Journal of Machine Learning Research 13 (2012) 459-489 Submitted 9/10; Revised 6/11; Published 2/12

Multi-Assignment Clustering for Boolean Data

Mario Frank[*] MFRANK@BERKELEY.EDU
UC Berkeley, Computer Science Division
721 Soda Hall
Berkeley, CA, 94720, USA

Andreas P. Streich[*] ANDREAS.STREICH@PHONAK.COM
Phonak AG, Advanced Concepts & Technologies
Laubisrütistrasse 28
8712 Stäfa, Switzerland

David Basin BASIN@INF.ETHZ.CH
Joachim M. Buhmann JBUHMANN@INF.ETHZ.CH
ETH Zürich, Department of Computer Science
Universitätstrasse 6
8092 Zürich, Switzerland

Editor: Tony Jebara

Abstract

We propose a probabilistic model for clustering Boolean data where an object can be simultaneously assigned to multiple clusters. By explicitly modeling the underlying generative process that combines the individual source emissions, highly structured data are expressed with substantially fewer clusters compared to single-assignment clustering. As a consequence, such a model provides robust parameter estimators even when the number of samples is low. We extend the model with different noise processes and demonstrate that maximum-likelihood estimation with multiple assignments consistently infers source parameters more accurately than single-assignment clustering. Our model is primarily motivated by the task of role mining for role-based access control, where users of a system are assigned one or more roles. In experiments with real-world access-control data, our model exhibits better generalization performance than state-of-the-art approaches.

Keywords: clustering, multi-assignments, overlapping clusters, Boolean data, role mining, latent feature models

1. Introduction

Clustering defines the unsupervised learning task of grouping a set of data items into subsets such that items in the same group are similar. While clustering data into disjoint clusters is conceptually simple, the exclusive assignment of data to clusters is often overly restrictive, especially when data is structured. In this work, we advocate a notion of clustering that is not limited to partitioning the data set. More generally, we examine the task of inferring the hidden structure responsible for generating the data. Specifically, multiple clusters can simultaneously generate a data item using

∗. These authors contributed equally. When most of this work was conducted, all authors were affiliated to ETH Zurich. This project may be found at `http://www.mariofrank.net/MACcode/index.html`.

a problem dependent link function. By adopting a generative viewpoint, such data originate from multiple sources.

Consider, for instance, individuals' movie preferences. A person might belong to the "comedy" cluster or the "classics" cluster, where each cluster membership generates a preference for the respective genre of movies. However, some people like both comedy movies and classics. In standard single-assignment clustering, a third "comedy&classics" cluster would be created for them. Under the generative viewpoint, we may assign individuals simultaneously to both of the original clusters to explain their preferences. Note that this differs from "fuzzy" clustering, where objects are partially assigned to clusters such that these fractional assignments (also called "mixed membership") add up to 1. In our approach, an object can be assigned to multiple clusters *at the same time*, that is, the assignments of an object can sum to a number larger than 1. Membership in a second cluster does not decrease the intensity of the membership in the first cluster. We call this approach *multi-assignment clustering* (MAC).

In a generative model that supports multi-assignments, one must specify how a combination of sources generates an object. In this paper, we investigate clustering for Boolean data. The combined emissions from individual sources generate an object by the Boolean OR operation. In the example of the movie preferences, this means that an individual belonging to both the comedy and the classics cluster likes a comedy film like "Ghostbusters" as much as someone from the comedy cluster, and likes the classic movie "Casablanca" as much as someone who only belongs to the classics group.

In this paper, we develop a probabilistic model for structured Boolean data. We examine various application-specific noise processes that account for the irregularities in the data and we theoretically investigate the relationships among these variants. Our experiments show that multi-assignment clustering computes more precise parameter estimates than state-of-the art clustering approaches. As a real-world application, our model defines a novel and highly competitive solution to the *role mining* problem. This task requires to infer a user-role assignment matrix and a role-permission assignment matrix from a Boolean user-permission assignment relation defining an access-control system. The generalization ability of our model in this domain outperforms other multi-assignment techniques.

The remainder of this paper is organized as follows. In the next section, we survey the literature on Boolean matrix factorization and the clustering of Boolean data. In Section 3, we derive our generative model and its variants and describe parameter inference in Section 4. In Section 5, we present experiments on synthetic and real-world data generated from multiple sources.

2. Related Work

In this section, we provide an overview of existing methods for the exploratory analysis of Boolean data. The described approaches have been developed within different research areas and have different objectives. However, they all aim to produce a structured representation of given binary data. The research areas include association-rule mining, formal concept analysis, clustering, dimension reduction, latent feature models, and database tiling. We distinguish between methods that search for an exact representation of the data and methods that approximate the representation. In the following, we review several related problem formulations and compare the approaches used to solve them.

2.1 Problem Formulations

There are different problem formulations that arise in the context of Boolean matrix factorization. In this section, we explain the most characteristic ones and relate them to each other.

2.1.1 EXACT BOOLEAN MATRIX DECOMPOSITION AND EQUIVALENT PROBLEMS

These methods aim at an exact Boolean factorization of the input matrix. The earliest formulation of such problems is presumably the set-cover problem (also called set basis problem) presented by Gimpel (1974) and Cormen et al. (2001).

Definition 1 *(**Set-Cover Problem**) Given a set of finite sets* $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$*, find a basis* $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_K\}$ *with minimal cardinality K such that each* $\mathbf{x}_i$ *can be represented as a union of a subset of* $\mathbf{u}$*.*

All sets in $\mathbf{x}$ have a vector representation in a D-dimensional Boolean space, where a 1 at dimension d indicates the membership of item d in the respective set. D is the cardinality of the union of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$. The matrix $\mathbf{z} \in \{0,1\}^{N \times K}$ then indicates which subsets of $\mathbf{u}$ cover the sets in $\mathbf{x}$: $z_{ik} = 1$ indicates that $\mathbf{u}_k$ covers $\mathbf{x}_i$. Using this notation, the set-covering problem is equivalent to finding an exact Boolean decomposition of a binary matrix $\mathbf{x}$ with minimal K. An exact Boolean decomposition is $\mathbf{x} = \mathbf{z} * \mathbf{u}$, where the Boolean matrix product $*$ is defined such that

$$x_{id} = \bigvee_{k=1}^{K} (z_{ik} \wedge u_{kd}) \,. \tag{1}$$

Belohlavek and Vychodil (2010) show that the set cover problem is equivalent to Boolean factor analysis, where each factor corresponds to a row of $\mathbf{u}$. Keprt and Snásel (2004) show that the factors together with the objects assigned to them can, in turn, be regarded as formal concepts as defined in the field of Formal Concept Analysis (FCA) by Ganter and Wille (1999). Stockmeyer (1975) shows that the set-cover problem is NP-hard and the corresponding decision problem is NP-complete. Since the set-cover problem is equivalent to the other problems, this also holds for Boolean factor analysis, finding the exact Boolean decomposition of a binary matrix, and FCA. Approximation heuristics exist and are presented below.

2.1.2 APPROXIMATE BOOLEAN MATRIX DECOMPOSITION

An approximate decomposition of a matrix $\mathbf{x}$ is often more useful than an exact one. One can distinguish two problems, which we refer to as the lossy compression problem (LCP) and the structure inference problem (SIP). For LCP, two different formulations exist. In the first formulation of Miettinen et al. (2006), the size of the matrix $\mathbf{u}$ is fixed and the reconstruction error is to be minimized.

Definition 2 *(**LCP1: Minimal Deviation for given** K) For a given binary* $N \times D$ *matrix* $\mathbf{x}$ *and a given number* $K < \min(N, D)$*, find an* $N \times K$ *matrix* $\mathbf{z}$ *and a* $K \times D$ *matrix* $\mathbf{u}$ *such that the deviation* $||\mathbf{x} - \mathbf{z} * \mathbf{u}||$ *is minimal.*

Alternatively, the deviation is given, as in Vaidya et al. (2007), and the minimal $\mathbf{z}$ and $\mathbf{u}$ must be found to approximate $\mathbf{x}$.

Definition 3 *(**LCP2: Minimal** K **for Given Deviation**) For a given binary* $N \times D$ *matrix* $\mathbf{x}$ *and a given deviation* δ*, find the smallest number* $K < min(N, D)$*, a* $N \times K$ *matrix* $\mathbf{z}$*, and a* $K \times D$ *matrix* $\mathbf{u}$ *such that* $||\mathbf{x} - \mathbf{z} * \mathbf{u}|| \leq \delta$*.*

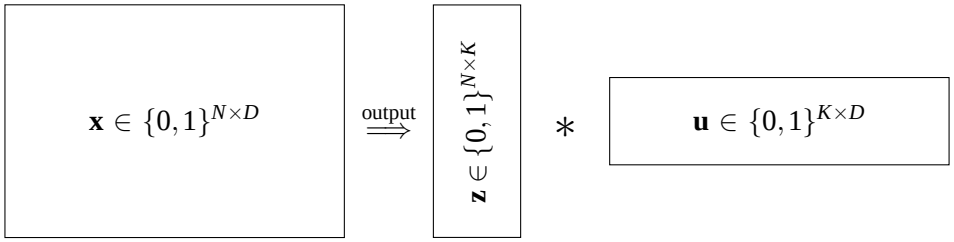


Figure 1: Dimensions of input data and output of the problems defined in Definitions 1–4.

The norm in both formulations of LCP is usually the Hamming distance. Both problems are NP-hard as shown by Vaidya et al. (2007).

In the structure inference problem (SIP), the matrix $\mathbf{x}$ is assumed to be generated from a structure part $(\mathbf{z}^*,\mathbf{u}^*)$ and a random noise part. The goal is to find the decomposition $(\mathbf{z}^*,\mathbf{u}^*)$ that recovers the structure and disregards the noise.

Definition 4 ***(SIP)*** *Let the binary $N \times D$ matrix $\mathbf{x}$ be given. Assuming that $\mathbf{x}$ was generated from a hidden structure $(\mathbf{z}^*,\mathbf{u}^*)$ and perturbed by noise Θ such that $\mathbf{x} \sim p(\mathbf{x}|\Theta, \mathbf{z}^* * \mathbf{u}^*)$, infer the underlying structure $(\mathbf{z}^*,\mathbf{u}^*)$.*

There is a substantial difference between SIP and the two lossy compression problems LCP1 and LCP2. Assuming that some of the entries are corrupted, neither the closest approximation of the original matrix nor the best compression is desirable. Instead, the goal is to infer a structure underlying the data at hand rather than to decompose the matrix with the highest possible accuracy. Since the structure of the data is repeated across the samples, whereas its noise is irregular, better structure recovery will also provide better prediction of new samples or missing observations.

2.2 Approaches

Depending on the problem formulation, there are several ways how the factorization problems are approached. In this section we provide an overview over related methods.

2.2.1 COMBINATORIAL APPROACHES

The problems LCP1 and LCP2 are NP-hard. Heuristic methods to find approximate solutions usually construct candidate sets for the rows of the matrix $\mathbf{u}$, and then greedily pick candidates such that, in each step, the reconstruction error is minimal. For the set covering problem defined in Cormen et al. (2001), the candidate set is the set of all possible formal concepts. For the approximate decomposition problem described in Miettinen et al. (2006), candidates are computed using association rule mining as presented in Agrawal et al. (1993). A predefined number of candidates is then iteratively chosen and assigned to the objects such that, in each step, the data set is optimally approximated. We will refer to this algorithm, originally presented in Miettinen et al. (2006), as the Discrete Basis Problem Solver (DBPS) and use Miettinen's implementation of DBPS in some of our experiments. In the greedy algorithm proposed in Belohlavek and Vychodil (2010), the construction of a large candidate set is avoided by iteratively constructing the next best candidate.

