first explain the principle of minimal transfer costs and we address the conceptual question of how to map a trained model to a previously unseen dataset. In the following Sections 3, 6, 7 we invoke the MTC principle to select a plausible ("true") number of centroids for the widely used Gaussian mixture model, the optimal number of clusters for correlation clustering and for the k-means algorithm. In Section 4, we apply MTC to SVD for image denoising and detecting errors in access-control configurations. In Section 5, we use MTC for selecting the number of factors for Boolean matrix factorization on role mining data.

2 Minimum Transfer Costs

2.1 Notational Preliminaries

Let O be a set of N objects with corresponding measurements. The measurements can be characterized in several ways: (i) objects can be identified with the measurements and we can use the terms synonymously, i.e., the i^{th} object is described by the vector $\mathbf{x}_i$; (ii) measurements are pairwise (dis)similarities between objects. In the first case, the objects O are directly characterized by the measurements $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathcal{X}$. In the second case, a graph $\mathcal{G}(O, \mathbf{X})$ with (dis)similarity measurements $\mathbf{X} := \{X_{ij}\}$ characterizes the relations for all pairs of objects (i, j), $1 \leq i \leq N, 1 \leq j \leq N$. Furthermore, let $\{O^{(1)}, \mathbf{X}^{(1)}\}$ and $\{O^{(2)}, \mathbf{X}^{(2)}\}$ be two datasets given from a unique source. Often in practical situations, only one such dataset is available. Then we randomly partition it into $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

A data model is usually characterized as an optimization problem with an associated *cost function*. We denote the potential outcome of optimizing a cost function by the term *solution*. A cost function $R(\boldsymbol{s}, \mathbf{X}, k)$ quantifies how well a particular solution $\boldsymbol{s} \in \mathcal{S}$ explains the measurements $\mathbf{X}$. For parametric models, the solution includes a set of

model parameters which are learned through an inference procedure. The number k quantifies the number of model parameters and thereby identifies the model order. In clustering, for instance, k would be the number of clusters of the solution $\boldsymbol{s}(\mathbf{X})$.

2.2 Minimum Transfer Costs

A cost functions imposes a partial order on all possible solutions given the data. Since usually the measurements are contaminated by noise, one aims at finding solutions that are robust against the noise fluctuations and thus generalize well to future data. Learning theory demands that a well-regularized model explains not only the dataset at hand, but also new datasets generated from the same source and thus drawn from the same probability distribution.

Let $\boldsymbol{s}^{(1)}$ be the solution (e.g. model parameters) learned from a given set of objects $O^{(1)} = \{i : 1 \leq i \leq N_1\}$ and the corresponding measurements $\mathbf{X}^{(1)}$. Let the set $O^{(2)} = \{i' : 1 \leq i' \leq N_2\}$ represent the objects of a second dataset $\mathbf{X}^{(2)}$ drawn from the same distribution as $\mathbf{X}^{(1)}$. In a supervised learning scenario, the given class labels of both datasets guide a natural and straightforward mapping of the trained solution from the first to the second dataset: the model should assign objects of both sets with same labels to the same classes. However, when no labels are available, it is unclear how to transfer a solution. To enable the use of cross-validation, we propose to compute the costs of a learned solution on a new dataset in the following way. We start with defining a mapping ψ from objects of the second dataset to objects of the first dataset:

$$\psi : \; O^{(2)} \times \mathcal{X} \times \mathcal{X} \to O^{(1)} \; , \quad \left(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right) \mapsto \psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \tag{1}$$

This mapping function aligns each object from the second dataset with its nearest neighbor in $O^{(1)}$. We have to compute such a mapping in order to transfer a solution. Let's assume, for the moment, that the given model is a sum over independent partial costs

$$R(\boldsymbol{s}, \mathbf{X}, k) = \sum_{i=1}^{N} R_i(\boldsymbol{s}(i), \mathbf{x}_i, k). \tag{2}$$

$R_i(\boldsymbol{s}(i), \mathbf{x}_i, k)$ denotes the partial costs of object i and $s(i)$ denotes the structure part of the solution that relates to object i. For a parametric centroid-based clustering model $s(i)$ would be the centroid object i is assigned to. Using the object-wise mapping function ψ to map objects $i' \in O^{(2)}$ to objects in $O^{(1)}$, we define the **transfer costs** $R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k)$ of a solution $\boldsymbol{s}$ with model-order k as follows:

$$R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k) := \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{i=1}^{N_1} R_{i'}(\boldsymbol{s}^{(1)}(i), \mathbf{x}_{i'}^{(2)}, k) \; \mathbb{I}_{\{\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = i\}}. \tag{3}$$

For each object $i' \in O^{(2)}$ we compute the costs of i' with respect to the learned solution $\boldsymbol{s}(\mathbf{X}^{(1)})$. The mapping function $\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ ensures that the cost function treats the measurement $\mathbf{x}_{i'}^{(2)}$ with $i' \in O^{(2)}$ as if it was the object $i \equiv \psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \in O^{(1)}$. In the limit of many observations N_2, the transfer costs converge to $\mathbb{E}[R(\boldsymbol{s}^{(1)}, \mathbf{X}, k)]$,

the expected costs of the solution $\boldsymbol{s}^{(1)}$ with respect to the probability distribution of the measurements. Minimizing this quantity, with respect to the solution is what we are ultimately interested in. The minimum transfer cost principle (MTC) selects the model-order k with lowest transfer costs. MTC disqualifies models with a too high complexity that perfectly explain $\mathbf{X}^{(1)}$ but fail to fit $\mathbf{X}^{(2)}$ (overfitting), as well as models with too low complexity which insufficiently explain both of them (underfitting).

