A Probabilistic Approach to Hybrid Role Mining

Mario Frank mafrank@inf.ethz.ch
Andreas P. Streich astreich@inf.ethz.ch
David Basin basin@inf.ethz.ch
Joachim M. Buhmann jbuhmann@inf.ethz.ch

Department of Computer Science
ETH Zurich, Switzerland

ABSTRACT

Role mining algorithms address an important access control problem: configuring a role-based access control system. Given a direct assignment of users to permissions, role mining discovers a set of roles together with an assignment of users to roles. The results should closely agree with the direct assignment. Moreover, the roles should be understandable from the business perspective in that they reflect functional roles within the enterprise. This requires hybrid role mining methods that work with both direct assignments and business information from the enterprise.

In this paper, we provide statistical measures to analyze the relevance of different kinds of business information for defining roles. We then present an approach that incorporates relevant business information into a probabilistic model with an associated algorithm for hybrid role mining. Experiments on actual enterprise data show that our algorithm yields roles that both explain the given user-permission assignments and are meaningful from the business perspective.

Categories and Subject Descriptors: K.6 [Management of Computing and Information Systems]: Security and Protection

General Terms: Security, Management, Algorithms

Keywords: RBAC, Role Mining, Hybrid Role Mining, Machine Learning, Business Meaning

1. INTRODUCTION

Role-Based Access Control (RBAC) [10] is an access control model used in many systems. In RBAC, rather than assigning permissions directly to users, one introduces a set of roles and defines two relations: a user-role relation that assigns users to roles and a role-permission relation that assigns roles to permissions. This decomposition facilitates the administration of authorization policies since roles are (or should be) natural abstractions of functional roles within an enterprise and the two relations are conceptually easier to work with than a direct assignment of users to permissions.

Despite the many advantages of RBAC, it is surprisingly difficult to configure and maintain an RBAC system for large enterprises. This task, called *role engineering* [6], can be approached from two different directions, top-down or bottom-up, each with its own strengths and weaknesses.

Top-down role engineering [9, 15] starts by analyzing an enterprise's business structure. This structure includes business information such as the organizational hierarchy, employees' job descriptions, or their workplace. This information is used to determine the permissions that users should have and to bundle these permissions into roles. The resulting roles are easy to understand from the business perspective as they are derived from business concepts. However, the business information alone is unlikely to contain enough information to derive an RBAC configuration that closely corresponds to the existing direct assignment of users to permissions, i.e., the authorizations may change considerably.

In contrast, bottom-up approaches start with the direct assignment of users to permissions available, for instance, as access-control lists. One then analyzes these assignments for patterns, attempting to capture the underlying structure with roles and an assignment of users to roles. This analysis can be automated using data mining algorithms and is called *role mining*. Interestingly, top-down approaches complement bottom-up methods in terms of their strengths and weaknesses. Role mining algorithms often achieve a good fit with the existing user-permission assignments but they discover roles that are difficult to interpret from a business perspective and that are cumbersome for administrators to work with, e.g., to maintain the RBAC configuration as the enterprise evolves.

In this paper, we propose an approach for *hybrid role mining* that incorporates top-down business information into a bottom-up role-mining process and thereby combines the strengths of both approaches. Our method has two parts, with associated measures, models, and algorithms.

1. Identify business information and determine its relevance for roles with respect to the existing access-control data.

2. Incorporate the relevant data into the role mining process itself.

In the first part, we begin by identifying business information that could be relevant for roles. Since roles should represent functions within an enterprise, the enterprise's Human Relations department is likely to be a good source of

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

such data, e.g., employees' positions, working groups, locations, etc. But not all such data is equally relevant. Indeed, there is unlikely to be any business information that precisely captures the roles of all system users. For example, employees with the same position (group or location) usually differ in some of their permissions. Moreover, simply using all available data is not the solution: It not only increases the computational cost of hybrid role mining, it can actually lead to worse results, as we will see later (Section 4). Hence it is necessary to select the most relevant business information for use in hybrid role mining. To support this selection, we define an appropriate entropy-based notion of *relevance* and show how to compute it.

In the second part, we present a method to incorporate business information into a role mining algorithm based on a probabilistic model of an RBAC system. Our probabilistic model encodes how *likely* it is that a particular role decomposition underlies a given user-permission assignment. We combine this model with an objective function on business information that optimizes business relevance: users with the same business attribute should ideally be assigned to the same set of roles.

Our contribution is as follows. First, we develop a method for quantitatively analyzing any kind of business information, establishing a formal notion of relevance that can be used with any hybrid role mining algorithm. This is the first such method proposed for tackling the data selection problem in hybrid role mining. Second, we present a hybrid role mining method based on the combination of a sound statistical model with an objective function that accounts for business information. Through experiments, we demonstrate that our hybrid role mining method finds roles that both generalize well and correspond to the business information at hand. Finally, we provide two quantitative measures to objectively assess the results of role mining methods: a generalization test and an entropy-based measure of the business relevance of an RBAC system. These measures are general in that they can be used to assess the results returned by any role-mining approach.

The remainder of the paper is organized as follows. In Section 2, we examine related work in role mining. Afterwards, in Section 3, we introduce background concepts before explaining our relevance measure in Section 4. In Section 5, we develop our model for hybrid role mining and the corresponding algorithm. We report on experimental results obtained with data from a real-world enterprise in Section 6. Finally, we draw conclusions in Section 7.

2. RELATED WORK

The term role mining was coined by [13] in 2003. Since then, a number of different bottom-up approaches have been presented, e.g., [3, 11, 19, 20, 22, 23]. In [21], the role mining problem and some of its variants were defined. Almost all previous approaches are combinatorial, based on different ways of searching through possible roles. Exceptions are [11, 20], which proposed a probabilistic model for RBAC where bottom-up role mining is recast as the problem of finding the most likely roles underlying a given user-permission assignment relation. This approach has the feature that role mining can be used not only to discover roles, but also to detect exceptional assignments (e.g., assignments needed for a "special operation") or even wrong assignments. Following this idea, our approach is also based on a probabilistic model, and, as in [20], we explicitly model the processes leading to exceptional assignments and wrong assignments. More importantly, we provide a way to combine our model with top-down business information.