2.2.2 MODEL-BASED APPROACHES

Solutions to the structure inference problem as presented in Wood et al. (2006), Šingliar and Hauskrecht (2006), Kabán and Bingham (2008), and Streich et al. (2009) are often based on probabilistic models. The likelihood that is most similar to the one we propose is the noisy-OR gate introduced in Pearl (1988). Our model allows random flips in *both* directions. The noisy-OR model, which is constrained to random bit flips from zeros to ones, is thus a special case of the noise model that we present in Section 3.2.4. A detailed comparison of the relationship between the noisy-OR model and our approach follows in Section 3.

There are two models that use a noisy-OR likelihood. Noisy-OR component analysis (NOCA), as in Šingliar and Hauskrecht (2006), is based on a variational inference algorithm by Jaakkola and Jordan (1999). This algorithm computes the global probabilities $p(u_j = 1)$, but does not return a complete decomposition. A non-parametric model based on the Indian-Buffet process (Griffiths and Ghahramani, 2011) and a noisy-OR likelihood is presented in Wood et al. (2006). We call this approach infinite noisy-OR (INO). Our method differs from INO with respect to the data likelihood and with respect to optimization. While our model yields an exact solution to an approximate model, replacing the binary assignments by probabilistic assignments, the inference procedure for INO aims at solving the exact model by sampling. INO is a latent feature model, as described by Ghahramani et al. (2007), with Boolean features. Latent feature models explain data by combinations of multiple features that are indicated as active (or inactive) in a binary matrix $\mathbf{z}$. Being a member in multiple clusters (encoded in $\mathbf{z}$) is technically equivalent to having multiple features activated.

Binary independent component analysis (BICA) of Kabán and Bingham (2008) is a factor model for binary data. The combination of the binary factors is modeled with linear weights and thus deviates from the goal of finding binary decompositions as we defined it above. However, the method can be adapted to solve binary decomposition problems and performs well under certain conditions as we will demonstrate in Section 5.2.

Two other model-based approaches for clustering binary data are also related to our model, although more distantly. Kemp et al. (2006) presented a biclustering method that infers concepts in a probabilistic fashion. Each object and each feature is assigned to a single bicluster. A Dirichlet process prior (Antoniak, 1974; Ferguson, 1973) and a Beta-Bernoulli likelihood model the assignments of the objects. Heller and Ghahramani (2007) presented a Bayesian non-parametric mixture model including multiple assignments of objects to binary or real-valued centroids. When an object belongs to multiple clusters, the product over the probability distributions of all individual mixtures is considered, which corresponds to the conjunction of the mixtures. This constitutes a probabilistic model of the Boolean AND, whereas in all the above methods mentioned, as well as in our model, the data generation process uses the OR operation to combine mixture components.

In this paper, we provide a detailed derivation and an in-depth analysis of the model that we proposed in Streich et al. (2009). We thereby extend the noise part of the model to several variants and unify them in a general form. Moreover, we provide an approach for the model-order selection problem.

2.3 Applications

There are numerous applications for Boolean matrix factorization. In this paper we will focus on one specific application, the role mining problem, which was first formulated by Kuhlmann et al. (2003). This problem can be approached as a multi-assignment clustering problem since in role mining the

associated data sets are clearly generated by multiple sources. Our model was motivated by this security-relevant problem. In the following, we will describe this problem and give representative examples of the main role mining methods that have been developed.

2.3.1 ROLE MINING AND RBAC

The goal of role mining is to automatically decompose a binary user-permission assignment matrix **x** into a role-based access control (RBAC) configuration consisting of a binary user-role assignment matrix **z** and a binary role-permission assignment matrix **u**. RBAC, as defined in Ferraiolo et al. (2001), is a widely used technique for administrating access-control systems where users access sensitive resources. Instead of directly assigning permissions to users, users are assigned to one or more roles (represented by the matrix **z**) and obtain the permissions contained in these roles (represented by **u**).

The major advantages of RBAC over a direct user-permission assignment (encoded in the matrix **x**) are ease of maintenance and increased security. RBAC simplifies maintenance for two reasons. First, roles can be associated with business roles, i.e. tasks in an enterprise. This business perspective on the user is more intuitive for humans than directly assigning individual low-level permissions. Second, assigning users to just a few roles is easier than assigning them to hundreds of individual permissions. RBAC increases security over access-control on a user-permission level because it simplifies the implementation and the audit of security policies. Also, it is less likely that an administrator wrongly assigns a permission to a user. RBAC is currently the access control solution of choice for many mid-size and large-scale enterprises.

2.3.2 STRUCTURE AND EXCEPTIONS IN ACCESS CONTROL MATRICES

The regularities of an access control matrix **x**, such as permissions that are assigned together to many users, constitute the structure of **x**. Exceptional user-permission assignments are not replicated over the users and thus do not contribute to the structure. There are three reasons for the existence of such exceptional assignments. First, exceptional assignments are often granted for 'special' tasks, for example if an employee temporarily substitutes for a colleague. Such exceptions may initially be well-motivated, but often the administrator forgets to remove them when the user no longer carries out the exceptional task. The second reason for exceptional assignments is simply administrative mistakes. Errors may happen when a new employee enters the company, or permissions might not be correctly updated when an employee changes position within the company. Finally, exceptional assignments can be intentionally granted to employees carrying out highly specialized tasks.

The role mining step should ideally migrate the regularities of the assignment matrix **x** to RBAC, while filtering out the remaining exceptional permission assignments. We model exceptional assignments (all three cases) with a noise model described in Section 3.2. We are not aware of any way to distinguish these three cases when only user-permission assignments are given as an input. However, being able to separate exceptional assignments from the structure in the data substantially eases the manual search for errors.

2.3.3 PRIOR ART

There is no consensus in the literature on the objective of role mining. An overview of all existing problem definitions is provided in Frank et al. (2010). We consider role mining as an inference problem, which we defined in Definition 4. Numerous algorithms for role mining exist. Molloy et al.

(2008) apply an algorithm from formal concept analysis (see Ganter and Wille, 1999) to construct candidate roles (rows in $\mathbf{u}$). The technique presented in Vaidya et al. (2007) uses an improved version of the database tiling algorithm from Han et al. (2000). In contrast to the method presented in Agrawal and Srikant (1994), their tiling algorithm avoids the construction of all concepts by using an oracle for the next best concept. A method based on a probabilistic model is proposed in Frank et al. (2008). The model is derived from the logical representation of a Boolean two-level hierarchy and is divided into several subcases. For one of the cases with only single assignments of objects, the bi-clustering method presented in Kemp et al. (2006) is used for inference.

3. Generative Model for Boolean Data from Multiple Sources

In this section we explain our model of the generation process of binary data, where data may be generated by multiple clusters. The observed data stems from an underlying structure that is perturbed by noise. We will first present our model for the structure and afterwards provide a unifying view on several noise processes presented in the literature.

We use a probabilistic model that describes the generative process. This has two advantages over discrete optimization approaches. First, considering a separate noise process for the irregularities of the data yields an interpretation for deviations between the input matrix $\mathbf{x}$ and its decomposition $(\mathbf{z},\mathbf{u})$. Second, the probabilistic representation of the sources $\mathbf{u}$ is a relaxation of the original computationally hard problem, as explained in the previous sections.

Let the observed data consist of N *objects*, each associated with D binary *dimensions*. More formally, we denote the data matrix by $\mathbf{x}$, with $\mathbf{x} \in \{0,1\}^{N\times D}$. We denote the i^{th} row of the matrix by x_{i*}, and the d^{th} column by x_{*d}. We use this notation for all matrices.

3.1 Structure Model

The systematic regularities of the observed data are captured by its structure. More specifically, the sources associated with the clusters generate the structure $\mathbf{x}^S \in \{0,1\}^{N\times D}$. The association of data items to sources is encoded in the binary assignment matrix $\mathbf{z} \in \{0,1\}^{N\times K}$, with $z_{ik} = 1$ if and only if the data item i belongs to the source k, and $z_{ik} = 0$ otherwise. The sum of the assignment variables for the data item i, $\sum_k z_{ik}$, can be larger than 1, which denotes that a data item i is assigned to multiple clusters. This multiplicity gives rise to the name *multi-assignment clustering* (MAC). The sources are encoded as rows of $\mathbf{u} \in \{0,1\}^{N\times K}$.

Let the set of the sources of an object be $\mathcal{L}_i := \{k \in \{1,\ldots,K\} \,|\, z_{ik} = 1\}$. Let $\mathbb{L}$ be the set of all possible *assignment sets* and $\mathcal{L} \in \mathbb{L}$ be one such an assignment set. The value of x^S_{id} is a Boolean disjunction of the values at dimension d of all sources to which object i is assigned. The Boolean disjunction in the generation process of an x^S_{id} results in a probability for $x^S_{id} = 1$, which is strictly non-decreasing in the number of associated sources $|\mathcal{L}_i|$: If any of the sources in $\mathcal{L}_i$ emits a 1 in dimension d, then $x^S_{id} = 1$. Conversely, $x^S_{id} = 0$ requires that all contributing sources have emitted a 0 in dimension d.

Let β_{kd} be the probability that source k emits a 0 at dimension d: $\beta_{kd} := p(u_{kd} = 0)$. This parameter matrix $\beta \in [0,1]^{K\times D}$ allows us to simplify notation and to write

$$p_S\left(x^S_{id} = 0 \,|\, z_{i*},\beta\right) = \prod_{k=1}^{K} \beta_{kd}^{z_{ik}} \qquad \text{and } p_S\left(x^S_{id} = 1 \,|\, z_{i*},\beta\right) = 1 - p_S\left(x^S_{id} = 0 \,|\, z_{i*},\beta\right).$$

The parameter matrix β encodes these probabilities for all sources and dimensions. Employing the notion of assignment sets, one can interpret the product

$$\beta_{\mathcal{L}_i d} := \prod_{k=1}^{K} \beta_{kd}^{z_{ik}} \tag{2}$$

as the source of the assignment set $\mathcal{L}_i$. However, note that this interpretation differs from an actual single-assignment setting where $L := |\mathbb{L}|$ independent sources are assumed and must be inferred. Here, we only have $K \times D$ parameters β_{kd} whereas in single-assignment clustering, the number of source parameters would be $L \times D$, which can be up to $2^K \times D$. The expression $\beta_{\mathcal{L}_i d}$ is rather a 'proxy'-source, which we introduce just for notational convenience. The probability distribution of a x_{id} generated from this structure model given the assignments $\mathcal{L}_i$ and the sources β is then

$$p_S\left(x_{id}^S \mid \mathcal{L}_i, \beta\right) = \left(1 - \beta_{\mathcal{L}_i d}\right)^{x_{id}^S} \left(\beta_{\mathcal{L}_i d}\right)^{1 - x_{id}^S}. \tag{3}$$

Note that we include the empty assignment set in the hypothesis class, i.e. a data item i need not belong to any class. The corresponding row x_{i*}^S contains only zeros and any element with the value 1 in the input matrix is explained by the noise process.

In the following sections, we describe various noise models that alter the output of the structure model. The structure part of the model together with a particular noise process is illustrated in Figure 2.

3.1.1 STRUCTURE COMPLEXITY AND SINGLE-ASSIGNMENT CLUSTERING

In the general case, which is when no restrictions on the assignment sets are given, there are $L = 2^K$ possible assignment sets. If the number of clusters to which an object can be simultaneously assigned is bounded by M, this number reduces to $L = \sum_{m=0}^{M} \binom{K}{m}$.

The particular case with $M = 1$ provides a model variant that we call *Single-Assignment Clustering* (SAC). In order to endow SAC with the same model complexity as MAC, we provide it with L clusters. Each of the assignment sets is then identified with one of the clusters. The clusters are treated (and, in particular, updated) independently of each other by computing the cluster parameters $\beta_{\mathcal{L}*}$ for each $\mathcal{L}$, discarding the dependencies in the original formulation. The underlying generative model of SAC, as well as the optimality conditions for its parameters, can be obtained by treating all assignment sets $\mathcal{L}$ independently in the subsequent equations. With all centroids computed according to Equation 2, the single-assignment clustering model yields the same probability for the data as the multi-assignment clustering model.