We would like to emphasize the relation of our method to cross-validation in supervised learning which is frequently used in classification or regression. In supervised learning a model is trained on a set of given observations $\mathbf{X}^{(1)}$ and labels (or output variables) $\mathbf{y}^{(1)}$. Usually , we assume i.i.d. training and test data in classification and, therefore, the transfer problem disappears.

A variant and a special case of the mapping function: In the following, we will describe two other mapping variants. In many problems such as clustering, a solution is a set of structures where the objects inside a structure are statistically indistinguishable by the algorithm. Therefore, the objects $O^{(2)}$ can directly be mapped to the structures inferred from $\mathbf{X}^{(1)}$ rather than to individual objects, since the objects in each structure are unidentifiable. In this way, the mapping function assigns the objects $O^{(2)}$ to the solution $\boldsymbol{s}(\mathbf{X}^{(1)}) \in \mathcal{S}$:

$$\psi^s : O^{(2)} \times \mathcal{S} \times \mathcal{X} \to S(O^{(1)}) , \quad \left(i', \boldsymbol{s}(\mathbf{X}^{(1)}), \mathbf{X}^{(2)}\right) \mapsto \psi\left(i', \boldsymbol{s}(\mathbf{X}^{(1)}), \mathbf{X}^{(2)}\right) . \quad (4)$$

The *generative* mapping, another variant of the ψ function, is obtained in a natural way by data construction. Given the true model parameters, we randomly sample pairs of data items. This gives the identity mapping between the pairs in $O^{(1)}$ and $O^{(2)}$ and can be used whenever the data is artificially generated.

$$\psi^G : O^{(2)} \to O^{(1)} , \quad i' \mapsto \psi(i') = i . \quad (5)$$

In practice, however, the data is usually generated in an unknown way. One has a single dataset $\mathbf{X}$ and subdivides it (eventually multiple times) into random subsets $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$ which are not necessarily of equal cardinality. The *nearest-neighbor* mapping is obtained by assigning each object $i' \in O^{(2)}$ to the structure or object where the costs of $R(\boldsymbol{s}, \mathbf{X}(O^{(1)} \cup i'), k)$ is minimized. In the cases where multiple objects or structures satisfy this condition, i' is randomly assigned to one of them.

3 The Easy Case: Gaussian Mixture Models

We start with mixtures of Gaussians (GMM). We will see that for this model, the transfer of the learned solution to a second dataset is straightforward and requires no particular mapping function. This case is still a good example to start with as it demonstrates that cross-validation for unsupervised learning is a powerful technique that can compete with well known model-selection scores such as BIC and AIC.

A GMM solution consist of the centers $\boldsymbol{\mu}_t$ and the covariances $\boldsymbol{\Sigma}_t$ of the Gaussians, as well as the mixing coefficients π_t. The model order is the number of Gaussians k and the cost function is the negative log likelihood of the model

$$R(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{X}, k) = -\sum_{i=1}^{N} \ln \left(\sum_{t=1}^{k} \pi_t N(\mathbf{x}_i | \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \right) \tag{6}$$

As all model parameters are independent of the object index i, it is straightforward to compute the transfer costs on a second dataset. The learned model parameters provide a probability density estimate for the entire measurement space such that the individual likelihood of each new data item can be readily computed. The transfer costs are $R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k) = R(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}, \mathbf{X}^{(2)}, k)$

We carry out experiments by generating 500 items from three Gaussians. As we increase their variances to increase their overlap, we learn GMM's with varying number of Gaussians k and compute the BIC score, the AIC score, as well as the transfer costs. Two exemplary results are illustrated in Figure 1: an easy setting in the upper row and a difficult setting with high overlap in the lower row. In the easy case, each of the four methods selects the correct number of clusters. For increasing overlap, AIC exhibits a tendency to select a too high number of components. At the variance depicted in the lower plots, BIC starts selecting $k < 3$, while MTC still estimates 3 Gaussians. For very high overlap, we observe that both BIC and MTC select $k = 1$ while AIC selects the maximum number of Gaussians that we offered. The interval of the standard deviation where BIC selects a lower number of Gaussians than MTC ranges from 60% of the distance between the centers (illustrated in Figure 1 bottom) to 85%. The reason for this discrepancy has to be theoretically explored. This gap might be due to MTC being less accurate than the BIC score that is exact in the asymptotic limit of many observations. Maybe, BIC underfits due to non-asymptotic corrections. Visual inspection of the data suggests that this discrepancy regime poses a hard model-order selection problem.

4 Model Order Selection for Truncated SVD

4.1 Image Denoising with Rank-Limited SVD

SVD provides a powerful, yet simple method of denoising images. Given a noisy image, one extracts small $n \times m$ patches from the image (where usually $m = n$) and computes a rank-limited SVD on the matrix $\mathbf{X}$ containing the ensemble of all patches, i.e. the pixel values of one patch are one row in $\mathbf{X}$. SVD provides a dictionary that describes the image content on a local level. Restricting the rank of the decomposition, the image content is approximated and, hopefully, denoised. SVD has been frequently applied to image denoising in the described way or as part of more sophisticated methods (e.g. [8]). Thereby, selecting the rank of the decomposition poses a crucial modeling choice. In [8], for instance, the rank is selected by experience of the authors and the issue of automatic selection is shifted to further research. Here, we address this specific part of the problem. The task is to select the rank of the SVD decomposition such that the denoised image is closest to the noise-free image. Please note that our goal is not primarily to achieve the very best denoising error given an image (clearly, better image denoising techniques than SVD exist). Therefore, we do not optimize on other parameters such as the size of the patches. The main goal is to demonstrate that MTC selects the optimal rank for a defined task, such as image denoising, conditioned on a predefined method.