Several approaches have been previously proposed for top-down role engineering, all of which are manual. [17] proposed a method to derive roles by analyzing business processes and carried out a case-study on one enterprise as a proof of principle. Similarly, [15] presented an approach based on analyzing business scenarios to find appropriate user-role and role-permission assignments. Both approaches are time consuming for large companies as they require humans to reason about the business processes or scenarios in the enterprise. Organizational theory is used to define criteria for creating roles in [7]. The criteria are based on a user's position in the enterprise hierarchy, his job function, and the resources he requires for his work. As we demonstrate in this paper, these types of business information can also be used for automated role engineering.

To the best of our knowledge, there are only two other approaches to hybrid role mining. In [14], a candidate set of roles is created using an algorithm from formal concept analysis [12]. [4] proposes a hybrid approach that extends the bottom-up algorithm proposed in [3]. It is based on an algorithm from [1] for association rule mining. Both of these approaches work by creating a candidate set of roles based solely on bottom-up data and afterwards using the business information in a post-processing step to select roles in a greedy fashion. In contrast, our algorithm uses business information during the role-creation step. We thereby explore more of the solution space and find solutions that cannot be reached using a preprocessed set of candidate roles. Another difference is that, in contrast to our approach, [14, 4] both lack a probabilistic model whose parameters are to be optimized. The outcome is thus determined by design decisions made for the combinatorial post-processing steps. Moreover, neither approach provides a way to measure the relevance of business information for role mining.

3. PRELIMINARIES

Following the RBAC standard [10], we will work with the following sets in this paper:

- $USERS$, the set of users (or employees),
- $PERMS$, the set of permissions (or privileges),
- $ROLES$, the set of roles,
- $UA \subseteq USERS \times ROLES$, a user-role assignment relation,
- $PA \subseteq ROLES \times PERMS$, a role-permission assignment relation, and
- $UPA \subseteq USERS \times PERMS$, a user-permission assignment relation.

Let $N := |USERS|$, $D := |PERMS|$, and $K := |ROLES|$. We will assume that the sets of users, permissions, and roles all can be ordered, e.g., we will speak of the i^{th} user, for $i \in \{1, \ldots, N\}$. The three relations above are all many-to-many. For notational convenience, we encode each of them as a binary matrix: We represent UPA as $\mathbf{x} \in \{0,1\}^{N \times D}$,

UA as $\mathbf{z} \in \{0,1\}^{N \times K}$, and PA as $\mathbf{u} \in \{0,1\}^{K \times D}$. In this representation, $x_{ij} = 1$ ($x_{ij} = 0$) indicates that user i is (not) assigned permission j. The representation is analogous for z_{ik} (user i and role k) and u_{kj} (role k and permission j).

The relation PA induces a collection of K sets of permissions, namely those sets of permissions assigned to the same role k, for each $k \in \{1, \ldots, K\}$. As is often done in practice, we will identify these sets with the set of roles $ROLES$. The indices i, j, and k will be consistently range over $USERS$, $PERMS$, and $ROLES$ respectively, throughout the paper and will often be used synonymously with the objects that they index.

In bottom-up role mining, the user-permission assignment matrix $\mathbf{x}$ is given and must be approximated with a user-role relation $\mathbf{z}$ and a role-permission relation $\mathbf{u}$. In the approximation, the number of differing assignments $\|\mathbf{x} - \mathbf{z} \otimes \mathbf{u}\|$ should be small, where $\mathbf{a} \otimes \mathbf{b}$ denotes the Boolean matrix product, defined such that $c_{ij} = \bigvee_k [a_{ik} \wedge b_{kj}]$. Sometimes we will refer to the pair $(\mathbf{z}, \mathbf{u})$ as the *role decomposition* of the direct user-permission assignment $\mathbf{x}$.

Enterprises maintain different types of business information about each user. Examples include a user's working address, job code, organization unit, etc. We encode predicates specifying which users i have the business-information attribute s (for example, which users work in the accounting department) as a family of Boolean variables w_{is}. The variable w_{is} has the value of 1 if the user i has attribute s, and 0 otherwise. We shall assume that for each type of business information, each user has a single attribute s, e.g., a user is member of exactly one department.

Finally, we use the notation $z_{i\cdot}$ to represent the i^{th} row of the assignment-matrix $\mathbf{z}$, which encodes the roles possessed by the user i (note that a user may have more than one role). The same notation will be used for other matrices as well.

4. ENTROPY-BASED RELEVANCE MEASURES

An abundance of information is usually available in digital form within an enterprise, but most of it is ill-suited for hybrid role mining. To be useful, the data must provide information about the relationship between employees and the permissions they have been granted. In this section, we provide a measure that quantifies to what extent a given type of business information agrees with the direct user-permission assignment. When the agreement is high, we say that the data is relevant because it increases the information about whether a user has a particular permission.

Business information with too little relevance can actually lead to worse role mining results. This deterioration occurs when the objective of agreement between roles and business information conflicts with the objective of finding roles that best explain the direct user-permission assignment. This conflict can be avoided by carefully pre-selecting the business information. The relevance measure we provide can be used for such a pre-selection.

For the definition of relevance, we first introduce the following quantities. The random variable $X_j \in \{0,1\}$ denotes the assignment of permission j to a generic user. S is the random variable that corresponds to the business attribute of a generic user (e.g. "job code") and let s be one of the actual values that S can take (e.g. "accountant"). Let $p(x_j) := 1/N \cdot \sum_i x_{ij}$ be the empirical probability of j being assigned to an unspecified user, and let $p(x_j|S = s) := 1/N \cdot \sum_i x_{ij} w_{is}$ be the empirical probability of j being assigned to a user with business attribute s. The natural measure for the information of a random variable A is its entropy $H(A)$ [5], which in the case of a permission j is the binary entropy $h(X_j)$. The binary entropy, defined as

$$h(X_j) := -\sum_{x_j \in \{0,1\}} p(x_j) \log_2 \left(p(x_j)\right), \tag{1}$$

quantifies the missing information on whether the permission j is granted to some user. The conditional entropy

$$h(X_j|S) := -\sum_{s \in S} p(s) \sum_{x_j \in \{0,1\}} p(x_j|S = s) \log_2 \left(p(x_j|S = s)\right) \tag{2}$$

encodes how much of the missing information $h(X_j)$ of X_j remains if S is known. The mutual information

$$I(X_j; S) := h(X_j) - h(X_j|S) \tag{3}$$

measures how much the knowledge of S increases the information on X_j. We therefore propose the mutual information $I(X_j; S)$ to measure how much the knowledge of the business information S helps us to predict the assignment x_j of permission j to a generic user. In order to express this absolute reduction of missing information in a relative way, we define the measure of relevance $\rho_j(S)$ of business information S for permission j to be the relative mutual information ([5], p. 45)

$$\rho_j(S) := \frac{I(X_j; S)}{h(X_j)} = 1 - \frac{h(X_j|S)}{h(X_j)}. \tag{4}$$

This number can be interpreted as the fraction of all bits in X_j that are shared with S. Alternatively, $\rho_j(S)$ can be read as the fraction missing information on permission j that is removed by the knowledge of S.