3.2 Noise Models and their Relationship

In this section, we first present the *mixture noise model*, which interprets the observed data as a mixture of independent emissions from the structure part and a noise source. Each bit in the matrix can thus be generated either by the structure model or by an independent global noise process. We then derive a more general formulation for this noise model. Starting there, we derive the *flip model*, where some randomly chosen bits of the signal matrix $\mathbf{x}^S$ are flipped, either from 0 to 1 or from 1 to 0. The noisy-OR model (Pearl, 1988) is a special case of the flip noise model, allowing only flips from 0 to 1.

The different noise models have different parameters. We denote the noise parameters of a model α by Θ_N^α. The full set of parameters for structure and noise is then $\Theta^\alpha := (\beta, \Theta_N^\alpha)$. As additional notation, we use the indicator function $\mathbf{I}_{\{p\}}$ for a predicate p, defined as

$$\mathbf{I}_{\{p\}} := \begin{cases} 1 & \text{if } p \text{ is true} \\ 0 & \text{otherwise .} \end{cases}$$

3.2.1 MIXTURE NOISE MODEL

In the mixture noise model, each x_{id} is generated either by the signal distribution or by a noise process. The binary indicator variable ξ_{id} indicates whether x_{id} is a noisy bit ($\xi_{id} = 1$) or a signal bit ($\xi_{id} = 0$). The observed x_{id} is then generated by

$$x_{id} = (1 - \xi_{id}) x_{id}^S + \xi_{id} x_{id}^N \ ,$$

where the generative process for the signal bit x_{id}^S is either described by the deterministic rule in Equation 1 or by the probability distribution in Equation 3. The noise bit x_{id}^N follows a Bernoulli distribution that is independent of object index i and dimension index d:

$$p_N\left(x_{id}^N \,|\, r\right) = r^{x_{id}^N} (1-r)^{1-x_{id}^N} \ . \tag{4}$$

Here, r is the parameter of the Bernoulli distribution indicating the probability of a 1. Combining the signal and noise distributions, the overall probability of an observed x_{id} is

$$p_M^{\text{mix}}(x_{id} \,|\, \mathcal{L}_i, \beta, r, \xi_{id}) = p_N(x_{id} \,|\, r)^{\xi_{id}} \, p_S(x_{id} \,|\, \mathcal{L}_i, \beta)^{1-\xi_{id}} \ . \tag{5}$$

We assume ξ_{id} to be Bernoulli distributed with a parameter $\varepsilon := p(\xi_{id} = 1)$ called the *noise fraction*. The joint probability of x_{id} and ξ_{id} given the assignment matrix $\mathbf{z}$ and all parameters is thus

$$p_M^{\text{mix}}(x_{id}, \xi \,|\, \mathbf{z}, \beta, r, \varepsilon) = p_M(x_{id} \,|\, \mathbf{z}, \beta, r, \xi) \cdot \varepsilon^{\xi_{id}} (1-\varepsilon)^{1-\xi_{id}} .$$

Since different x_{id} are conditionally independent given the assignments $\mathbf{z}$ and the parameters Θ^{mix}, we have

$$p_M^{\text{mix}}(\mathbf{x}, \xi \,|\, \mathbf{z}, \beta, r) = \prod_{id} p_M^{\text{mix}}(x_{id}, \xi \,|\, \mathbf{z}, \beta, r) .$$

The noise indicators ξ_{id} cannot be observed. We therefore marginalize out all ξ_{id} to derive the probability of $\mathbf{x}$ as

$$\begin{aligned} p_M^{\text{mix}}(\mathbf{x} \,|\, \mathbf{z}, \beta, r, \varepsilon) &= \sum_{\{\xi\}} p_M^{\text{mix}}(\mathbf{x}, \xi \,|\, \mathbf{z}, \beta, r, \varepsilon) \\ &= \prod_{id} \left(\varepsilon \cdot p_N(x_{id}) + (1-\varepsilon) \cdot p_S(x_{id})\right) \ . \end{aligned}$$

The observed data $\mathbf{x}$ is thus a mixture between the emissions of the structure part (which has weight $1-\varepsilon$) and the noise emissions (with weight ε). Introducing the auxiliary variable

$$q_{\mathcal{L}_i d}^{\text{mix}} := p_M^{\text{mix}}(x_{id} = 1 \,|\, \mathbf{z}, \beta, r, \varepsilon) = \varepsilon r + (1-\varepsilon)(1 - \beta_{\mathcal{L}_i d})$$

to represent the probability that $x_{id} = 1$ under this model, we get a data-centric representation of the probability of $\mathbf{x}$ as

$$p_M^{\text{mix}}(\mathbf{x} \,|\, \mathbf{z}, \beta, r, \varepsilon) = \prod_{id} (x_{id} \ q_{\mathcal{L}_i d}^{\text{mix}} + (1 - x_{id}) \left(1 - q_{\mathcal{L}_i d}^{\text{mix}}\right)) \ . \tag{6}$$

The parameters of the mixture noise model are $\Theta_N^{\text{mix}} := (\varepsilon, r)$. Since ε and r are independent of d and i, we will refer to ε and r as parameters of a 'global' noise process.

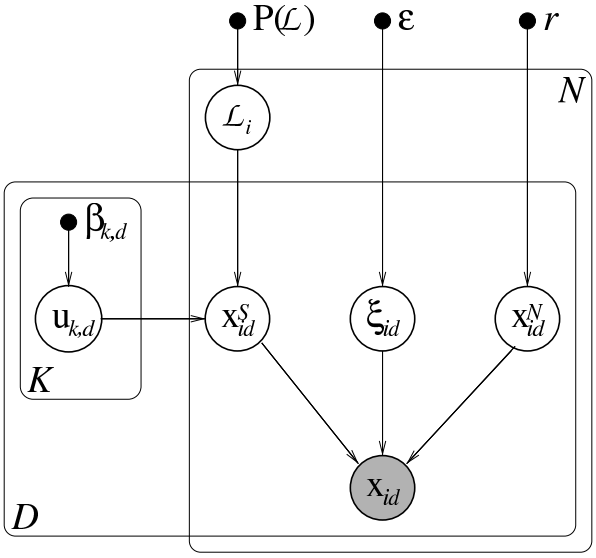


Figure 2: The generative model of Boolean MAC with mixture noise. $\mathcal{L}_i$ is the assignment set of object i, indicating which Boolean sources from $\mathbf{u}$ generated it. The bit ξ_{id} selects whether the noise-free bit x^S_{id} or the noise bit x^N_{id} is observed.

3.2.2 GENERALIZED NOISE MODEL

In this section, we generalize the mixture noise model presented above. Doing so, we achieve a generalized formulation that covers, among others, the mentioned noisy-OR model.

The overall generation process has two steps:

1. The signal part of the data is generated according to the sources, as described in Section 3.1. It is defined by the probability $p_S\left(x^S_{id} \mid \mathcal{L}_i, \beta\right)$ (Equation 3).

2. A noise process acts on the signal $\mathbf{x}^S$ and thus generates the observed data matrix $\mathbf{x}$. This noise process is described by the probability $p^\alpha(x_{id}|x^S_{id}, \Theta^\alpha_N)$, where α identifies the noise model and Θ^α_N are the parameters of the noise model α.

The overall probability of an observation x_{id} given all parameters is thus

$$p^\alpha_M\left(x_{id}|\mathcal{L}_i, \beta, \Theta^\alpha_N\right) = \sum_{x^S_{id}} p_S\left(x^S_{id} \mid \mathcal{L}_i, \beta\right) \cdot p^\alpha\left(x_{id}|x^S_{id}, \Theta^\alpha_N\right) \ .$$

3.2.3 MIXTURE NOISE MODEL

The mixture noise model assumes that each x_{id} is explained either by the structure model or by an independent global noise process. Therefore, the joint probability of $p^{\text{mix}}\left(x_{id}|x^S_{id}, \Theta^{\text{mix}}_N\right)$ can be factored as

$$p^{\text{mix}}\left(x_{id}|x^S_{id}, \Theta^{\text{mix}}_N\right) = p^{\text{mix}}_M\left(x_{id}|x^S_{id}, x^N_{id}, \xi_{id}\right) \cdot p^{\text{mix}}_N(x^N_{id}|r) \ ,$$

with

$$p^{\text{mix}}_M\left(x_{id}|x^S_{id}, x^N_{id}, \xi_{id}\right) = \left(\mathbf{I}_{\{x^S_{id}=x_{id}\}}\right)^{1-\xi_{id}} \left(\mathbf{I}_{\{x^N_{id}=x_{id}\}}\right)^{\xi_{id}} \ .$$

$p^S(x_{id}^S|\mathcal{L}_i,\beta)$ and $p_N^{\text{mix}}(x_{id}^N|r)$ are defined by Equation 3 and Equation 4 respectively. Summing out the unobserved variables x_{id}^S and x_{id}^N yields

$$\begin{aligned} p_M^{\text{mix}}(x_{id}|\mathcal{L}_i,\beta,r,\xi_{id}) &= \sum_{x_{id}^S=0}^{1}\sum_{x_{id}^N=0}^{1} p_M^{\text{mix}}\left(x_{id},x_{id}^S,x_{id}^N|\mathcal{L}_i,\beta,r,\xi_{id}\right) \\ &= p_S(x_{id}|\mathcal{L}_i,\beta)^{1-\xi_{id}}\cdot p_N^{\text{mix}}(x_{id}|r)^{\xi_{id}} \\ &= (1-\xi_{id})p_S(x_{id}|\mathcal{L}_i,\beta)+\xi_{id}p_N^{\text{mix}}(x_{id}|r)\ . \end{aligned}$$

Integrating out the noise indicator variables ξ_{id} leads to the same representation as in Equation 5.

3.2.4 FLIP NOISE MODEL

In contrast to the previous noise model, where the likelihood is a mixture of *independent* noise and signal distributions, the flip noise model assumes that the effect of the noise depends on the signal itself. The data is generated from the same signal distribution as in the mixture noise model. Individual bits are then randomly selected and flipped. Formally, the generative process for a bit x_{id} is described by

$$x_{id} = x_{id}^S \oplus \xi_{id}\ ,$$

where $\oplus$ denotes addition modulo 2. Again, the generative process for the structure bit x_{id}^S is described by either Equation 1 or Equation 3. The value of ξ_{id} indicates whether the bit x_{id}^S is to be flipped ($\xi_{id}=1$) or not ($\xi_{id}=0$). In a probabilistic formulation, we assume that the indicator ξ_{id} for a bit-flip is distributed according to $\xi_{id}\sim p(\xi_{id}|x_{id}^S,\varepsilon_0,\varepsilon_1)$. Thus, the probability of a bit-flip, given the signal and the noise parameters $(\varepsilon_0,\varepsilon_1)$, is

$$p(\xi_{id}|x_{id}^S,\varepsilon_0,\varepsilon_1) = \left(\varepsilon_1^{x_{id}^S}\varepsilon_0^{1-x_{id}^S}\right)^{\xi_{id}}\left((1-\varepsilon_1)^{x_{id}^S}(1-\varepsilon_0)^{1-x_{id}^S}\right)^{1-\xi_{id}}\ ,$$

with the convention that $0^0=1$. Given the flip indicator ξ_{id} and the signal bit x_{id}^S, the final observation is deterministic:

$$p_M^{\text{flip}}(x_{id}|\xi_{id},x_{id}^S) = x_{id}^{\mathbf{I}_{\{\xi_{id}\neq x_{id}^S\}}}(1-x_{id})^{\mathbf{I}_{\{\xi_{id}=x_{id}^S\}}}\ .$$

The joint probability distribution is then given by

$$p^{\text{flip}}\left(x_{id}|x_{id}^S,\Theta_N^{\text{flip}}\right) = \sum_{\xi_{id}=0}^{1} p_M^{\text{flip}}(x_{id}|\xi_{id},x_{id}^S)\cdot p(\xi_{id}|x_{id}^S,\varepsilon_0,\varepsilon_1)\ .$$

3.2.5 RELATION BETWEEN THE NOISE PARAMETERS

Our unified formulation of the noise models allows us to compare the influence of the noise processes on the clean signal under different noise models. We derive the parameters of the flip noise model that is equivalent to a given mixture noise model based on the probabilities $p^{\text{mix}}(x_{id}|x_{id}^S,\Theta_N^\alpha)$ and $p^{\text{flip}}(x_{id}|x_{id}^S,\Theta_N^\alpha)$, for the cases $(x_{id}=1,x_{id}^S=0)$ and $(x_{id}=0,x_{id}^S=1)$:

The mixture noise model with $\Theta_N^{\text{mix}}=(\varepsilon,r)$ is equivalent to the flip noise model with $\Theta_N^{\text{flip}}=(\varepsilon\cdot r,\varepsilon\cdot(1-r))$. Conversely, we have that the flip noise model with $\Theta_N^{\text{flip}}=(\varepsilon_0,\varepsilon_1)$ is equivalent to the **mixture noise model** with $\Theta_N^{\text{mix}}=\left(\varepsilon_0+\varepsilon_1,\frac{\varepsilon_0}{\varepsilon_0+\varepsilon_1}\right)$.