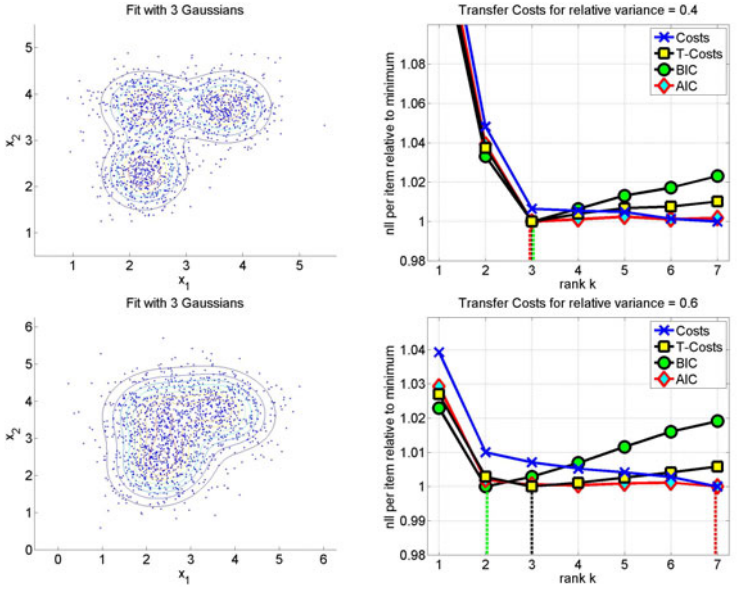


Fig. 1. Selecting the number of Gaussians k. Data is generated from 3 Gaussians. Going from the upper to the lower row, their overlap is increased. For very high overlap, BIC and MTC select $k = 1$. The lower row illustrates the smallest overlap where BIC selects $k < 3$.

We extract $N = 4096$ patches of size $D = 8 \times 8$ from the image and arrange each of them in one row of a matrix $\mathbf{X}$. We randomly split this matrix along the rows into two sub-matrices $\mathbf{X^{(1)}}$ and $\mathbf{X^{(2)}}$ and select the rank k that minimizes the transfer costs

$$R^T(\boldsymbol{s}, \mathbf{X}, k) = \frac{1}{N_2} \left\| \psi_{NN}(\mathbf{X^{(1)}}, \mathbf{X^{(2)}}) \circ \mathbf{X^{(2)}} - \left(\mathbf{U_k^{(1)}} \mathbf{S_k^{(1)}} \mathbf{V_k^{(1)T}} \right) \right\|_2^2 . \quad (7)$$

The mapping $\psi_{NN}(\mathbf{X^{(1)}}, \mathbf{X^{(2)}})$ reindexes all objects of the test set with the indices of their nearest neighbors in the training set. We illustrate the results for the Lenna image in Figure 2 by color-coding the peak-SNR of the image reconstruction. As one can see, there is a crest ranging from a low standard deviation of the added Gaussian noise and maximal rank ($k = 64$) down to the region with high noise and low optimal rank ($k = 1$). The top of the crest marks the optimal rank for given noise (dashed magenta line). The rank selected by MTC is highlighted by the solid black line (dashed lines are three times the standard deviation). The selected rank is always very close to the optimum. At low noise where the crest is rather broad, the deviation from the optimum is maximal. There the selection problem is most difficult. However, in this parameter range the choice of the rank has little influence on the error. For high noise, where a deviation from the optimum has higher influence, our method finds the optimal rank.

4.2 Denoising Boolean Matrices with SVD

In this section, we investigate how well the appropriate rank of SVD is found in a model-mismatch situation. Here, we apply SVD to Boolean data, namely to Boolean

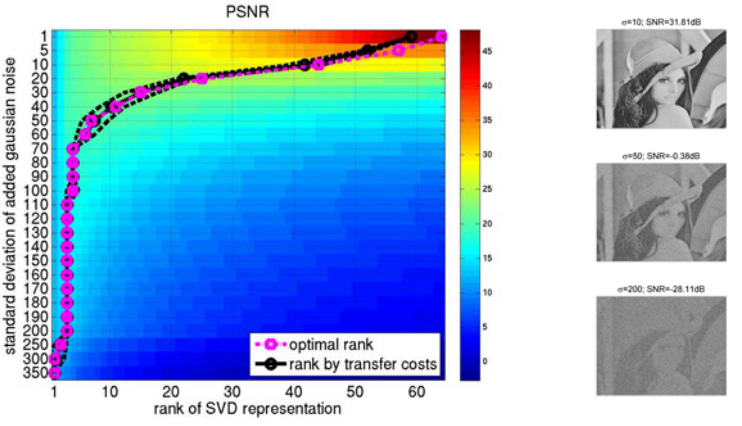


Fig. 2. PSNR (logarithmic) of the denoised image as a function of the added noise and the rank of the SVD approximation of the image patches. The crest of this error marks the optimal rank at a given noise level and is highlighted (dashed magenta). The rank selected by MTC (solid black) is close to this optimum.

access-control configurations. Such a configuration indicates which user has the permission to access which resources and it is encoded in a Boolean matrix $\mathbf{X}$, where a 1-entry means that the permission is granted to the user. In practice, a given user-permission assignment is often noisy, meaning that some individual user-permission assignments do not correspond to the regularities of the data and should thus be regarded as exceptions or might even be errors. Such irregularities pose not only a security-relevant risk but they also constitute a problem when such direct access control systems are to be migrated to role based access control (RBAC) via so-called role mining methods [14]. As most existing role mining methods today are very sensitive to noise [9], they could benefit a lot from denoising as a preprocessing step. In [16], SVD and other continuous factorization techniques for denoising $\mathbf{X}$ are proposed. Molloy et al. compute a rank-k approximation $\mathbf{U_k S_k V_k^T}$ of $\mathbf{X}$. Then, a function g maps all individual entries higher than 0.5 to 1 and the others to 0. The distance of the resulting denoised matrix $\tilde{\mathbf{X}}_\mathbf{k} = g(\mathbf{U_k S_k V_k^T})$ to the error-free matrix $\mathbf{X}^*$ depends heavily on k. The authors propose two methods for selecting the rank k. The first method takes the minimal rank such that the approximation $\tilde{\mathbf{X}}_\mathbf{k}$ covers 80% of the entries of $\mathbf{X}$ (this heuristic originates from the rule of thumb that 20% of the entries of $\mathbf{X}$ are corrupted). The second method selects the smallest rank that decreases the approximation increment $||(\tilde{\mathbf{X}}_\mathbf{k} - \tilde{\mathbf{X}}_\mathbf{k+1})||_1 \Big/ ||\mathbf{X}||_1$ below 0.001.