For each kind of business information S that appears potentially useful for role mining, one can now compute $\rho_j(S)$ for all permissions j and examine their distribution (e.g. Fig. 2). The larger the overall decrease in entropy of the permissions under the knowledge of the business information S, the better qualified S is as a candidate for hybrid role mining. Given different types of business information that are expected to be helpful for role mining, one can compare them and their combinations according to the proposed measure and pick the most relevant one.

In principle, this relevance analysis can be carried out using any kind of (digitally) available business information. We will give examples in Section 6.1.

5. HYBRID ROLE MINING

Our goal is to infer user-role and role-permission assignments based on a direct user-permission assignment matrix and additional business information. The basic assumption of role mining is the existence of a role structure underlying the direct assignment. It is this structure that should be discovered by a role mining algorithm. Our method searches for the role decomposition that is most likely to explain the direct user-permission assignment.

In Section 5.1, we explain the probabilistic model [20] underlying our computations of the probabilities of different role decompositions. Afterwards, we show how to combine business information with this model in Section 5.2 and we

present an optimization strategy for inferring the parameters of the combined model in Section 5.3.

5.1 The Likelihood of a Role Decomposition

Our model has two parts: a structured and an unstructured part. The structured part $p_S(\cdot)$ represents the role structure, i.e., the entries of the user-permission matrix $\mathbf{x}$ that can be explained by the matrix decomposition $\mathbf{x} = \mathbf{z} \otimes \mathbf{u}$. Computing this decomposition with minimal error was shown to be NP-hard in [21]. We solve a polynomial variant of the problem by optimizing the (soft) probabilities $\beta_{kj} := P(u_{kj} = 0)$ that the role k does not contain permission j, instead of the (hard) assignments u_{kj}. β denotes the full $K \times D$-matrix of these probabilities. Upon convergence of the optimization algorithm, the β_{kj} are very close to either 0 or 1. Formally, the probability that user i who is assigned to roles $z_{i\cdot}$ has permission j under the structured part is

$$p_S\left(x_{ij} \mid z_{i\cdot}, \beta\right) = \left(1 - \prod_{k=1}^{K} \beta_{kj}^{z_{ik}}\right)^{x_{ij}} \left(\prod_{k=1}^{K} \beta_{kj}^{z_{ik}}\right)^{1-x_{ij}} , \quad (5)$$

where z_{ik} is the hard (non-probabilistic) binary assignment of user i to role k. As explained above, the goal of role mining is to determine the parameters β and $\mathbf{z}$. Since a users' roles are combined using Boolean disjunction, the probability of not having a permission decreases as the number of roles that are likely to contain this permission increases.

The unstructured part $p_U(\cdot)$ represents all elements of the user-permission matrix that cannot be explained with the detected role structure. It comprises namely the permissions a user gets exceptionally (e.g., for "special tasks") and the mistakes made when originally specifying the user-permission matrix. As we assume exceptional assignments as well as errors to be uniformly distributed over all users and permissions, we model the probability $p\left(x_{ij} \mid r\right)$ for the user i to have the permission j by a global Bernoulli process with parameter r:

$$p_U\left(x_{ij} \mid r\right) = r^{x_{ij}} \left(1 - r\right)^{1-x_{ij}} . \quad (6)$$

This process allows a user to have a permission without getting it from the structured part p_S. Technically, this part of the model allows one to explain the predominant structure of the data without letting exceptional or erroneous permission assignments influence the roles. Exceptions can then be automatically reported and manually checked for errors.

Let ϵ be the probability that an assignment x_{ij} is generated by the unstructured part of the model p_U. Then the full model p_M gives the likelihood

$$p_M\left(x_{ij} | z_{i\cdot}, \beta, r, \epsilon\right) = \epsilon\, p_U(x_{ij}|r) + (1-\epsilon)\, p_S(x_{ij}|z_{i\cdot}, \beta) . \quad (7)$$

For a more convenient notation, we introduce the notion of a *role set* $\mathcal{L}_i$, which contains the indices of all roles that the user i belongs to, i.e. $\mathcal{L}_i := \{k \in \{1, \ldots, K\} \mid z_{ik} = 1\}$. $\mathcal{L}_i$ is an alternative representation of the row $z_{i\cdot}$. The set of all possible role sets is denoted by $\mathbb{L}$. With this notation, the likelihood (7) becomes

$$p_M\left(x_{ij} | \mathcal{L}_i, \beta, r, \epsilon\right) = \epsilon p_U(x_{ij}|r) + (1-\epsilon) p_S(x_{ij}|\mathcal{L}_i, \beta) . \quad (8)$$

Assuming that different elements of the user-permission matrix $\mathbf{x}$ are independent of each other given the parameters β and $\mathbf{z}$, the total probability is given as

$$p_M(\mathbf{x}|\mathbf{z}, \beta, r, \epsilon) = \prod_{i,j} p_M\left(x_{ij} \mid \mathcal{L}_i, \beta, r, \epsilon\right) . \quad (9)$$

The unknown model parameters that must be inferred are thus the user-role assignment $\mathbf{z}$, the roles expressed in terms of their probabilities β of not containing particular permissions, the global noise probability ϵ, and the probability r of exceptionally getting a permission. These parameters will be chosen to maximize the likelihood of the observed data. While direct maximization of (9) is computationally demanding, its logarithm is easier to handle (when taking derivatives) and attains the maximum at the same parameter values. We therefore define the *bottom-up costs* of assigning a given user i to a set of roles $\mathcal{L}$ as the negative logarithm of the likelihood function (8):

$$\begin{aligned} R^{(ll)}_{i,\mathcal{L}} &= -\log\left(\prod_j p_M\left(x_{ij} \mid \mathcal{L}, \beta, r, \epsilon\right)\right) \\ &= -\sum_j \log\left(\epsilon \cdot p_U(x_{ij}) + (1-\epsilon) \cdot p_S(x_{ij}|\mathcal{L}, \beta)\right) . \quad (10) \end{aligned}$$

The costs $R^{(ll)}$ for all users are then

$$R^{(ll)} = \sum_{i,\mathcal{L}} z_{i\mathcal{L}} R^{(ll)}_{i,\mathcal{L}} , \quad (11)$$

with $z_{i\mathcal{L}} \in \{0, 1\}$ indicating the assignment of i to the set of roles $\mathcal{L}$.