Hence the two noise-processes are just different representations of the same process. We therefore use only the mixture noise model in the remainder of this paper and omit the indicator α to differentiate between the different noise models.

3.2.6 OBJECT-WISE AND DIMENSION-WISE NOISE PROCESSES

In the following, we extend the noise model presented above. Given the equivalence of mix and flip noise, we restrict ourselves to the mixture noise model.

Dimension-wise Noise. Assume a separate noise process for every dimension d, which is parameterized by r_d and has intensity ε_d. We then have

$$p(\mathbf{x} \mid \mathbf{z}, \beta, \varepsilon) = \prod_{i,d} \left(\varepsilon_d r_d^{x_{id}} (1 - r_d)^{1 - x_{id}} + (1 - \varepsilon_d) (1 - \beta_{\mathcal{L}_i d})^{x_{id}} \beta_{\mathcal{L}_i d}^{1 - x_{id}} \right) .$$

Object-wise Noise. Now assume a separate noise process for every object i, which is parameterized by ε_i and r_i. As before, we have

$$p(\mathbf{x} \mid \mathbf{z}, \beta, \varepsilon) = \prod_{i,d} \left(\varepsilon_i r_i^{x_{id}} (1 - r_i)^{1 - x_{id}} + (1 - \varepsilon_i) (1 - \beta_{\mathcal{L}_i d})^{x_{id}} \beta_{\mathcal{L}_i d}^{1 - x_{id}} \right) .$$

Note that these local noise models are very specific and could be used in the following application scenarios. In role mining, some permissions are more critical than others. Hence it appears reasonable to assume a lower error probability for the dimension representing, for example, root access to a central database server than for the dimension representing the permission to change the desktop background image. However we observed experimentally that the additional freedom in these models often leads to an over-parametrization and thus worse overall results. This problem could possibly be reduced by introducing further constraints on the parameters, such as a hierarchical order.

4. Inference

We now describe an inference algorithm for our model. While the parameters are ultimately inferred according to the maximum likelihood principle, we use the optimization method of *deterministic annealing* presented in Buhmann and Kühnel (1993) and Rose (1998). In the following, we specify the deterministic annealing scheme used in the algorithm. In Section 4.2 we then give the characteristic magnitudes and the update conditions in a general form, independent of the noise model. The particular update equations for the mixture model are then derived in detail in Section 4.3.

4.1 Annealed Parameter Optimization

The likelihood of a data matrix $\mathbf{x}$ (Equation 6) is highly non-convex in the model parameters and a direct maximization of this function will likely be trapped in local optima. Deterministic annealing is an optimization method that parameterizes a smooth transition from the convex problem of maximizing the entropy (i.e. a uniform distribution over all possible clustering solutions) to the problem of minimizing the empirical risk R. The goal of this heuristic is to reduce the risk of being trapped in a local optimum. Such methods are also known as continuation methods (see Allgower and Georg, 1980). In our case, R is the negative log likelihood. Formally, the Lagrange functional

$$F := -T \log Z = \mathbb{E}_G [R] - TH$$

is introduced, with Z being the *partition function* over all possible clustering solutions (see Equation 10), and G denotes the Gibbs distribution (see Equation 9 and Equation 8). The Lagrange parameter T (called the *computational temperature*) controls the trade-off between entropy maximization and minimization of the empirical risk. Minimizing F at a given temperature T is equivalent to constraint minimization of the empirical risk R with a lower limit on the entropy H. In other words, H is a uniform prior on the likelihood of the clustering solutions. Its weight decreases as the computational temperature T is incrementally reduced.

At every temperature T, a gradient-based expectation-maximization (EM) step computes the parameters that minimize F. The E-step computes the risks $R_{i\mathcal{L}}$ (Equation 7) of assigning data item i to the assignment set $\mathcal{L}$. The corresponding responsibilities $\gamma_{i\mathcal{L}}$ (Equation 8) are computed for all i and $\mathcal{L}$ based on the current values of the parameters. The M-step first computes the optimal values of the noise parameters. Then it uses these values to compute the optimal source parameters β. The individual steps are described in Section 4.3.

We determine the initial temperature as described in Rose (1998) and use a constant cooling rate ($T \leftarrow \vartheta \cdot T$, with $0 < \vartheta < 1$) . The cooling is continued until the responsibilities $\gamma_{i\mathcal{L}}$ for all data items i peak sharply at single assignment sets $\mathcal{L}_i$.

4.2 Characteristic Magnitudes and Update Conditions

Following our generative approach to clustering, we aim at finding the maximum likelihood solution for the parameters. Taking the logarithm of the likelihood simplifies the calculations as products become sums. Also, the likelihood function conveniently factors over the objects and features enabling us to investigate the risk of objects individually. We define the *empirical risk* of assigning an object i to the set of clusters $\mathcal{L}$ as the negative log-likelihood of the feature vector x_{i*} being generated by the sources contained in $\mathcal{L}$:

$$R_{i\mathcal{L}} := \log p(x_{i\cdot}|\mathcal{L}_i,\Theta) = -\sum_d \log\left(x_{id}\,(1-q_{\mathcal{L}d}) + (1-x_{id})q_{\mathcal{L}d}\right) \ . \tag{7}$$

The *responsibility* $\gamma_{i\mathcal{L}}$ of the assignment-set $\mathcal{L}$ for data item i is given by

$$\gamma_{i\mathcal{L}} := \frac{\exp\left(-R_{i\mathcal{L}}/T\right)}{\sum_{\mathcal{L}'\in\mathbb{L}} \exp\left(-R_{i\mathcal{L}'}/T\right)} \ . \tag{8}$$

The matrix γ defines a probability distribution over the space of all clustering solutions. The expected *empirical risk* $\mathbb{E}_G[R]$ of the solutions under this probability distribution G is

$$\mathbb{E}_G[R_{i\mathcal{L}}] = \sum_i \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} R_{i\mathcal{L}} \ . \tag{9}$$

Finally, the *state sum* Z and the *free energy* F are defined as follows.

$$Z := \prod_i \sum_{\mathcal{L}} \exp\left(-R_{i\mathcal{L}}/T\right) \tag{10}$$

$$F := -T\log Z = -T\sum_i \log\left(\sum_{\mathcal{L}} \exp\left(-R_{i\mathcal{L}}/T\right)\right)$$

Given the above, we derive the updates of the model parameters based on the first-order condition of the free energy F. We therefore introduce the generic model parameter θ, which stands for

any of the model parameters, i.e. $\theta \in \{\beta_{\mu\nu}, \varepsilon_0, \varepsilon_1, \varepsilon, r\}$. Here, μ is some particular value of source index k and ν is some particular value of dimension index d. Using this notation, the derivative of the free energy with respect to θ is given by

$$\frac{\partial F}{\partial \theta} = \sum_i \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} \frac{\partial R_{i\mathcal{L}}}{\partial \theta} = \sum_i \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} \sum_d \frac{(1-2x_{id}) \frac{\partial q_{\mathcal{L}d}}{\partial \theta}}{x_{id}(1-q_{\mathcal{L}d}) + (1-x_{id}) q_{\mathcal{L}d}} .$$

4.3 Update Conditions for the Mixture Noise Model

Derivatives for the mixture noise model ($\theta \in \{\beta_{\mu\nu}, \varepsilon, r\}$) are:

$$\frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial \beta_{\mu\nu}} = (1-\varepsilon)\, \beta_{\mathcal{L}\setminus\{\mu\},d}\, \mathbf{I}_{\{\nu=d\}} \mathbf{I}_{\{\mu\in\mathcal{L}\}}, \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial \varepsilon} = 1 - r - \beta_{\mathcal{L}d}, \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial r} = -\varepsilon.$$

This results in the following first-order conditions for the mixture noise model:

$$\frac{\partial F^{\text{mix}}}{\partial \beta_{\mu\nu}} = (1-\varepsilon) \sum_{\mathcal{L}\mu\in\mathcal{L}} \beta_{\mathcal{L}\setminus\{\mu\},\nu} \left\{ \frac{\sum_{i:x_{i\nu}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}\nu})} - \frac{\sum_{i:x_{i\nu}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1-\varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}\nu})} \right\} = 0,$$

$$\frac{\partial F^{\text{mix}}}{\partial \varepsilon} = \sum_d \left\{ \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L}d}) \sum_{i:x_{id}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}d})} - \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L}d}) \sum_{i:x_{id}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1-\varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}d})} \right\} = 0,$$

$$\frac{\partial F^{\text{mix}}}{\partial r} = \varepsilon \sum_d \left\{ \sum_{\mathcal{L}} \frac{\sum_{i:x_{id}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1-\varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}d})} - \sum_{\mathcal{L}} \frac{\sum_{i:x_{id}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}d})} \right\} = 0.$$

There is no analytic expression for the solutions of the above equations, the parameters $\beta_{\mu\nu}$, ε, and r are thus determined numerically. In particular, we use Newton's method to determine the optimal values for the parameters. We observed that this method rapidly converges, usually needing at most 5 iterations.

The above equations contain the optimality conditions for the single-assignment clustering (SAC) model as a special case. As only assignment sets $\mathcal{L}$ with one element are allowed in this model, we can globally substitute $\mathcal{L}$ by k and get $\beta_{\mathcal{L}*} = \beta_{k*}$. Furthermore, since 1 is the neutral element for multiplication, we get $\beta_{\mathcal{L}\setminus\{\mu\},\nu} = 1$.

In the noise-free case, the value for the noise fraction is $\varepsilon = 0$. This results in a significant simplification of the update equations.

5. Experiments

In this section, we first introduce the measures that we employ to evaluate the quality of clustering solutions. Afterwards, we present results on both synthetic and real-world data.

5.1 Evaluation Criteria

For synthetic data, we evaluate the estimated sources by their Hamming distance to the true sources being used to generate the data. For real-world data, the appropriate evaluation criteria depend on the task. Independent of the task, the generalization ability of a solution indicates how well the solution fits to the unknown underlying probability distribution of the data. Moreover, as argued in Frank et al. (2010), the ability of a solution to generalize to previously unseen users is the appropriate

quality criterion for the role mining problem. In the following, we introduce these two measures, parameter mismatch and generalization ability.

The following notation will prove useful. We denote by $\hat{\mathbf{z}}$ and $\hat{\mathbf{u}}$ the estimated decomposition of the matrix $\mathbf{x}$. The reconstruction of the matrix based on this decomposition is denoted by $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} := \hat{\mathbf{z}} * \hat{\mathbf{u}}$. Furthermore, in experiments with synthetic data, the signal part of the matrix is known. As indicated in Section 3, it is denoted by $\mathbf{x}^S$.