We also compare with the rank selected by the Bi-crossvalidation method for SVD presented by Owen and Perry [19]. This method, which we will term OP-CV, divides the $n \times d$ input matrix $\mathbf{X}_{1:n,1:d}$ into four submatrices, $\mathbf{X}_{1:p,1:q}$, $\mathbf{X}_{1:p,q+1:d}$, $\mathbf{X}_{p+1:n,1:q}$, and $\mathbf{X}_{p+1:n,q+1:d}$ with $p < n$ and $q < d$. Let $\mathbf{M}^\dagger$ be the Moore-Penrose inverse of the matrix $\mathbf{M}$. OP-CV learns the truncated SVD $\hat{\mathbf{X}}^{(k)}_{p+1:n,q+1:d}$ from $\mathbf{X}_{p+1:n,q+1:d}$ and

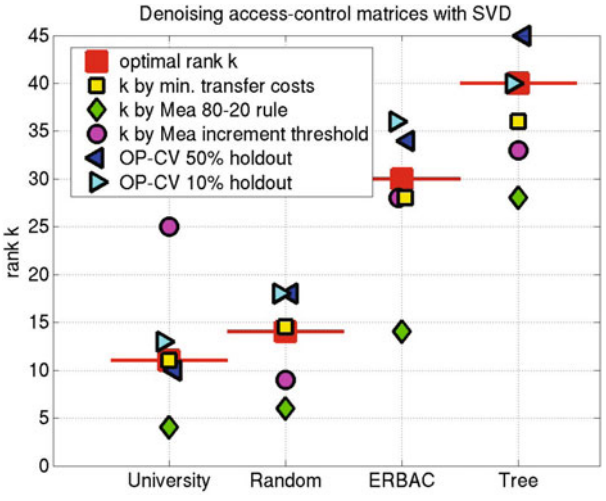


Fig. 3. Denoising four different access-control configurations via rank-limited SVD. The ranks selected by transfer costs and OP-CV are significantly closer to the optimal rank than the ranks selected by the originally proposed methods [Molloy et al., 2010].

computes the error score $\epsilon = \mathbf{X}_{1:p,1:q} - \mathbf{X}_{1:p,q+1:d}(\hat{\mathbf{X}}^{(k)}_{p+1:n,q+1:d})^{\dagger}\mathbf{X}_{p+1:n,1:q}$. In our experiments, we compute ϵ for 20 permutations of the input matrix and select the rank with lowest median error.

We compare the rank selected by the described approaches to the rank selected by MTC with nearest-neighbor mapping and Hamming distance. The four different datasets are taken from [16]. The first dataset 'University' is the access control configuration of a department, the other three are artificially created, each with differing generation processes as described in [16]. The sizes of the datasets are (users$\times$permissions) 493×56, 500×347, 500×101, and 500×190. We display the results in Figure 3. The optimal rank for denoising is plotted as a big red square. The statistics of the rank selected by MTC is plotted as small bounded squares. We select the median over 20 random splits of the dataset. As one can see, the minimum transfer cost rank is always significantly closer to the optimal rank than the ranks selected by the originally proposed methods. The performance of the 80-20 rule is very poor and performance of the increment threshold depends a lot on the dataset. The Bi-crossvalidation method by Owen and Perry (OP-CV) finds good ranks, although not so reliably as MTC. It has been reported that, for smaller validation sets, OP-CV tends to overfit. We could observe this effect in some of our experiments and also on the University dataset. However, on the Tree dataset it is actually the method with the larger validation set that overfits.

5 Minimum Transfer Costs for Boolean Matrix Factorization

In this section, we use MTC to select the number of factors in Boolean matrix factorization for role mining [14]. A real-world access-control matrix $\mathbf{X}$ with 3000 users and

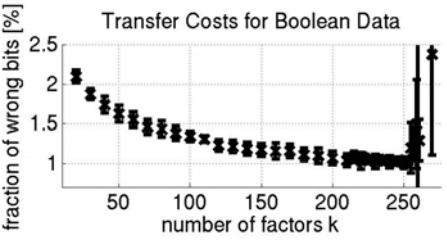


Fig. 4. Model-order selection for Boolean matrix factorization

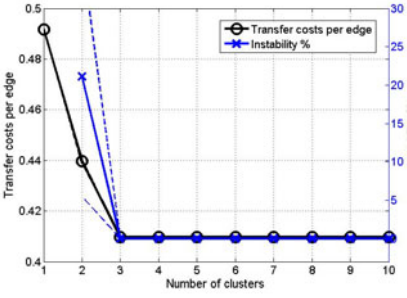
500 permissions defines the data set for role mining applications. We factorize this user-permission matrix into a user-role assignment matrix $\mathbf{Z}$ and a user-permission assignment matrix $\mathbf{U}$ by maximizing the likelihood derived in [22]. Five-fold cross-validation is performed with 2400 users in the training set and 600 users in the test set. As in the last section, the mapping function uses the nearest-neighbor rule with Hamming metric. Here, the MTC score in (Eq. 3) measures the number of bits in $\mathbf{x}_{i'}^{(2)}$ that do not match the decomposition: $R_{i'}(\boldsymbol{s}^{(1)}(i), \mathbf{x}_{i'}^{(2)}, k) = \sum_j \left| x_{i'j}^{(2)} - \bigvee_{t=1}^{k} (z_{it}^{(1)} \wedge u_{tj}^{(1)}) \right|$. This measure differs from the other experiments, where the cost function for optimization on the training data and the cost function for MTC are equal. MTC applies any desired cost function to the hold out dataset.