5.2 Incorporating Business Information

Business Information and Likelihood

Optimizing the model parameters with respect to the log-likelihood (11) seeks to find roles and user-role assignments that best explain the given direct-assignment data. Since many different user-role and role-permission assignments can equip the users with their permissions, there are many role configurations with very similar likelihood. Technically speaking, the solution space for a solution with maximum likelihood has many local optima with similar values for the objective function. However, many of these local optima represent RBAC configurations that are unintuitive from a business perspective. A hybrid role mining algorithm that combines the likelihood with business information will find solutions that are more meaningful.

More formally, incorporating business information leads to an optimization problem with two objectives.

1. The role decomposition $(\mathbf{z}, \mathbf{u})$ should accurately approximate a user-permission matrix, both for the current users and for new users.

2. The role assignments should agree with the business information.

These two objectives are weighted and combined to a unified objective function. The weighting allows us to choose the influence of each of the two sub-objectives. Note that if these sub-objectives conflict, the solutions of the joint objective will not be a solution of the single objectives. However, as we will show later (e.g. see Figure 3), the business meaning of a role decomposition can be substantially increased without significantly increasing the bottom-up costs (11).

As mentioned above, this behavior is caused by the fact that many configurations exist with similarly low bottom-up costs (i.e. similarly high likelihood) but differing degrees of business interpretability.

In the following, we will introduce a cost function $R^{(S)}$ for business information S, reflecting the above assumption. We then define a unified objective function as a linear combination of the business information costs and the log-likelihood costs (11):

$$R = R^{(ll)}/D + \lambda R^{(S)}\,, \tag{12}$$

where $\lambda \geq 0$ is the mixing parameter, weighting the influence of the business information. A weighted linear combination is the easiest way to merge the two cost functions into a single one, and allows a smooth transition from a scenario without business information ($\lambda = 0$) to one that is completely determined by the business information ($\lambda \to \infty$). The term $1/D$ makes the log-likelihood costs independent of the number of permissions D. This makes it easier to compare with the permission-independent term $R^{(S)}$, which we subsequently give in (13), for arbitrary sized systems.

Objective Function for Business Information

Setting up requirements for a role decomposition from the business information perspective is probably the most crucial step in designing a hybrid role-mining technique. Our goal is to make the role decomposition as meaningful as possible from the business perspective. This perspective is represented by the business information at hand which could denote, for instance, organizational units or contract types.

Our assumption about the relationship between business information and permissions is as follows: *The business information abstractly describes what users should be able to do.* This assumption implies that two users with the same business attributes will have essentially the same tasks within the company. This assumption, together with the principle of least privileges, which states that users should only have the permissions required for their tasks, therefore implies that users with the same business attributes should have similar permissions. Furthermore, note that only the entire set of roles assigned to a user determines his permissions. Hence, to evaluate if two users of the same business attribute have similar permissions, one must compute a measure of similarity based on their *full* role sets.

Summing up the above considerations, we assume that a role decomposition is meaningful if employees satisfying identical business predicates (i.e., having the same business information attributes) are also assigned to a similar (ideally the same) set of roles.

Note that this design decision is different from requiring that all users with identical roles have similar business attributes, as proposed in [4]. We advocate our approach for two reasons: First, it favors solutions where knowledge of the business attributes determines the roles, while the other approach leads to solutions where the roles determine the business attributes. In practice, one usually *seeks* the assignment of roles and *knows* the business information. Second, most enterprises have some permissions that are granted to almost all users, such as reading email. Our objective avoids an unnecessarily high number of roles by allowing roles capturing such permissions to be shared among users with different business attributes (e.g. across organizational units).

Given the above considerations, we propose an objective function that compares all pairs of users (i, i') having the business attribute s with respect to their role assignments $(z_{i\cdot},\, z_{i'\cdot})$. Using the Boolean variable $w_{is} \in \{0, 1\}$ to encode whether user i has business attribute s ($w_{is} = 1$) or not ($w_{is} = 0$), the total costs of a role assignment $\mathbf{z}$ are given as

$$R^{(S)} \;=\; \frac{1}{N} \sum_{s} \sum_{i,i'} w_{is} w_{i's} \sum_{k} z_{i'k} \left(1 - 2 z_{i'k} z_{ik}\right). \tag{13}$$

N is the total number of users and $k \in \{1, .., K\}$ is the role index. Each user has a single business attribute s, i.e. $\sum_s w_{is} = 1$, but can be assigned to multiple roles, $1 \leq \sum_k z_{ik} \leq K$. The term $\sum_k z_{i'k}\left(1 - 2 z_{i'k} z_{ik}\right)$ in (13) computes the agreement between the binary assignment vectors $(z_{i\cdot}, z_{i'\cdot})$ for all pairs of users (i, i') having the same attribute s (which is the case iff $w_{is} w_{i's} = 1$). The subterm $(1 - 2 z_{i'k} z_{ik})$ switches the sign of a single term such that agreements ($z_{ik} z_{i'k} = 1$) are rewarded and differences ($z_{ik} z_{i'k} = 0$) are penalized.[1]

For notational convenience, let $N_{sk} := \sum_i z_{ik} w_{is}$ be the number of users that have the business attribute s and are assigned to role k, and let s_i be the attribute of user i. With these auxiliary variables, we simplify the above expression as follows.

$$\begin{aligned} R^{(S)} &= \frac{1}{N} \sum_{s,i} w_{is} \sum_{k} \left(N_{sk} - 2 z_{ik} N_{sk}\right) \\ &= \frac{1}{N} \sum_{i} \sum_{k} \left(N_{s_i k} - 2 z_{ik} N_{s_i k}\right) \\ &= \sum_{i,k} \left(1 - z_{ik}\right) \frac{N_{s_i k}}{N} - \sum_{i,k} z_{ik} \frac{N_{s_i k}}{N} \end{aligned} \tag{14}$$

This formulation of the costs is more intuitive: a user i has a business attribute s_i and $N_{s_i k}$ is the number of users having the same attribute that are assigned to role k. User i should be assigned to k if $N_{s_i k}$ is high. The first term in (14) penalizes role decompositions *not* assigning i to such roles ($z_{ik} = 0$). The second term rewards solutions with such assignments ($z_{ik} = 1$).