5.1.1 PARAMETER MISMATCH

Experiments with synthetic data allow us to compare the values of the true model parameters with the inferred model parameters. We report below on the accuracies of both the estimated centroids $\hat{\mathbf{u}}$ and the noise parameters.

To evaluate the accuracy of the centroid estimates, we use the average Hamming distance between the true and the estimated centroids. In order to account for the arbitrary numbering of clusters, we permute the centroid vectors u_{k*} with a permutation $\pi(k)$ such that the estimated and the true centroids agree best. Namely,

$$a(\hat{\mathbf{u}}) := \frac{1}{K \cdot D} \min_{\pi \in P_K} \sum_{k=1}^{K} \left|\left| u_{k*} - \hat{u}_{\pi(k)*} \right|\right| \ ,$$

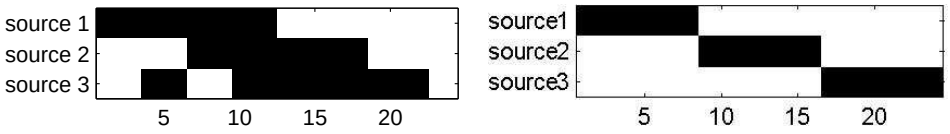
where P_K denotes the set of all permutations of K elements. Finding the $\pi \in P_K$ that minimizes the Hamming distance involves solving the assignment problem, which can be calculated in polynomial time using the Hungarian algorithm of Kuhn (2010). Whenever we know the true model parameters, we will assess methods based on parameter mismatch, always reporting this measure in percent.

5.1.2 GENERALIZATION ERROR

For real world data, the true model parameters are unknown and there might even exist a model mismatch between the learning model and the true underlying distribution that generated the input data set $\mathbf{x}^{(1)}$. Still, one can measure how well the method infers this distribution by testing if the estimated distribution generalizes to a second data set $\mathbf{x}^{(2)}$ that has been generated in the same way as $\mathbf{x}^{(1)}$. To measure this generalization ability, we first randomly split the data set along the objects into a training set $\mathbf{x}^{(1)}$ and a validation set $\mathbf{x}^{(2)}$. Then we learn the factorization $\hat{\mathbf{z}}$, $\hat{\mathbf{u}}$ based on the training set and transfer it to the validation set.

Note that the transfer of the learned solution to the validation set is not as straight-forward in such an unsupervised scenario as it is in classification. For transferring, we use the method proposed by Frank et al. (2011). For each object i in $\mathbf{x}^{(2)}$, we compute its nearest neighbor $\psi_{NN}(i)$ in $\mathbf{x}^{(1)}$ according to the Hamming distance. We then create a new matrix $\mathbf{z}'$ defined by $\mathbf{z}'_{i*} = \hat{\mathbf{z}}_{\psi_{NN}(i)*}$ for all i. As a consequence, each validation object is assigned to the same set of sources as its nearest neighbor in the training set. The possible assignment sets as well as the source parameters are thereby restricted to those that have been trained without seeing the validation data. The generalization error is then

$$G(\hat{\mathbf{z}}, \hat{\mathbf{u}}, \mathbf{x}^{(2)}, \psi_{NN}) := \frac{1}{N^{(2)} \cdot D} \left\| \mathbf{x}^{(2)} - \mathbf{z}' * \hat{\mathbf{u}} \right\|,$$
$$\text{with } \mathbf{z}' = \left(\hat{\mathbf{z}}_{\psi_{NN}(1)*}, \hat{\mathbf{z}}_{\psi_{NN}(2)*}, \ldots, \hat{\mathbf{z}}_{\psi_{NN}(N^{(2)})*} \right)^T \ ,$$



(a) Overlapping Sources (b) Orthogonal Sources

Figure 3: Overlapping sources (left) and orthogonal sources (right) used in the experiments with synthetic data. Black indicates a 1 and white a 0 for the corresponding matrix element. In both cases, the three sources have 24 dimensions.

where $N^{(2)}$ is the number of objects in the validation data set and $*$ is the Boolean matrix product as defined in Equation 1. This measure essentially computes the fraction of wrongly predicted bits in the new data set.

As some of the matrix entries in $\mathbf{x}^{(2)}$ are interpreted as noise, it might be impossible to reach a generalization error of 0%. However, this affects all methods and all model variants. Moreover, we are ultimately interested in the total order of models with respect to this measure and not in their absolute scores. Since we assume that the noise associated with the features of different objects is independent, we deduce from a low generalization error that the algorithm can infer sources that explain—up to residual noise—the features of new objects from the same distribution. In contrast, a high generalization error implies that the inferred sources wrongly predict most of the matrix entries and thus indicates overfitting.

Note that the computation of generalization error differs from the approach taken in Streich et al. (2009). There, only $\hat{\mathbf{u}}$ is kept fixed, and $\hat{\mathbf{z}}$ is ignored when computing the generalization error. The assignment sets $\mathbf{z}'$ of the new objects are recomputed by comparing all source combinations with a fraction κ of the bits of these objects. The generalization error is the difference of the remaining $(1-\kappa)$ bits to the assigned sources. In our experiments on model-order selection, this computation of generalization error led to overfitting. As $\mathbf{z}'$ was computed independently from $\hat{\mathbf{z}}$, fitting all possible role combinations to the validation data, it supports tuning one part of the solution to this data. With the nearest neighbor-based transfer of $\hat{\mathbf{z}}$, which is computed without using the validation set, this is not possible. Overfitting is therefore detected more reliably than in Streich et al. (2009).

In order to estimate the quality of a solution, we use parameter mismatch in experiments with synthetic data and generalization error in experiments with real data.

5.2 Experiments on Synthetic Data

This section presents results from several experiments on synthetic data where we investigate the performance of different model variants and other methods. Our experiments have the following setting in common. First, we generate data by assigning objects to one or more Boolean vectors out of a set of predefined sources. Unless otherwise stated, we will use the generating sources as depicted in Figure 3. Combining the emissions of these sources via the *OR* operation generates the structure of the objects. Note that the sources can overlap, i.e. multiple sources emit a 1 at a particular dimension. In a second step, we perturb the data set by a noise process.

With synthetic data, we control all parameters, namely the number of objects and sources, the geometry of the Boolean source vectors (i.e. we vary them between overlapping sources and or-

thogonal sources), the fraction of bits that are affected by the noise process, and the kind of noise process. Knowing the original sources used to generate the data set enables us to measure the accuracy of the estimators, as described in Section 5.1. The goal of these experiments is to investigate the behavior of different methods under a wide range of conditions. The results will help us in interpreting the results on real-world data in the next section.

We repeat all experiments ten times, each time with different random noise. We report the median (and 65% percentiles) of the accuracy over these ten runs.

5.2.1 COMPARISON OF MAC WITH OTHER CLUSTERING TECHNIQUES

The main results of the comparison between MAC and other clustering techniques are shown in Figure 4. Each panel illustrates the results of one of the methods under five different experimental setups. We generate 50 data items from each single source as well as from each combination of two sources. Furthermore, 50 additional data items are generated without a source, i.e. they contain no structure. This experimental setting yields 350 data items in total. The overlapping sources are used as shown in Figure 3(a), and the structure is randomly perturbed by a mixture noise process. The probability of a noisy bit being 1 is kept fixed at $r = 0.5$, while the fraction of noisy bits, ε, varies between 0% and 99%. The fraction of data from multiple sources is 50% for the experiments plotted with square markers. Experiments with only 20% (80%) of the data are labeled with circles (with stars). Furthermore, we label experiments with orthogonal sources (Figure 3(b)) with 'x'. Finally, we use '+' labels for results on data with a noisy-OR noise process, i.e. $r = 1$.

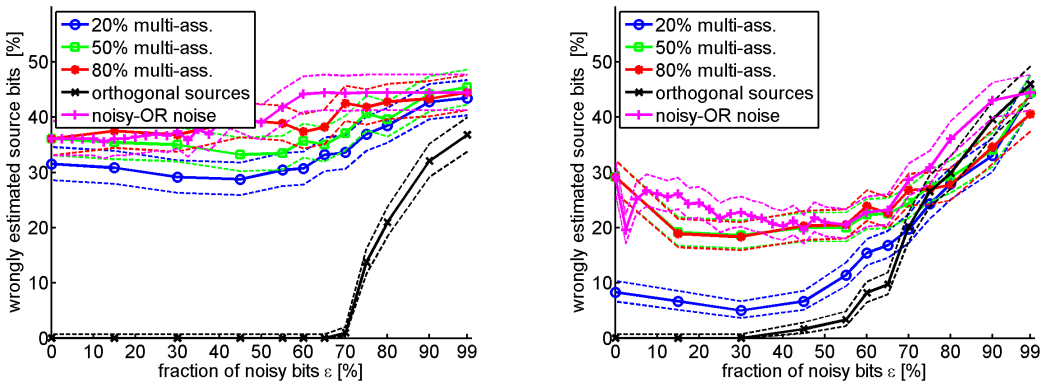
5.2.2 BINARY INDEPENDENT COMPONENT ANALYSIS (BICA)

BICA has a poor parameter accuracy in all experiments with data from overlapping clusters. This behavior is caused by the assumption of orthogonal sources, which fails to hold for such data. BICA performs better on data that was modified by the symmetric mixture noise process than on data from a noisy-OR noise process. Since BICA does not have a noise model, the data containing noise from the noisy-OR noise process leads to extra 1s in the source estimators. This effect becomes important when the noise fraction rises above 50%. We observe that, overall, the error rate does not vary much for overlapping sources.

The effect of the source geometry is particularly noticeable. On data generated by orthogonal sources, i.e. when the assumption of BICA is fulfilled, the source parameters are perfectly reconstructed for noise levels up to 65%. Only for higher noise levels, does the accuracy break down. The assumption of orthogonal source centroids is essential for BICA's performance as the poor results on data with non-orthogonal sources show. As more data items are generated by multiple, non-orthogonal sources, the influence of the mismatch between the assumption underlying BICA and the true data increases. This effect explains why the source parameter estimators for non-orthogonal centroids become less accurate when going from 20% of multi-assignments to 80%.

5.2.3 DISCRETE BASIS PROBLEM SOLVER (DBPS)

Figure 4(b) shows that this method yields accurate source parameter estimators for data generated by orthogonal sources, and, to a lesser degree, for data sets that contain a small percentage of multi-assignment data. As the fraction of multi-assignment data increases, the accuracy of DBPS decreases.



(a) Accuracy of BICA

(b) Accuracy of DBPS

(c) Accuracy of INO

(d) Accuracy of MAC

Figure 4: Accuracy of source parameter estimation for five different types of data sets in terms of mismatch to the true sources. We use (circle, square, star) symmetric Bernoulli noise and overlapping sources with three different fractions of multi-assignment data, (x) orthogonal sources and symmetric noise, and (+) overlapping sources and a noisy-or noise process. Solid lines indicate the median over 10 data sets with random noise and dashed lines show the 65% confidence intervals.

The reason for the low accuracy on multi-assignment data arises from the greedy optimization of DBPS. It selects a new source out of a candidate set such that it can explain as many objects as possible by the newly chosen source. In a setting where most of the data is created by a combination of sources, DBPS will first select a single source that equals the disjunction of the true sources because this covers the most 1s. We call this effect *combination-singlet confusion*. It is a special case of the typical problem of forward selection. Lacking a generative model for source-combinations, DBPS cannot use the observation of objects generated by source-combinations to gather evidence for the individual sources. As a consequence, the first selected source estimates fit to the source-combinations and not to the true individual sources. Often, the last selected sources are left empty, leading to a low estimation accuracy.

Note the effect of a small amount of noise on the accuracy of DBPS. The clear structure of the association matrix is perturbed, and the candidates might contain 0s in some dimensions. As a result, the roles selected in the second and subsequent steps are non-empty, making the solution more similar to the true sources. This results in the interesting effect where the accuracy increases when going from noise-free matrices to those with small amount of noise (for higher noise, it decreases again because of overfitting).