The number of factors with best generalization ability is $k = 248$. In the underfitting regime, the transfer costs have low variance because the structure in the data equals in all random validation sets. In the overfitting regime, the transfer costs vary significantly as the noisy bits in the validation set determine how well the overfitted model matches the data.

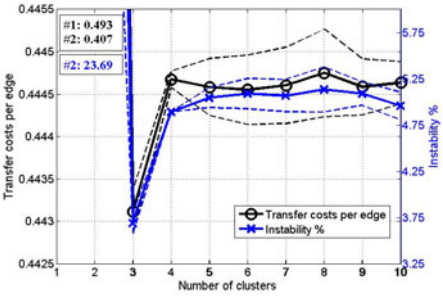
6 Minimum Transfer Costs for Non-factorial Models

The representation of the measurements plays an important role for optimization. In parametric or central clustering, the cost function can be written as a sum over independent object-wise costs $R_i(\boldsymbol{s}, \mathbf{x}_i, k)$ as shown in Eq. (2). When the measurements are characterized by pairwise (dis)similarities, instead of explicit coordinates, then such a function form of the costs as in Eqs (2), (3) does not exist. An example is the quadratic cost function for correlation clustering [3]. In the following, we explain how to obtain the transfer costs for such models.

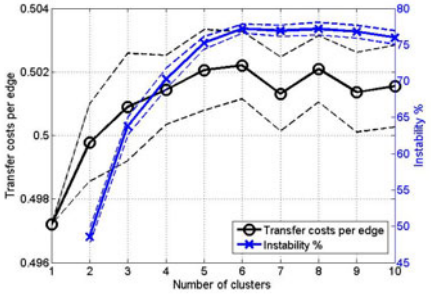
Correlation clustering partitions a graph with positive and negative edge labels. Given a graph $\mathcal{G}(O, \mathbf{X})$ with similarity matrix $\mathbf{X} := \{X_{ij}\} \in \{\pm 1\}^{\binom{N}{2}}$ between objects i and j and a clustering solution $\boldsymbol{s}$, the set of edges between two clusters u and v is defined as $E_{u,v} = \{(i,j) \in E : \boldsymbol{s}(i) = u \wedge \boldsymbol{s}(j) = v\}$, where $\boldsymbol{s}(i)$ is the cluster index of object i. $E_{u,v}, v \neq u$ are *inter-cluster edges* and $E_{u,u}$ are *intra-cluster edges*. Given the noise parameter p and the complexity parameter q, the correlation graph is generated in the following way:



(a) $p = 0.70$



(b) $p = 0.80$



(c) $p = 0.95$

Fig. 5. Transfer costs and instability for various noises p. The complexity q is kept fixed at 0.30.

1. Construct a perfect graph, i.e. assign the weight $+1$ to all intra-cluster edges and -1 to all inter-cluster edges.
2. Change the weight of each inter-cluster edge in $E_{u,v}, v \neq u$ to $+1$ with probability q, increasing structure complexity.
3. With probability p, replace the weight of each edge ($E_{u,v}, v \neq u$ and $E_{u,u}$) by a random weight.

Let N and k be the number of objects and the number of clusters, respectively. The cost function counts the number of disagreements, i.e. the number of negative intra-cluster edges plus the number of positive inter-cluster edges:

$$R(\boldsymbol{s}, \mathbf{X}, k) = -\frac{1}{2} \sum_{1 \leq u \leq k} \sum_{(i,j) \in E_{u,u}} (X_{ij} - 1) + \frac{1}{2} \sum_{1 \leq u \leq k} \sum_{1 \leq v < u} \sum_{(i,j) \in E_{u,v}} (X_{ij} + 1). \quad (8)$$

To transfer the clustering solution $\boldsymbol{s}^{(1)}$ to the second dataset $\mathbf{X}^{(2)}$, we use the Hamming distances between objects i' from $O^{(2)}$ and the clusters inferred from $\mathbf{X}^{(1)}$. The cluster index of object i' is determined by:

$$\boldsymbol{s}^{(1)}(i') = \arg \min_{1 \leq v \leq k} H(i', \boldsymbol{s}_v^{(1)}), \quad \text{with} \quad (9)$$

$$H(i', \boldsymbol{s}_v^{(1)}) = -\frac{1}{2}\sum_{j\in\boldsymbol{s}_v}(X_{ij}-1) + \frac{1}{2}\sum_{1\le u\le k, u\neq v}\sum_{j\in\boldsymbol{s}_u}(X_{ij}+1), \tag{10}$$

where $\boldsymbol{s}_v$ includes the set of objects whose cluster indices are v.

For our experiments, we construct a graph with 900 nodes and 3 clusters. We fix the structure complexity at $q = 0.30$ and vary the noise level p from 0.7 to 0.95. We then divide the graph into two smaller graphs of identical cardinality $N_1 = N_2 = 450$. For clustering, we use Gibbs sampling since, according to our experiments, it usually achieves lower costs than approximation algorithms such as CC-Pivot [1]. We run the sampler with a number of clusters varying from 1 to 10 each for 10 different random initializations. We compare the transfer costs with the instability measure proposed in [15]. The results are summarized in Figure 5. At $p = 0.70$ the problem is simple, which means that the Gibbs sampler, even when initialized with a large number of clusters, always selects the correct number of clusters on its own. The extra clusters are simply left empty. As a consequence, the transfer costs are indifferent for a number of clusters larger than or equal to the correct number (Figure 5(a)). At $p = 0.80$ the problem is complicated but still learnable. Here, the inferred clustering and also the transfer costs vary for different choices of the number of clusters. As illustrated in Figure 5(b) the minimal transfer cost selects the true number of clusters. For both $p = 0.70$ and $p = 0.80$ the instability measure is consistent with the transfer costs. At $p = 0.95$ the edge labels are almost entirely random, hiding all structure in the data. Therefore, as Figure 5(c) confirms, the number of learnable clusters is 1. In this regime, instability cannot determine the correct number of clusters as it is not defined for $k = 1$.