We would like to directly compare this function with the costs $R^{(ll)}_{i,\mathcal{L}}$ of assigning a given user i to a set of roles $\mathcal{L}$. We therefore restate the above expression by substituting z_{ik} by the assignments $z_{i\mathcal{L}}$ from user i to the set of roles $\mathcal{L}$ and the assignments $z_{\mathcal{L}k}$ from role sets to roles. Then, $z_{ik} = \sum_{\mathcal{L}} z_{i\mathcal{L}} z_{\mathcal{L}k}$, and therefore

$$\begin{aligned} R^{(S)} &= \sum_{i,k} \left(\left(1 - \sum_{\mathcal{L}} z_{i\mathcal{L}} z_{\mathcal{L}k}\right) \frac{N_{s_i k}}{N} - \sum_{\mathcal{L}} z_{i\mathcal{L}} z_{\mathcal{L}k} \frac{N_{s_i k}}{N} \right) \\ &= \sum_{i,\mathcal{L}} z_{i\mathcal{L}} \left(\sum_{k} \frac{N_{s_i k}}{N} - \sum_{k} z_{\mathcal{L}k} \frac{N_{s_i k}}{N} - \sum_{k} z_{\mathcal{L}k} \frac{N_{s_i k}}{N} \right) \\ &= \sum_{i,\mathcal{L}} z_{i\mathcal{L}} \left(\sum_{k \notin \mathcal{L}} \frac{N_{s_i k}}{N} - \sum_{k \in \mathcal{L}} \frac{N_{s_i k}}{N} \right) \\ &= \sum_{i,\mathcal{L}} z_{i\mathcal{L}} R^{(S)}_{i,\mathcal{L}} \quad . \end{aligned} \tag{15}$$

[1] An alternative to (13) would be to compute the Hamming distance between the two assignment vectors. However, this has the drawback of penalizing pairs with differently sized role sets.

In the second line we made use of the fact that a user is only assigned to a single *set* of roles $\mathcal{L}$.

Given the top-down objective function in this form, we can directly compare it with the log-likelihood costs given by (10).

5.3 Inference Algorithm

We use Deterministic Annealing (DA) [2, 18], an iterative gradient-descent optimization method, to infer the model parameters. In the following, we explain how we compute the objective function derived above in such an iterative setting. Afterwards, we briefly describe this iterative optimization scheme and explain how, in each DA-step, we update the parameters to be optimized in our particular problem.

Computation of $R_{i,\mathcal{L}}^{(S)}$.

Given an iterative optimization scheme for minimizing an objective function $R^{(S)}$, one faces a computational problem with the above quantities: compute the optimal assignments $z_{i\mathcal{L}}$ from the N_{s_ik}, which are, in turn, computed from the $z_{i\mathcal{L}}$ themselves. To make this computation at step t of our algorithm feasible, we use the expected assignments $\gamma_{i\mathcal{L}}^{(t-1)} := \mathbb{E}\left[z_{i\mathcal{L}}^{(t-1)}\right]$ of the previous step instead of the Boolean $z_{i\mathcal{L}}^{(t)}$ to approximate $N_{s_ik}^{(t)}$ by its expectation:

$$N_{s_ik}^{(t)} \approx \mathbb{E}\left[N_{s_ik}^{(t-1)}\right] = \sum_{\mathcal{L}} z_{\mathcal{L}k} \sum_{i'} w_{i's_i} \gamma_{i'\mathcal{L}}^{(t-1)} . \tag{16}$$

This so-called *mean-field approximation* [2] makes the computation of $R_{i,\mathcal{L}}^{(S)(t)}$ feasible. Therewith, the costs of a user belonging to a set of roles are

$$R_{i,\mathcal{L}}^{(S)(t)} \approx \sum_{k \notin \mathcal{L}} \frac{\mathbb{E}\left[N_{s_ik}^{(t-1)}\right]}{N} - \sum_{k \in \mathcal{L}} \frac{\mathbb{E}\left[N_{s_ik}^{(t-1)}\right]}{N} \quad . \tag{17}$$

Deterministic Annealing.

Deterministic Annealing is a gradient-descent algorithm for optimizing an objective function. At each step t of the algorithm, it enables a smoothly varying trade-off between the cost function to be optimized and the uniform distribution controlled by the Lagrange parameter T. The cost function $R(\cdot)$ of a problem determines the Gibbs distribution $p(\cdot) = 1/Z \exp(-R(\cdot)/T)$, where $Z = \sum_{\{\cdot\}} \exp(-R(\cdot)/T)$ is the normalizing constant and the sum is over all points in the solution space. Minimizing the Lagrangian $F = -T\log(Z) = \mathbb{E}[R] - TH$ at a given T is equivalent to maximizing the entropy H (seeking a solution close to the uniform distribution) while minimizing the expected costs $\mathbb{E}[R]$ (seeking a minimum cost solution). For historical reasons, T is often called the (computational) temperature. Starting the optimization at a high T and successively decreasing it, smooths the costs landscape in the beginning and helps the gradient-based optimization procedure to avoid getting trapped in local minima.

We choose an initial temperature and a constant rate cooling scheme ($T^{(t)} = \alpha \cdot T^{(t-1)}$, with $\alpha < 1$) as described in [18]. At each value of the temperature, we run one step of the Expectation-Maximization (EM) algorithm [8]. First, the expected value of the data likelihood is computed, given the current set of parameters. Second, the parameters are chosen such that this quantity is maximized. Finally, the temperature is decreased and the solution of the previous step is used to estimate the likelihood.