DBPS obtains accurate estimators in the setting where the data is generated by orthogonal data (labeled 'x'). Here, the candidate set does not contain sources that correspond to combinations of true sources, and the greedy optimization algorithm can only select a candidate source that corresponds to a true single source. DBPS thus performs best with respect to source parameter estimation when the generating sources are orthogonal. In contrast to BICA, which benefits from the explicit assumption of orthogonal sources, DBPS favors such sources because of the properties of its greedy optimizer.

5.2.4 INFINITE NOISY-OR (INO)

The infinite noisy-OR is a non-parametric Bayesian method. To obtain a single result, we approximate the a posteriori distribution by sampling and then choose the parameters with highest probability. This procedure estimates the maximum a posterior solution. Furthermore, in contrast to BICA, DBPS, and all MAC variants, INO determines the number of sources by itself and might obtain a value different than the number of sources used to generate the data. If the number inferred by INO is smaller than the true number, we choose the closest true sources to compute the parameter mismatch. If INO estimates a larger set of sources than than the true one, the best-matching INO sources are used. This procedure systematically overestimates the accuracy of INO, whereas INO actually solves a harder task that includes model-order selection. A deviation between the estimated number of sources and the true number mainly occurs at the mid-noise level (approximately 30% to 70% noisy bits).

In all settings, except the case where 80% of the data items are generated by multiple sources, INO yields perfect source estimators up to noise levels of 30%. For higher noise levels, its accuracy rapidly drops. While the generative model underlying INO enables this method to correctly interpret data items generated by multiple sources, a high percentage (80%) of such data poses the hardest problem for INO.

For noise fractions above approximately 50%, the source parameter estimators are only slightly better than random in all settings. On such data, the main influence comes from the noise, while the contribution of different source combinations is no longer important.

5.2.5 MULTI-ASSIGNMENT CLUSTERING (MAC)

The multi-assignment clustering method yields perfect parameter estimators for noise levels up to 40% in all experimental settings considered. The case with 80% of multi-assignment data is the most challenging one for MAC. When only 50% or 20% of the data items are generated by more than one source, the parameter estimates are accurate for noise levels up to 55% or 60% of noisy bits. When few data items originate from a single source, MAC fails to separate the contributions of the individual sources. These single-source data items function as a kind of 'anchor' and help the algorithm to converge to the true parameters of the individual sources. For very high noise levels (90% and above), the performance is again similar for all three ratios of multi-assignment data.

In comparison to the experiments with overlapping sources described in the previous paragraph, MAC profits from orthogonal centroids and yields superior parameter accuracy for noise levels above 50%. As for training data with little multi-assignment data, orthogonal centroids simplify the task of disentangling the contributions of the individual sources. When a reasonable first estimate of the source parameters can be derived from single-assignment data, a 1 in dimension d of a data item is explained either by the unique source which has a high probability of emitting a 1 in this dimension, or by noise—even if the data item is assigned to more than one source.

Interestingly, MAC's accuracy peaks when the noise is generated by a noisy-OR noise process. The reason is that observing a 1 at a particular bit creates a much higher entropy of the parameter estimate than observing a 0: a 1 can be explained by all possible combinations of sources having a 1 at this position, whereas a 0 gives strong evidence that all sources of the object are 0. As a consequence, a wrong bit being 0 is more severe than a wrong 1. The wrong 0 forces the source estimates to a particular value whereas the wrong 1 distributes its 'confusion' evenly over the sources. As the noisy-OR creates only 1s, it is less harmful. This effect could, in principle, also help other methods if they managed to appropriately disentangle combined source parameters.

5.2.6 PERFORMANCE OF MAC VARIANTS

We carry out inference with the MAC model and the corresponding Single-Assignment Clustering (SAC) model, each with and without the mixture noise model. These model variants are explained in Section 3.1.1. The results illustrated in Figure 5 are obtained using data sets with 350 objects. The objects are sampled from the overlapping sources depicted in Figure 3(a). To evaluate the solutions of the SAC variants in a fair way, we compare the estimated sources against all combinations of the true sources.

5.2.7 INFLUENCE OF SIGNAL MODEL AND NOISE MODEL

As observed in Figure 5, the source parameter estimators are much more accurate when a noise model is employed. For a low fraction of noisy bits ($< 50\%$), the estimators with a noise model are perfect, but are already wrong for 10% noise when not using a noise model. When inference is carried out using a model that lacks the ability to explain individual bits by noise, the entire data set must be explained with the source estimates. Therefore, the solutions tend to overfit the data set. With a noise model, a distinction between the structure and the irregularities in the data is possible and allows one to obtain more accurate estimates for the model parameters.

Multi-Assignment Clustering (MAC) provides more accurate estimates than SAC and the accuracy of MAC breaks down at a higher noise level than the accuracy of SAC. The reason is twofold. First, the ratio of the number of observations per model parameter differs for both model variants. MAC explains the observations with combinations of sources whereas SAC assigns each object to a single source only. SAC therefore uses only those objects for inference that are exclusively assigned to a source, while MAC also uses objects that are simultaneously assigned to other sources. Second, using the same source in different combinations with other sources implicitly provides a consistency check for the source parameter estimates. SAC lacks this effect as all source parameters are independent. The difference between MAC and SAC becomes apparent when the data set is noisy. For low fractions of noise, the accuracy is the same for both models.

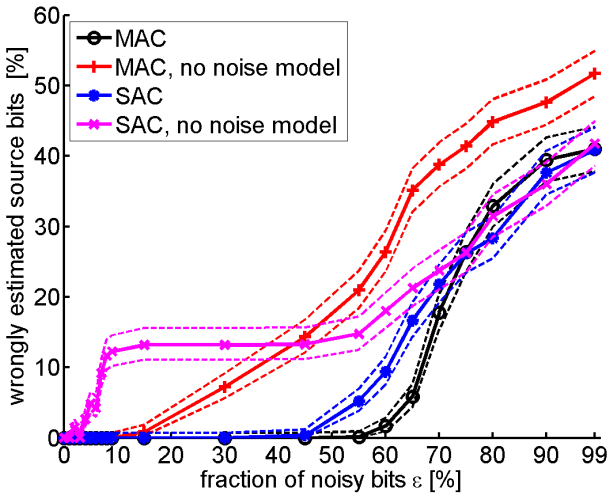


Figure 5: Average Hamming distance between true and estimated source prototypes for MAC and SAC with and without noise models respectively.

We conducted the same experiments on data sets that are ten times larger and observed the same effects as the ones described above. The sharp decrease in accuracy is shifted to higher noise levels and appears in a smaller noise window when more data is available.

5.3 Experiments on Role Mining Data

To evaluate the performance of our algorithm on real data, we apply MAC to mining RBAC roles from access control configurations. We first specify the problem setting and then report on our experimental results.

5.3.1 SETTING AND TASK DESCRIPTION

As explained in Section 2, role mining must find a suitable RBAC configuration based on a binary user-permission assignment matrix $\mathbf{x}$. An RBAC configuration is the assignment of K roles to permissions and assignments of users to these roles. A user can have multiple roles, and the bit-vectors representing the roles can overlap. The inferred RBAC configuration is encoded by the Boolean assignment matrices $(\hat{\mathbf{z}}, \hat{\mathbf{u}})$.

We emphasize the importance of the generalization ability of the RBAC configuration: The goal is not primarily to compress the existing user-permission matrix $\mathbf{x}$, but rather to infer a set of roles that generalizes well to new users. An RBAC system's security and maintainability improve when the roles do not need to be redefined whenever there is a small change in the enterprise, such as a new user being added to the system or users changing positions within the enterprise. Moreover, as previously explained, it is desirable that the role mining step identifies exceptional permission assignments. Such exceptional assignments are represented by the noise component of the mixture model. In practice, one must check whether the suspected erroneous bits are really errors or if they were (and still are!) intended. Without additional input, one can at most distinguish between reg-

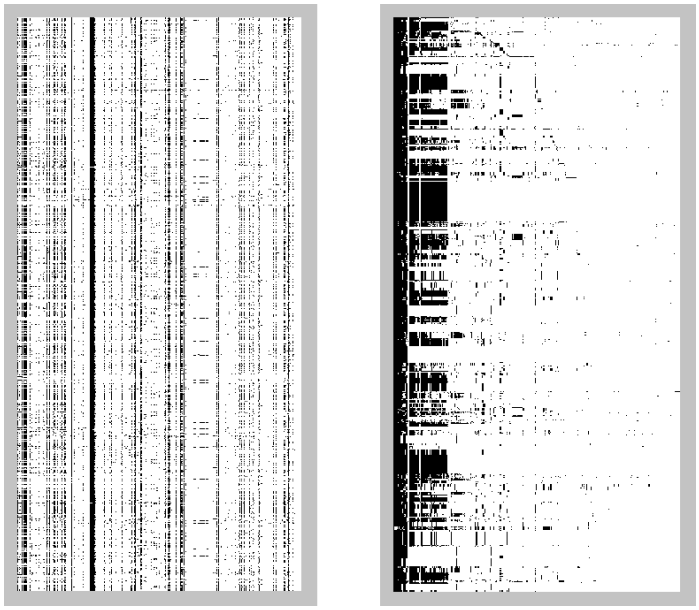
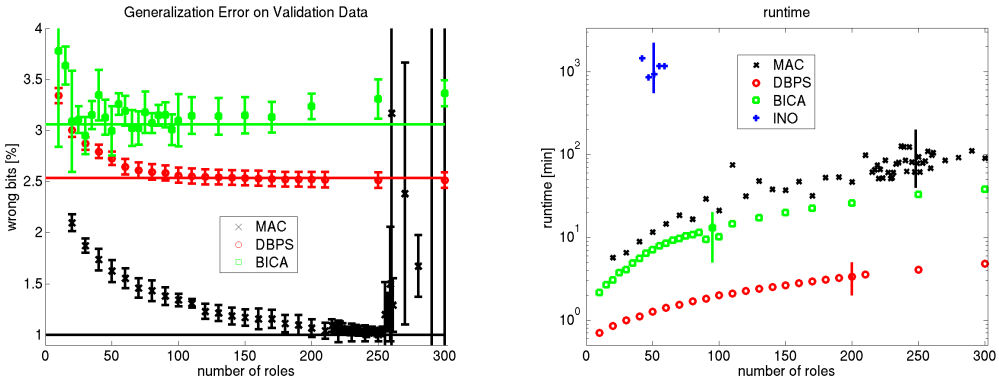


Figure 6: A 2400×500 part of the data matrix used for model-order selection. Black dots indicate a 1 at the corresponding matrix element and white dots indicate a 0. The full data matrix has size 4900×1300. Rows and columns of the right matrix are reordered such that users with the same role set and permissions of the same role are adjacent to each other, if possible. Note that there does not exist a permutation that satisfies this condition for all users and permissions simultaneously.

ularities and irregularities. This is a problem for all role mining algorithms: The interpretation of the irregularities and any subsequent corrections must be performed by a domain expert. However, minimizing the number of suspicious bits and finding a decomposition that generalizes well is already a highly significant advantage over manual role engineering. See Frank et al. (2010) for an extended discussion of this point.

In our experiments, we use a data set from our collaborator containing the user-permission assignment matrix of $N = 4900$ users and $D = 1300$ permissions. We will call this data set C_{orig} in subsequent sections. A part of this data matrix is depicted in Figure 6. Additionally, we use the publicly available access control configurations from HP labs published by Ene et al. (2008).