7 Transfer Costs for k-means Clustering

In this last example, we investigate a conceptually difficult task, namely the application of k-means to Gaussian data. A solution $\boldsymbol{s}$ of k-means is an assignment vector $\boldsymbol{c} \in \{1, .., k\}^N$ and k centroids $\boldsymbol{\mu}_t : t \in \{1, .., k\}$. Thereby, $c(i) = t$ means that object i is assigned to cluster t. The model order is the number of centroids k. The cost function of k-means is the sum of distances between each object and its centroid, i.e. $R(\boldsymbol{s}, \mathbf{X}, k) = \sum_i d\left(\boldsymbol{\mu}_c(i), \mathbf{x}_i\right)$. The distance function d depends on the data type (Hamming, squared Euclidean ...). As k-means provides a disjoint partitioning of the objects into the k clusters, one can rewrite the transfer cost formula:

$$\begin{aligned} R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k) &= \frac{1}{N_2}\sum_{i'=1}^{N_2}\sum_{i=1}^{N_1} d\left(\boldsymbol{\mu}_{c(i)}^{(1)}, \mathbf{x}_{i'}^{(2)}\right)\mathbb{I}_{\{\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)})=i\}} \\ &\approx \frac{1}{N_2}\sum_{i'}\sum_{t} d\left(\boldsymbol{\mu}_t^{(1)}, \mathbf{x}_{i'}^{(2)}\right)\mathbb{I}_{\{\psi^s(i', \boldsymbol{s}^{(1)}, \mathbf{X}^{(2)})=t\}}, \end{aligned} \tag{11}$$

whereas ψ is the nearest-neighbor mapping between objects and ψ^s is the mapping of objects to the nearest centroid as defined in Eq. (4). We use the fact that the centroids represent the objects which are assigned to them. Therefore, the centroid closest to $\mathbf{x}_{i'}^{(2)}$ is on average approximately as far away as the centroid of the nearest neighbor of $\mathbf{x}_{i'}^{(2)}$ in $O^{(1)}$. For high N_1 and N_2 this approximation becomes very precise.

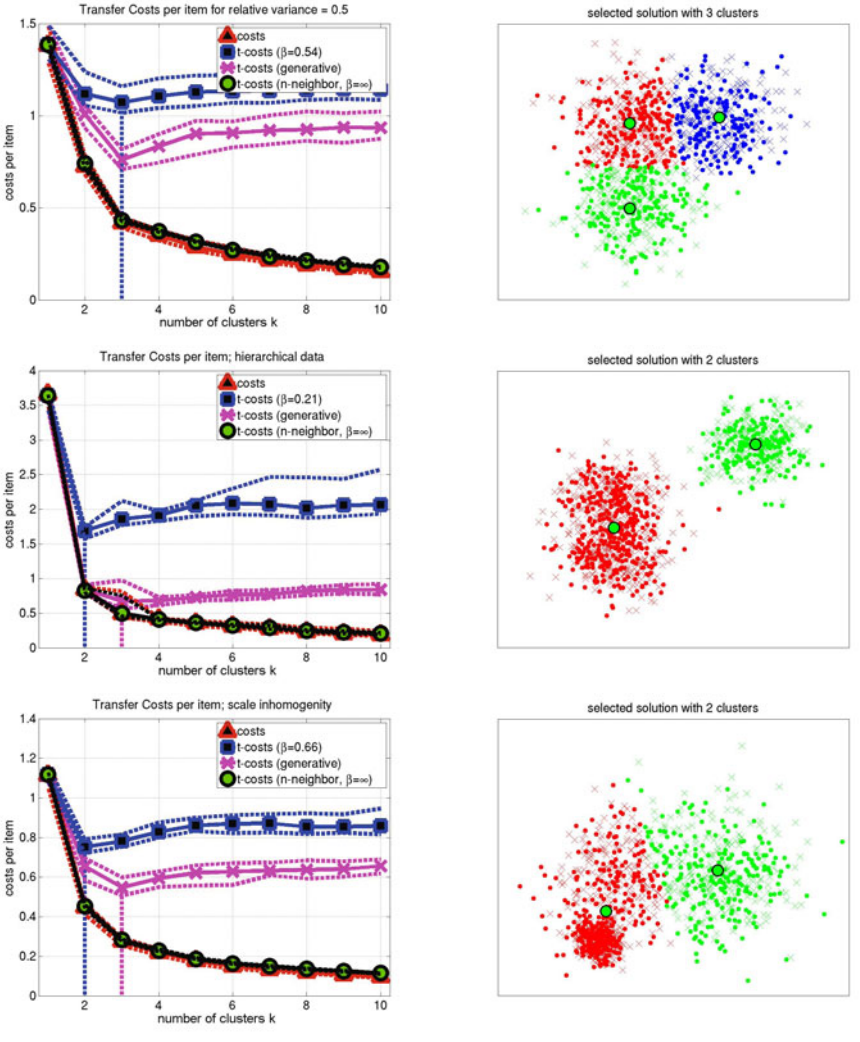


Fig. 6. Costs and transfer costs (computed with mappings: nearest-neighbor, generative, soft) for k-means clustering of three Gaussians. Solid lines indicate the median and dashed lines are the 25% and 75% percentiles. The right panel shows the clustering result selected by soft mapping MTC. Top: equidistant centers and equal variance. Middle: heterogeneous distances between centers (hierarchical). Bottom: heterogeneous distances and variances.