Parameter Estimation.

We now give the concrete expressions for our setting. To simplify notation, we define the non-normalized responsibilities of the role set $\mathcal{L}$ for the data item i as

$$c_{i,\mathcal{L}} := \exp\left(-R_{i,\mathcal{L}}/T\right) . \tag{18}$$

The normalized responsibilities $\gamma_{i,\mathcal{L}}$ (the expectation of $z_{i\mathcal{L}}$ according to the Gibbs distribution) and the Lagrangian F are defined as follows:

$$\gamma_{i,\mathcal{L}} := \frac{c_{i,\mathcal{L}}}{\sum_{\mathcal{L}'} c_{i,\mathcal{L}'}} \tag{19}$$

$$F := -T \sum_i \log\left(\sum_{\mathcal{L}} c_{i,\mathcal{L}}\right) \tag{20}$$

In the expectation step, the costs of assigning user i to the set of roles $\mathcal{L}$ is computed for all users and role sets according to (10) and (17) using the estimated parameters from the previous maximization step (computed at a higher temperature). The normalized responsibilities are then computed using Equations (18) and (19).

In the maximization step, the model parameters β, ϵ, and r are estimated such that they minimize the Lagrangian F, i.e. they are updated to the values where the partial derivative of F is zero. For the probabilistic parameters β, we get

$$\frac{\partial F}{\partial \beta_{pq}} = (1-\epsilon) \sum_i \sum_{\{\mathcal{L} \in \mathbb{L} | p \in \mathcal{L}\}} \left(f_{\mathcal{L},q,1}^{x_{iq}} \cdot f_{\mathcal{L},q,0}^{1-x_{iq}} \cdot \gamma_{i,\mathcal{L}} \cdot \beta_{\mathcal{L}\setminus\{p\},q} \right) , \tag{21}$$

$$\text{with} \quad \beta_{\mathcal{L}\setminus\{p\},j} := \prod_{k \in \mathcal{L}, k \neq p} \beta_{k,j}$$

$$f_{\mathcal{L},q,1} := \frac{-1}{\epsilon r + (1-\epsilon)(1-\beta_{\mathcal{L},q})}$$

$$f_{\mathcal{L},q,0} := \frac{1}{\epsilon(1-r) + (1-\epsilon)\beta_{\mathcal{L},q}}.$$

Deriving F with the weight of the unstructured part ϵ gives

$$\frac{\partial F}{\partial \epsilon} = -\sum_i \sum_{\mathcal{L}} \gamma_{i,\mathcal{L}} \sum_j \left(g_{\mathcal{L},j,1}^{x_{ij}} \cdot g_{\mathcal{L},j,0}^{1-x_{ij}} \right) , \tag{22}$$

$$\text{with} \quad g_{\mathcal{L},j,1} := \frac{r - (1-\beta_{\mathcal{L},j})}{\epsilon r + (1-\epsilon)(1-\beta_{\mathcal{L},j})}$$

$$g_{\mathcal{L},j,0} := \frac{(1-r) - \beta_{\mathcal{L},j}}{\epsilon(1-r) + (1-\epsilon)\beta_{\mathcal{L},j}}.$$

The optimal parameter r of the unstructured part is found with the derivative

$$\frac{\partial F}{\partial r} = \sum_{i,j} \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} \cdot h_{\mathcal{L},j,1}^{x_{ij}} \cdot h_{\mathcal{L},j,0}^{1-x_{ij}} , \tag{23}$$

$$\text{with } h_{\mathcal{L},j,1} := \frac{-1}{\epsilon r + (1-\epsilon)(1-\beta_{\mathcal{L}j})}$$

$$h_{\mathcal{L},j,0} := \frac{1}{\epsilon(1-r) + (1-\epsilon)\beta_{\mathcal{L}j}}.$$

When optimizing one of the model parameters, all other model parameters are kept fixed. Since the update equations (21)–(23) are not analytically solvable, the roots are

determined using bisection search. The optimization algorithm terminates if for each user i the $\gamma_{i,\mathcal{L}}$ have converged to 1 for a single set of roles $\mathcal{L}$. The probabilistic roles β are then rounded to get the estimated matrix $\hat{\mathbf{u}}$ representing the estimated role-permission assignment relation. Note that, upon convergence, the entries of β are usually already very close to 0 or 1.

Starting from different random initializations, we run the algorithm multiple times (e.g., 20) and pick the best solution according to the objective function (12). We give in Figure 3 the log-likelihood costs and the business information costs of the best solution in our case-study for different values of λ. A description of the algorithm in pseudo-code is given in Algorithm 1.

input : user permission matrix $\mathbf{x}$,
business information S,
parameters λ, T_0, α
output: role assignment matrices $\mathbf{z}$,
probabilistic role prototypes β

1: Randomly initialize β, ϵ, and r
2: $T = T_0$
3: **while** not converged **do**
4: compute $R_{i,\mathcal{L}}$ according to (10) & (17)
5: $c_{i,\mathcal{L}} \leftarrow \exp\left(-R_{i,\mathcal{L}}/T\right)$
6: $\gamma_{i,\mathcal{L}} \leftarrow \frac{c_{i,\mathcal{L}}}{\sum_{\mathcal{L}'} c_{i,\mathcal{L}'}}$
7: Solve (21)–(23) for β, ϵ, and r, respectively
8: $T \leftarrow \alpha \cdot T$
9: **end while**

Algorithm 1: Probabilistic Hybrid Role Mining

6. EXPERIMENTAL RESULTS

In this section, we report on experimental results on a dataset from an actual enterprise. The dataset contains assignments between 22,352 users and 1,786 permissions. Furthermore we had access to two kinds of business information provided by the company's Human Resources department: each user's organizational unit (OU) and job-code (JC). The organizational unit groups the users based on their division and section within the enterprise. The job-code is a number identifying the kind of employment contract that the employee has. For example, an employee may be in the division "customer service, overseas" and have job-code 4, indicating that her contract is of the type "head of division". Each employee has a single job-code and a single organizational unit. The enterprise considered has 6,630 OUs and 1,030 JCs.

In the following, we will determine the relevance of these two kinds of business information for role mining using the measures introduced in Section 4. Afterwards, we report on experiments that illustrate some of the advantages and drawbacks of hybrid role-mining.