To evaluate the different methods on more complex data with a higher noise level, we generate another data set $\overline{\mathbf{x}}$ as follows: For the original user-permission assignment matrix of C_{orig} we combine the first 500 columns and the second 500 columns by an element-wise *OR* operation to give the structure part $\overline{\mathbf{x}}^S$. Afterwards, we replace 33% of the matrix entries by random bits to yield the modified matrix $\overline{\mathbf{x}}$. This matrix exhibits both a higher structural complexity and a substantially increased noise level than the original matrix $\mathbf{x}$. We will call this modified data set C_{mod}. We explain the individual steps of the experiments based on C_{orig} as a running example. All other experiments, those on C_{mod} and on the HP data, are carried out in the same way.



(a) Generalization Error (b) run-time

Figure 7: Left: Generalization error on the hold-out validation set in terms of wrongly predicted bits versus the number of roles. The other external parameters for BICA and DBPS are determined by exhaustive search. Right: Run-time versus number of roles on a 2400 $\times$ 500 access-control matrix. The selected number of roles is highlighted by vertical lines.

5.3.2 MODEL-ORDER SELECTION

INO is a non-parametric model that can compute probabilities over the infinite space of all possible binary assignment matrices. It is therefore able to select the number of roles K during inference and needs no external input. For DBPS, BICA, and MAC, the number of roles must be externally selected and for DBPS and BICA, also rounding thresholds and approximation weights must be tuned. The number of roles K is the most critical parameter.

As a principle for guiding these model selection tasks, we employ the generalization error as defined in Section 5.1. Out of the total of 4900 users from C_{orig}, we use five-fold cross-validation on a subset of 3000 users. In each step, we split them into 2400 users for training the model parameters and 600 users for validating them, such that each user occurs once in the validation set and four times in the training set. The number of permissions used in this experiment is 500. We increase the number of roles until the generalization error increases. For a given number of roles, we optimize the remaining parameters (of DBPS and BICA) on the training sets and validation sets. For continuous parameters, we quantize the parameter search-space into 50 equally spaced values spanning the entire range of possible parameter values.

To restrict the cardinality of the assignment sets (for MAC), we make one trial run with a large number of roles and observe how many of the roles are involved in role combinations. A role that is involved in role combinations is at least once assigned to a user together with at least one other role. In our experiments on C_{orig}, for instance, 10% of $K = 100$ roles are used in role combinations and no roles appear in combinations with more than two roles. Therefore, for subsequent runs of the algorithm, we set $M = 2$ and limit the number of roles that can belong to a multiple assignment set to 10% of K. For large K, such a restriction drastically reduces the run-time as the solution space is much smaller than the space of all possible role combinations. See Section 5.4 for an analysis of the run-time complexity of all investigated methods.

Restricting the number of roles that can belong to a multiple assignment set risks having too few role combinations available to fit the data at hand. However, such circumstances cannot lead to underfitting when K is still to be computed in the cross-validation phase. In the worst case, an unavailable role combination would be substituted by an extra single role.

The performance of the three methods MAC, DBPS, and BICA as a function of the number of roles is depicted in Figure 7(a), left. The different models favor a substantially different number of roles on this data set (and also on other data sets, see Table 1). For MAC, there is a very clear indication of overfitting for $K > 248$. For DBPS, the generalization error monotonically decreases for $K < 150$. As K further increases, the error remains constant. In the cross-validation phase, the internal threshold parameter of DPBS is adapted to minimize the generalization error. This prevents excessive roles from being used as, with the optimal threshold, they are left empty. We select $K = 200$ for DBPS, where more roles provide no improvement. INO selects 50 roles on average. BICA favors a considerably smaller number of roles, even though the signal is not as clear. We select $K = 95$ for BICA, which is the value that minimizes the median generalization error on the validation sets.

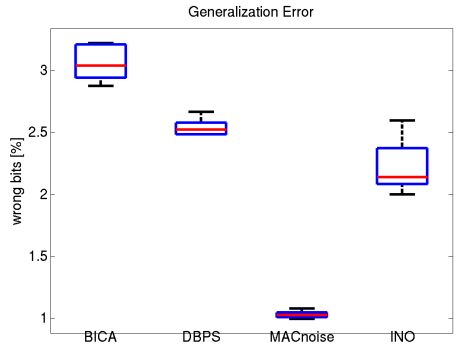
5.3.3 RESULTS OF DIFFERENT METHODS

The results of the generalization experiments for the four methods MAC, DBPS, BICA, and INO are depicted in Figure 8. Overall, all methods have a very low generalization error on the original data set. The error spans from 1% to 3% of the predicted bits. This result indicates that, on a global scale, C_{orig} has a rather clean structure. It should be stressed that most permissions in the input data set are only rarely assigned to users, whereas some are assigned to almost everyone, thereby making up most of the 1s in the matrix (see a part of the data set in Figure 6). Therefore, the most trivial role set where roles are assigned no permissions already yields a generalization error of 13.5%. Assigning everyone to a single role that contains all permissions that more than 50% percent of the users have, achieves 7.1%. One should keep this baseline in mind when interpreting the results.

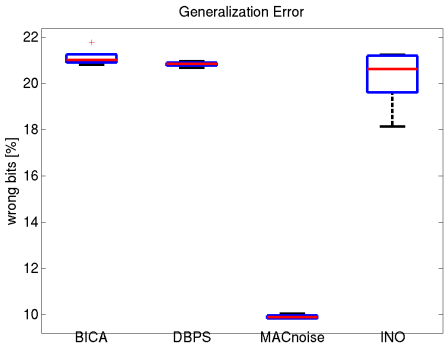
INO, DBPS, and BICA span a range from 2.2% generalization error to approximately 3% with significant distance to each other. MAC achieves the lowest generalization error with slightly more than 1%. It appears that INO is misled by its noisy-OR noise model, which seems to be inappropriate in this case. MAC estimates the fraction of noisy bits by $\hat{\varepsilon} \approx 2.8\%$ and the probability for a noisy bit to be 1 by $\hat{r} \approx 20\%$. This estimate clearly differs from a noisy-OR noise process (which would have $r = 1$). With more than 3% generalization error, BICA performs worst. As all other methods estimate a considerable centroid overlap, the assumption of orthogonal (non-overlapping) centroids made by BICA seems to be inappropriate here and might be responsible for the higher error.

In our experiments on the modified data set with more structure and a higher noise level, Figure 8(b), all methods have significantly higher generalization errors, varying between approximately 10% to 21%. The trivial solution of providing each user all those permissions assigned to more than 50% of the users, leads to an error of 23.3%. Again, MAC with 10% generalization error yields significantly lower generalization error than all the other methods. INO, DBPS, and BICA perform almost equally well each with a median error of 20% to 21%. A generalization error of 10% is still very good as this data set contains at least 33% random bits, even though a random bit can take the correct value by chance.

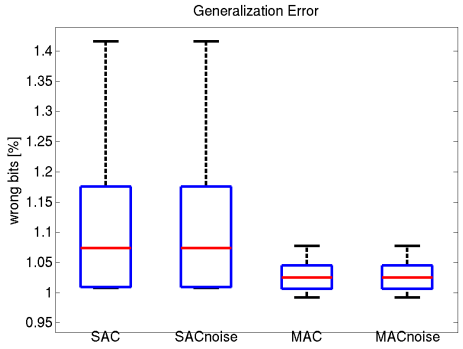
The lower row of Figure 8 shows the average role overlap between the roles obtained by the different methods. This overlap measures the average number of permissions that the inferred roles



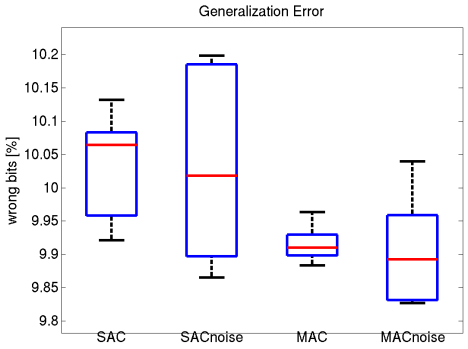
(a) Generalization Error on Original Data



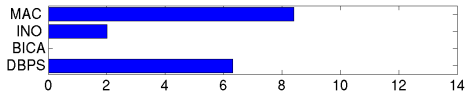
(b) Generalization Error on Modified Data



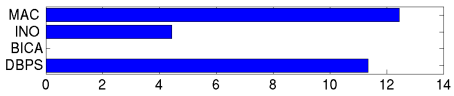
(c) MAC variants on Original Data



(d) MAC variants on Modified Data



(e) Average Role Overlap (%)



(f) Average Role Overlap (%)

Figure 8: Generalization experiment on real data. Graphs (a)-(d) show the generalization error obtained with the inferred roles, and graphs (e)-(f) display the average overlap between roles.

have in common. For BICA, the roles never overlap, by the definition of the method. For all other methods, the increased overlap of the data's structure is reflected in the estimated roles. The decrease in the difference in performance between BICA and the other models after processing the modified data set indicates that the main difficulty for models that can represent overlapping roles is the increased noise level rather than the overlapping structure. We will return to the influence of the data set in our discussion of the results of the MAC model variants in the next section.

5.3.4 RESULTS OF MAC MODEL VARIANTS

To investigate the influence of the various model variants of MAC, we compare the performance reported above for MAC with i) the results obtained by the single-assignment clustering variant (SAC) of the model and ii) with the model variants without a noise part. The middle row of Figure 8 shows the generalization error of SAC and MAC, both with and without a noise model. On the original data set, Figure 8(c), all model variants perform almost equally well. The noise model seems to have little or no impact, whereas the multi-assignments slightly influence the generalization error. Taking MAC's estimated fraction of noisy bits $\hat{\varepsilon} \approx 2.8\%$ into account, we interpret this result by referring to the experiments with synthetic data. There the particular model variant has no influence on the parameter accuracy when the noise level is below 5% (see Figure 5.2.7). As we seem to operate with such low noise levels here, it is not surprising that the model variants do not exhibit a large difference on that data set. On the modified data with more complex structure and with a higher noise level than the original data (Figure 8(d)), the difference between multi-assignments and single-assignments becomes more apparent. Both MAC and SAC benefit from a noise part in the model, but the multi-assignments have a higher influence.

5.3.5 RESULTS ON HP DATA

With all methods described above, we learn RBAC configurations on the publicly available data sets from HP labs (first presented by Ene et al., 2008). The data set 'customer' is the access control matrix of an HP customer. 'americas small' is the configuration of Cisco firewalls that provide users limited access to HP network resources. The data set 'emea' is created in a similar way and 'firewall 1' and 'firewall 2' are created by Ene et al. (2008) by analyzing Checkpoint firewalls. Finally, 'domino' is the access profiles of a Lotus Domino server.

We run the same analysis as on C_{orig}. For the data sets 'customer', 'americas small', and 'firewall 1', we first make a trial run with many roles to identify the maximum cardinality of assignment sets M that MAC uses. We then restrict the hypothesis space of the model accordingly. For 'customer' and 'firewall 1', we use $M = 3$, for 'americas small' we use $M = 2$. For the smaller data sets, we simply offered MAC all possible role configurations, although the model does not populate all of them.

In the cross-validation phase we select the number of roles for each of the methods (except for INO), and the thresholds for BICA and DBPS in the previously described way. Afterwards we compute the generalization error on hold-out test data.

Our experimental findings are summarized in Table 1. We report the favored number of roles, the median generalization error and its average difference to the 25% and 75%-percentiles, and the run-time of one run, respectively. Overall, the MAC variants achieve the lowest generalization error within the variance of this measure. For 'americas small' and 'emea' all methods generalize equally well (note the high variance for 'emea', which is an effect of the small sample size and the high dimensionality of that data set). Here differences between the methods are dominated by run-time and the number of roles that have been found. For 'dominos', INO and BICA are almost as good as MAC, although with a significantly higher variance. Visual inspection of the 'dominos' matrix indicates that this data set has a sparse and simple structure. Differences between the methods are most pronounced on the two 'firewall' data sets. Remarkably, INO finds 80 roles for 'emea', although this data set has only 35 users.