The setup of the experiment is as follows: We sample 200 objects from three bivariate Gaussian distributions (see for instance Figure 6 top right). The task is to find the appropriate number of clusters. By altering the variances and the pairwise distances of the centers, we control the difficulty of this problem and especially tune it such that selecting the number of clusters is hard. We investigate the selection of k by the

nearest-neighbor mapping of the objects from the second dataset to the centroids $\boldsymbol{\mu}^{(1)}$ as well as by the generative mapping where the two data subsets are aligned by construction. We report the statistics over 20 random repetitions of generating the data.

Our findings for three different problem difficulties are illustrated in Figure 6. As expected, the costs on the training dataset monotonically decrease with k. When the mapping is given by the generation process of the data (generative mapping), MTC provides the true number of clusters in all cases. However, recall that the generative mapping requires knowledge of the true model parameters and leaks information about the true number of clusters to the costs. Interestingly, MTC with a nearest-neighbor mapping follows almost exactly the same trend as the original costs on the first dataset and therefore proposes selecting the highest model-order that we offer to MTC. The higher the number of clusters is, the closer are the centroids of the nearest neighbors of each object. This reduces the transfer costs of high k. The only difference between original costs and transfer costs stems from the average distance between nearest neighbors (the data granularity). Only when the pairwise centroid distances become smaller than this distance, the transfer costs increase again. Ultimately, the favored solution is a vector quantization at the level of the data granularity. This is the natural behavior of k-means, as its cost function has no variances. As we have seen in the first experiments with Gaussian mixture models, fitting Gaussian data with MTC imposes no particular difficulties when the appropriate model (here GMM) is used. The k-means behavior is due to a model mismatch.

Probabilistic Mapping: A variant of MTC can be used to still make k-means applicable to estimating the true model order of Gaussian data. As follows, we extend the notion of a strict mapping to a probabilistic mapping between objects. Let $p_{i'i} := p(\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = i)$ be the probability that ψ maps object i' from the second dataset to object i of the first dataset. We define $p_{i'i}$ as

$$p_{i'i} := Z^{-1} \exp\left(-\beta\, d(\mathbf{x}_i^{(1)}, \mathbf{x}_{i'}^{(2)})\right) \;,\; Z = \sum_i \exp\left(-\beta\, d(\mathbf{x}_i^{(1)}, \mathbf{x}_{i'}^{(2)})\right) \tag{12}$$

This mapping distribution is parameterized by the computational temperature β^{-1} and depends on the problem-specific dissimilarity function $d(\mathbf{x}_i^{(1)}, \mathbf{x}_{i'}^{(2)})$. A probabilistic mapping is more general than the deterministic function ψ. When β has a finite value, then objects are mapped to more than one other object. In the case of $\beta \to \infty$, it reduces to a deterministic nearest-neighbor mapping between $O^{(2)}$ and $O^{(1)}$. When $\beta = 0$ then object $i' \in O^{(2)}$ is mapped to all N_1 objects in $O^{(1)}$ with equal probability, thereby maximizing the entropy of $p_{i'i}$.

Using this probabilistic mapping, we define the transfer costs $R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k)$ of a factorial model with model-order k as follows:

$$R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k) = \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{i=1}^{N_1} p_{i'i} R_{i'}(\boldsymbol{s}^{(1)}(i), \mathbf{x}_{i'}^{(2)}, k). \tag{13}$$

For k-means, taking the object to centroid approximation, this becomes

$$R^T(\boldsymbol{s}^{(1)}, \mathbf{X}^{(2)}, k) \approx \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{t=1}^{k} d\left(\boldsymbol{\mu}_t^{(1)}, \mathbf{x}_{i'}^{(2)}\right) \frac{e^{-\beta d(\boldsymbol{\mu}_t^{(1)}, \mathbf{x}_i^{(2)})}}{\sum_{t'=1}^{k} e^{-\beta d(\boldsymbol{\mu}_{t'}^{(1)}, \mathbf{x}_i^{(2)})}} \tag{14}$$

We fix the inverse temperature by the costs of the data with respect to a single cluster: $\beta = 0.75 * R(\boldsymbol{s}^{(1)}, \mathbf{X}^{(1)}, 1)^{-1}$. This choice defines the dynamic range of the model-order selection problem. When fixing β roughly at the costs of one cluster, the resolution of individual pairwise distances resembles the visual situation where one looks at the entire data cloud as a whole.

Results of probabilistic mapping MTC: The probabilistic mapping finds the true number of clusters when the variances of the Gaussians are roughly the same, even for a substantial overlap of the Gaussians (Figure 6, top row). Please note that although the differences of the transfer costs are within the plotted percentiles, the rank-order of the number of clusters in each single experiment is preserved over the 20 repetitions, i.e. the variance mainly results from the data and not from the selection of k.

When the problem scale varies on a local level, fixing the temperature at the $k = 1$ solution does not resolve the dynamic range of the costs. We illustrate this by two hard problems: The middle problem in Figure 6 has a hierarchical structure, i.e. the pairwise distances between centers vary a lot. In the bottom problem in Figure 6, both the distances and the individual variances of the Gaussians vary. In both cases the number of clusters is estimated too low. When inspecting the middle plot, this choice seems reasonable, whereas in the bottom plot clearly three clusters would be desirable. The introduction of a computational temperature simulates the role of the variances in Gaussian mixture models. However, as the temperature is the same for all clusters, it fails to mimic situations where the variances of the Gaussians substantially differ. A Gaussian mixture model would be more appropriate than modeling Gaussian data with k-means.