6.1 Top-Down Information Analysis

As described in Section 5.2, we assume that a user's organizational unit and job-code provide information about his duties and thus his required system permissions. We now test this assumption and measure the information gain using the analytic methods described in Section 4.

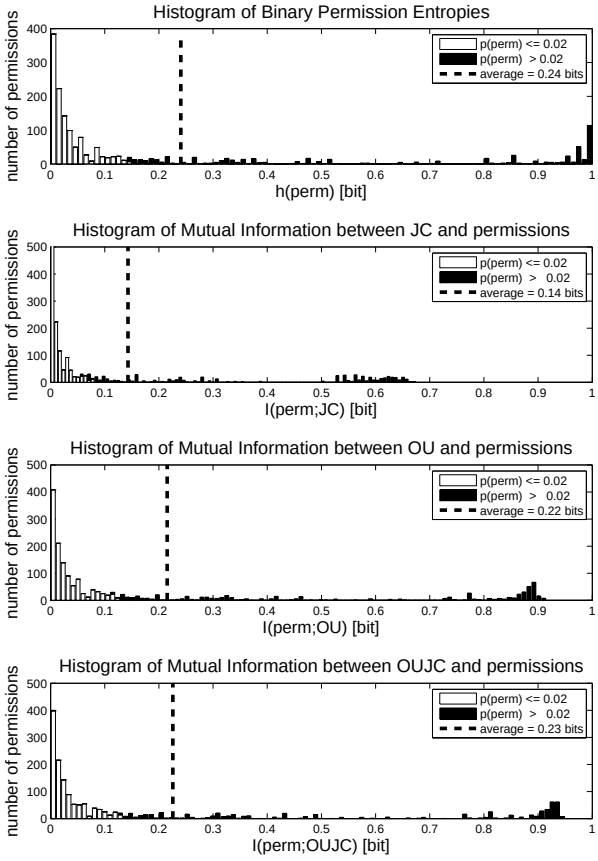


Figure 1: Histograms of the mutual information between permissions and different kinds of business information: organization units, job-codes, and combinations of the two. In the top histogram, the overall entropy of the permissions is shown. The bars for permissions possessed by more than 2% of the users (in black) are stacked on top of the bars for all the other permissions (white).

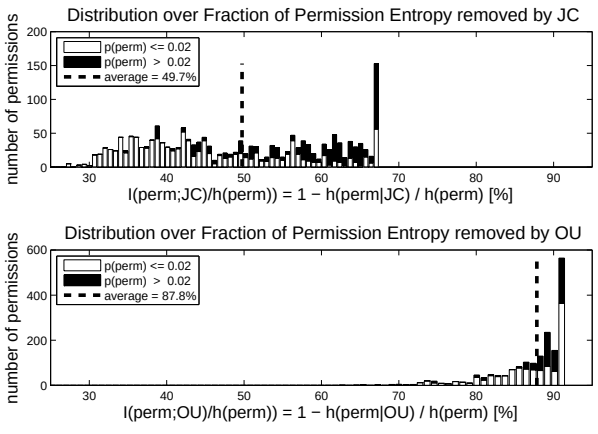


Figure 2: Distribution of the measure of relevance (4), the mutual information weighted with inverse permission entropy.

The top histogram in Figure 1 illustrates the distribution of the permission entropy $h(X_j)$ for the direct user-permission assignment. Since the assignment of a permission is either one or zero, the maximum entropy is one bit, which corresponds to a permission that is possessed by exactly half of the users. Permissions possessed by either very few, or almost all, users have low entropy. For the enterprise under consideration, all permissions with low entropy belong to only a few users. To make this distinction clear, in all of the histograms in Figures 1 and 2, we display counts of permissions shared by less than 2% of the employees in white and the counts of all other permissions in black. As can be seen in the top histogram of Figure 1, most of the permissions have low entropy, but a significant number of permissions have very high entropy. The lower three histograms show (in this order) the distribution of the mutual information between permissions and job-codes, organization unit, and the combination of the two.

The results are surprising. Since the job-code provides an abstract high-level job description, one might expect it to be highly relevant. However, the results show that a user's job-code carries only little information about his permissions. The reason is that, in this enterprise, job-codes are not really abstract *task* descriptions. Instead they express other properties that are interesting for Human Relations, such as the employee's salary class, contract duration, seniority, etc. In contrast, we found that the organization unit is much more relevant for the user's permissions. On average, the organizational unit reduces the entropy by 0.22 bits. Moreover, the entropy of a large number of permissions is even reduced by 0.9 bits (right peak in second lowest histogram of Figure 1). Combining the two attributes yields only a slight gain of 0.01 bits on average (see the lower two histograms of Figure 1). Hence, we conclude that most of the information gained by using job-codes is already contained in the organizational units.

For the bottom three histograms, note that the bimodal distribution of the permission-entropy histogram is preserved: a high peak at very low entropy and a smaller peak at high entropy. This leads us to a general problem in interpreting mutual information. In many cases, the mutual information $I(X_j; S)$ is low simply because the entropy $h(X_j)$ of the permission j is low. This is the case for permissions that almost all users have (for instance, reading email) or, as is usually the case in this data, permissions that very few users have. In Figure 1, we highlighted such permissions in white. This illustrates that almost all permissions whose entropy is not reduced by the knowledge of the given business information have a very low entropy anyway.

To overcome this problem, we weight $I(X_j; S)$ by $1/h(X_j)$ and obtain the relative mutual information $\rho_j(S) = 1 - h(X_j|S)/h(X_j)$ (see Eq. 4) as a relevance measure that indicates the *fraction* of entropy that is explained by S (see Fig. 2). This relative representation better reveals the difference in information content between organization units and job-codes. Whereas, on average, job codes remove roughly 50% of the uncertainty, knowledge of the organization unit removes 88%. Admittedly, there is no way to really determine if knowledge of S *would* decrease more of the permission entropy if $h(X_j)$ *were* higher. Note that for all permissions with high entropy, the mutual information between business information and permissions is high (compare the white and black bars in the lower three histograms of Fig. 1).