Given the overall good generalization performance of MAC, we conclude that this model is a good 'allrounder'. This also confirms our findings in the experiments with synthetic data. Each of the other methods shows a good performance on individual data sets but not as reliably as MAC. Comparison with the results on synthetic data suggests that their differing performance on different data sets is either due to different fractions of random noise or to true underlying sources with different overlap.

	customer 10,021 users × 277 perms.			americas small 3,477 users × 1,587 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	187	2.40 ± 0.03	49	139	1.03 ± 0.01	80
DBPS	178	2.54 ± 0.05	43	105	1.00 ± 0.03	187
INO	20	7.8 ± 1.6	996	65.6	1.05 ± 0.01	3691
BICA	82	2.66 ± 0.02	200	63	1.00 ± 0.01	64
	firewall1 365 users × 709 perms.			firewall2 325 users × 590 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	49	4.57 ± 0.01	10	10	3.40 ± 0.00	1.8
DBPS	21	13.6 ± 3.1	5	4	19.5 ± 4.4	2
INO	38.2	8.04 ± 0.00	96	6.2	11.15 ± 0.00	14
BICA	18	12.8 ± 3.0	2.1	4	19.9 ± 4.5	0.9
	dominos 79 users × 231 perms.			emea 35 users × 3,046 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	7	1.73 ± 0.00	1.1	3	8.7 ± 1.2	0.7
DBPS	9	2.3 ± 0.5	0.2	8	7.3 ± 2.6	1.1
INO	26	1.7 ± 0.1	9.0	80.4	10.1 ± 2.4	204
BICA	3	1.9 ± 0.3	0.1	5	8.6 ± 2.8	1.0

Table 1: Results on HP labs data for different methods. We report the number of roles, the median run-time of one run, as well as the median generalization error and the half inter-percentile distance between 25% and 75%.

5.4 Complexity and Runtime

The complexity of the optimization problem is determined by the number of objects and features and by the number of possible assignment sets $L := |\mathbb{L}|$. As L can be large for even a small number of clusters, the complexity is dominated by that number. Let the number of clusters that a data item can simultaneously belong to be limited by the *degree* M, i.e. $\max_{\mathcal{L}\in\mathbb{L}} |\mathcal{L}| = M$. Then the size of the assignment set is limited by $L = \sum_{m=0}^{M} \binom{K}{m} \leq 2^K$. Even for moderately sized K and M, this dependence results in computationally demanding optimization problems both for the inference step as well as for assigning new data items to previously obtained clusters. However, if the data at hand truly exhibits such a high complexity (high K and M) then also a single assignment model needs such a high complexity (to prevent the model from underfitting). In this case, a SAC model

must learn L sources, while the MAC variant learns the L possible combinations out of K sources. The number of responsibilities $\gamma_{i\mathcal{L}}$ (Equation 8) to be computed in the E-step is the same for both models. However, in the M-step, MAC shares the source parameters while SAC must estimate them separately. We will shortly elaborate on the relationship between MAC and SAC from the inference perspective. Coming back to the complexity, the high number of responsibilities $\gamma_{i\mathcal{L}}$ to be computed for MAC appears to be a model-order selection issue. One can drastically reduce its complexity by limiting the number of assignment sets as described in Section 5.3.2.

In our experiments on real-world data in Section 5.3, we monitored the run-time, which is depicted in Figure 7(b). Each point represents the runtime for a single run of the different algorithms on an access-control matrix with $N = 2400$ users and $D = 500$ permissions. The number of roles chosen by the respective method is indicated by a vertical line. For INO we report the median number of roles selected. Note that in one run of INO, the model-order selection task is solved 'on-the-fly' while the other methods require multiple runs and an external validation. This overhead is reflected in the runtime. Considerable care is required in interpreting these results since the different methods were implemented by different authors in different languages (Matlab for INO, BICA and MAC, and C++ for DBPS). The DBPS implementation in C++ is impressively fast while the trend of the generalization error over the number of roles is roughly comparable to MAC and BICA. Thus, for large and demanding data sets, one could employ DBPS as a fast 'scout' to obtain an educated guess of the model-order. In conclusion, for all the investigated algorithms the runtime is not a limiting factor in role mining. This computation is only performed once when migrating an access-control system to another one. It is therefore not a problem if the computation takes hours.

5.5 Relationship Between SAC and MAC

In the following, we show that MAC can be interpreted as a SAC model with a parameter sharing rule. In the limit of many observations, MAC is equivalent to SAC with proxy-sources substituting MAC's source combinations. In order to understand the parameter sharing underlying MAC, we write the set of admissible assignment sets $\mathbb{L}$ as a Boolean matrix $\mathbf{z}^{\mathbb{L}} \in \{0,1\}^{L\times K}$. Assuming an arbitrary but fixed numbering of assignment sets in $\mathbb{L}$, $z^{\mathbb{L}}_{lk} = 1$ means that the l^{th} assignment set contains source k, and $z^{\mathbb{L}}_{lk} = 0$ otherwise. Hence, the assignment matrix $\mathbf{z}$ decomposes into $\mathbf{z} = \mathbf{z}^{\mathcal{L}} * \mathbf{z}^{\mathbb{L}}$, where $\mathbf{z}^{\mathcal{L}} \in \{0,1\}^{N\times L}$ denotes the exclusive assignment of objects to assignment sets ($z^{\mathcal{L}}_{il}$ iff object i has assignment set l, and $\sum_l z^{\mathcal{L}}_{il} = 1$ for all i). Using this notation, the decomposition $\mathbf{x} \approx \mathbf{z} * \mathbf{u}$ can be extended to

$$\mathbf{x} \approx \left(\mathbf{z}^{\mathcal{L}} * \mathbf{z}^{\mathbb{L}}\right) * \mathbf{u} = \mathbf{z}^{\mathcal{L}} * \left(\mathbf{z}^{\mathbb{L}} * \mathbf{u}\right) = \mathbf{z}^{\mathcal{L}} * \mathbf{u}^{\text{SAC}} ,$$

where we have defined $\mathbf{u}^{\text{SAC}} := \mathbf{z}^{\mathbb{L}} * \mathbf{u}$ as the proxy-source parameters of the single-assignment clustering model. The same notion of proxy-sources, substituting the disjunction of individual sources, is used in Equation 2 for the probabilistic source parameters. Asymptotically, the two models are equivalent. However, SAC must estimate $L \cdot D$ parameters, while the MAC model only uses $K \cdot D$ parameters. By sharing the parameters of the assignment sets, MAC reduces the number of parameters to be estimated and thereby increases the number of data items available per parameter. Moreover, the sharing rule provides a mutual inconsistency check for the involved parameter estimates. This check is not available if parameters are estimated independently. These two points explain the higher accuracy in the parameter estimators, which we observe in the experiments reported in Section 5.2.

6. Conclusion and Outlook

We have presented a probabilistic method to cluster vectors of Boolean data. In contrast to the conventional approach of mutually exclusive cluster assignments, our method enables a data item to belong to multiple clusters. In our generative model, the clusters are the sources that generate the structure in the data and irregularities are explained by an independent noise process. In a detailed analysis of our model variants, we demonstrate that the proposed method outperforms state-of-the-art techniques with respect to parameter estimation accuracy and generalization ability. In experiments on a real world data set from the domain of role-based access control, our model achieves significantly lower generalization error than state-of-the-art techniques.

Throughout this paper, the Boolean *OR* combines the emissions of multiple sources. However, the proposed concept is neither limited to the Boolean *OR* nor to Boolean data. Further work will address the combination of other kinds of data and other combination rules such as additive combinations of real numbers.

Acknowledgments

This work was partially supported by the Zurich Information Security Center and by the CTI grant Nr. 8539.2;2 EPSS-ES. We thank the reviewers for their valuable comments.

References

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *Int Conf on Management of Data*, 22(2):207–216, 1993.

Eugene L. Allgower and Kurt Georg. Simplicial and continuation methods for approximations, fixed points and solutions to systems of equations. *SIAM Review*, 22:28–85, 1980.

Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, November 1974.

Radim Belohlavek and Vilem Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Comput. Syst. Sci.*, 76(1):3–20, 2010.

Joachim M. Buhmann and Hans Kühnel. Vector quantization with complexity costs. In *IEEE Trans on Information Theory*, volume 39, pages 1133–1145. IEEE, 1993.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 2nd ed.* MIT Press, 2001.

Alina Ene, William Horne, Nikola Milosavljevic, Prasad Rao, Robert Schreiber, and Robert E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *SACMAT '08: Proceeding of the 13th ACM Symposium on Access Control Models and Technologies*, pages 1–10, 2008.

Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1 (2):209–230, 1973.

David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4 (3):224–274, 2001.

Mario Frank, David Basin, and Joachim M. Buhmann. A class of probabilistic models for role engineering. In *CCS '08: Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 299–310, New York, NY, USA, 2008. ACM.

Mario Frank, Joachim M. Buhmann, and David Basin. On the definition of role mining. In *SACMAT '10: Proceeding of the 15th ACM Symposium on Access Control Models and Technologies*, pages 35–44, New York, NY, USA, 2010. ACM.

Mario Frank, Morteza Chehreghani, and Joachim M. Buhmann. The minimum transfer cost principle for model-order selection. In *ECML PKDD '11: Machine Learning and Knowledge Discovery in Databases*, pages 423–438. Springer Berlin / Heidelberg, 2011.

Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis - Mathematical Foundations*. Springer, 1999.

Zoubin Ghahramani, Thomas L. Griffiths, and Peter Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics 8. Oxford University Press*, pages 201–225, 2007.

James F. Gimpel. The minimization of spatially-multiplexed character sets. *Communications of the ACM*, 17(6):315–318, 1974.

Thomas L. Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, New York, NY, USA, 2000. ACM.

Katherine A. Heller and Zoubin Ghahramani. A nonparametric bayesian approach to modeling overlapping clusters. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-2007)*, pages 297–304, 2007.

Tommi S. Jaakkola and Michael I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of Artificial Intelligence Research*, 10(1):291–322, 1999.

Ata Kabán and Ella Bingham. Factorisation and denoising of 0-1 data: A variational approach. *Neurocomputing*, 71(10-12):2291–2308, 2008.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Nat Conf on Artificial Intelligence*, pages 763–770, 2006.

Ales Keprt and Václav Snásel. Binary factor analysis with help of formal concepts. In *Proc. of CLA 2004*, pages 90–101, 2004.

Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. Role mining — revealing business roles for security administration using data mining technology. In *SACMAT'03: Proceeding of the 8th ACM Symp on Access Control Models and Technologies*, pages 179–186, New York, NY, USA, 2003. ACM.

Harold W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008*, pages 29–47. Springer Berlin Heidelberg, 2010.

Pauli Miettinen, Taneli Mielikäinen, Aris Gionis, Gautam Das, and Heikki Mannila. The Discrete Basis Problem. In *Proc. of Principles and Practice of Knowledge Discovery in Databases*, pages 335–346. Springer, 2006.

Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. Mining roles with semantic meanings. In *SACMAT '08: Proceeding of the 13th ACM Symposium on Access Control Models and Technologies*, pages 21–30, 2008.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proc. of the IEEE*, pages 2210–2239, 1998.

Larry J. Stockmeyer. The set basis problem is NP-complete. *Report RC5431, IBM Watson Research*, 1975.

Andreas P. Streich, Mario Frank, David Basin, and Joachim M. Buhmann. Multi-assignment clustering for Boolean data. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 969–976, New York, NY, USA, 2009. ACM.

Jaideep Vaidya, Vijay Atluri, and Qi Guo. The Role Mining Problem: Finding a minimal descriptive set of roles. In *SACMAT '07: Proceeding of the 12th ACM Symposium on Access Control Models and Technologies*, pages 175–184. ACM Press, 2007.

Tomáš Šingliar and Miloš Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.

Frank Wood, Thomas L. Griffiths, and Zoubin Ghahramani. A non-parametric Bayesian method for inferring hidden causes. In *Conference on Uncertainty in Artificial Intelligence*, pages 536–543. AUAI Press, 2006.