8 Related Work

In this section we point to related work on model selection for unsupervised learning. Models that assume an explicit parametric form, are often controlled by a model complexity penalty (a regularizer). Akaike information criterion (AIC) [2] and Bayesian information criterion (BIC) [21] both trade off the goodness of fit measured in terms of a likelihood function against the number of model parameters used. In [18], the model evidence for probabilistic PCA is maximized with respect to the number of components. Introducing approximations, this score equals BIC. In [12] the number of principal components is selected by integrating over the sensitivity of the likelihood to the model parameters. Minimum description length (MDL) [20] selects the lowest model order that can explain the data. It essentially minimizes the negative log posterior of the model and is thus formally identical to BIC [13]. It is unclear how to generalize model-based critera like [2,21,18,12] to non-probabilistic methods such as, for instance, correlation clustering, being specified by a cost function instead of a likelihood.

For selecting the rank of truncated SVD, probably the most related approach is the cross-validation method proposed in [19]. It is a generalization of the method in [11] and was also applied to NMF. We explain it and compare with it in Section 4.2. A method with single hold-out entries (i, j) is proposed in [7]. It trains a SVD on the input matrix without row i and another one without column j. Then it combines $\mathbf{U}$ from one SVD and $\mathbf{V}$ from the other and averages their singular values to obtain an SVD which is independent of (i, j). The method in [7] has been reviewed in [19].

In [17], the authors abandon cross-validation for Boolean matrix factorization. They found that i) the method in [19] is not applicable and ii) using the rows of the second matrix of the factorization (here $\mathbf{U}$ in Section 5) to explain the hold-out data, tolerates overfitting. From our experience, cross-validation fails when only the second matrix is fixed and the first matrix is adapted to the new data. With a predefined mapping to transfer *both* matrices to the new data without adapting them, cross-validation works for Boolean matrix factorization as demonstrated in Section 5.

Specialized to selecting the number of clusters in clustering, gap statistics have been proposed in [23]. Stability analysis has also shown promising results [6,15]. Stability neglects to account the informativeness of solutions. An information theoretic model validation principle has been proposed in [4] to determine the tradeoff between stability and informativeness based on an information theoretic criterion called *approximation capacity*. So far, this principle has been applied to clustering [5] and SVD [10].

9 Conclusion

We defined the minimum transfer cost principle (MTC) and proposed several variants of how to apply it. Our method extends the cross-validation principle to unsupervised learning problems as it solves the problem of transferring a learned model from one dataset to another one when no labels are given. We demonstrated how to apply the principle to different problems such as max. likelihood inference, k-means clustering, correlation clustering, Gaussian mixture models, and rank-limited SVD, highlighting its broad applicability. For each problem, we explained the appropriate mapping function between datasets and we demonstrated how the principle can be employed with respect to the specifications of the particular tasks. In all cases, MTC makes a sensible choice of the model order. It finds the optimal rank for image denoising with SVD and for error correction in access-control configurations. Future work will cover the application of our principle to other models as well as to other tasks such as feature selection.

Acknowledgements. This work was partially supported by the Zurich Information Security Center, by the DFG-SNF research cluster FOR916, and by the FP7 EU project SIMBAD.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. Journal of the ACM 55, 23:1–23:27 (2008)
2. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
3. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2002)
4. Buhmann, J.M.: Information theoretic model validation for clustering. In: ISIT 2010 (2010)
5. Buhmann, J.M., Chehreghani, M.H., Frank, M., Streich, A.P.: Information theoretic model selection for pattern analysis. In: JMLR: Workshop and Conference Proceedings, vol. 7, pp. 1–8 (2011)

6. Dudoit, S., Fridlyand, J.: A prediction-based resampling method for estimating the number of clusters in a dataset. Genome biology 3(7) (2002)
7. Eastment, H.T., Krzanowski, W.J.: Cross-validatory choice of the number of components from a principal component analysis. Technometrics 24(1), 73–77 (1982)
8. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
9. Frank, M., Buhmann, J.M., Basin, D.: On the definition of role mining. In: SACMAT, pp. 35–44 (2010)
10. Frank, M., Buhmann, J.M.: Selecting the rank of truncated SVD by Maximum Approximation Capacity. In: IEEE International Symposium on Information Theory, ISIT (2011)
11. Gabriel, K.: Le biplotoutil dexploration de données multidimensionelles. Journal de la Societe Francaise de Statistique 143, 5–55 (2002)
12. Hansen, L.K., Larsen, J.: Unsupervised learning and generalization. In: IEEE Intl. Conf. on Neural Networks, pp. 25–30 (1996)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2001)
14. Kuhlmann, M., Shohat, D., Schimpf, G.: Role mining – revealing business roles for security administration using data mining technology. In: SACMAT 2003, p. 179 (2003)
15. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability-based validation of clustering solutions. Neural Computation 16(6), 1299–1323 (2004)
16. Molloy, I., et al.: Mining roles with noisy data. In: SACMAT 2010, pp. 45–54 (2010)
17. Miettinen, P., Vreeken, J.: Model Order Selection for Boolean Matrix Factorization. In: SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
18. Minka, T.P.: Automatic choice of dimensionality for PCA. In: NIPS, p. 514 (2000)
19. Owen, A.B., Perry, P.O.: Bi-cross-validation of the SVD and the nonnegative matrix factorization. Annals of Applied Statistics 3(2), 564–594 (2009)
20. Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)
21. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics 6, 461 (1978)
22. Streich, A.P., Frank, M., Basin, D., Buhmann, J.M.: Multi-assignment clustering for Boolean data. In: ICML 2009, pp. 969–976 (2009)
23. Tibshirani, R., Walther, G., Hastie, T.: Estimating the Number of Clusters in a Dataset via the Gap Statistic. Journal of the Royal Statistical Society, Series B 63, 411–423 (2000)