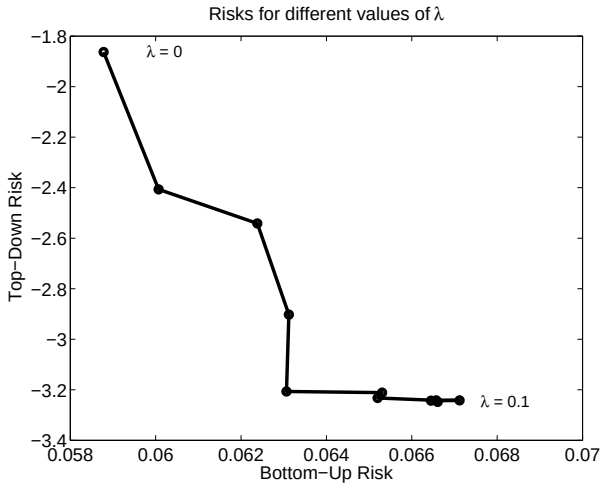


Figure 3: Direct comparison of the both objective functions: log-likelihood $R^{(ll)}$ and pairwise top-down costs $R^{(S)}$ for various values of the linear weight λ.

Therefore one can reason that knowledge of S might possibly (but not necessarily) also provide significant absolute information gain on permissions with low-entropy.

Given these findings, we conclude that for the enterprise considered, the organizational unit provides useful top-down information. The information provided by job-codes is already provided by the organizational unit as can be seen from the very small gain in mutual information when both are used together (compare the two lower histograms in Fig. 1). Therefore, for the data at hand, it is reasonable to ignore the job-codes and just incorporate the organization unit into the role-mining process. In the next section, we will report on several experiments with both types of business information.

6.2 Role Mining Experiments

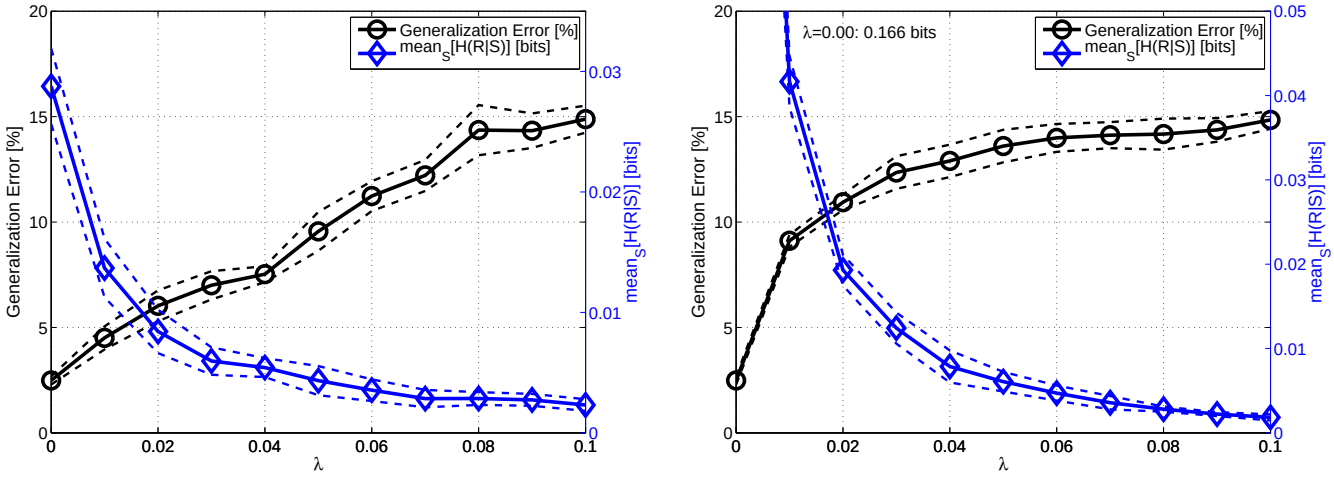
In this section, we evaluate our algorithm on real-world data. As explained in the introduction, the two criteria of a good hybrid role mining solution are:

1. the role decomposition captures most of the given user-permission assignment matrix without overfitting, and
2. the user-role assignment is easy to interpret from a business perspective.

In order to quantitatively assess a given role mining result, we introduce two measures: the *generalization ability* and the *interpretability* of an RBAC system.

Generalization Ability.

We quantify the ability of a set of roles to generalize to new users in order to measure how precisely the underlying structure of a user-permission assignment matrix is captured. To justify this measure, suppose that a new employee enters the company. It is clearly undesirable to re-design the set of roles to accommodate each new employee. Therefore, the existing set of roles should suffice to endow the new employee with all the permissions that he needs to accomplish his job. At the same time, according to the principle of least privilege,



Business Information: Organization Unit

Business Information: Job Codes

Figure 4: Generalization error (black circles) and conditional role entropy (blue diamonds) as a function of the weight of business information λ. Left: hybrid role mining with the organization units of the users. Right: hybrid role mining with job codes. The axes for the generalization error have same scale.

he should not be given any extra permissions beyond those he needs for his job (automated role provisioning is itself an interesting problem, which was recently considered in [16]).

In contrast, a role set that does not generalize to new users might well replace the existing direct user-permission assignments, but must be adapted for each new employee. This is the reason why we did not simply use the deviation $\|\mathbf{x} - \mathbf{z} \otimes \mathbf{u}\|$ to assess how well the structure of the direct assignment is discovered, even though this is often advocated in the context of role mining (e.g. [21]). The deviation measure alone would be appropriate in a lossy-compression scenario. However, it is inappropriate for access control data since it also accounts for the coverage of exceptional and wrong assignments. Moreover, note that a role structure that generalizes well substantially eases the problem of *maintaining* an RBAC configuration as an enterprise evolves, provided that the functions in the enterprise do not dramatically change during this evolution.

We propose the following two-step experiment to measure the generalization ability of a role mining algorithm:

1. The algorithm gets only a subset of the users-permission assignment as the input for finding a role decomposition.
2. For each user from a second, disjoint subset, we take the business attribute (e.g. his organizational unit) plus a small fraction of his permissions to identify the roles that best suit him. From these roles, all other permissions of the user are predicted.

Since the second (the hidden) set of users comes from the same enterprise (with the same probability distribution on the user-permission assignment), a role mining solution that generalizes well should be able to give a precise prediction.

We implement the experiment as follows. From the entire user-permission assignment and business information, we randomly choose a set of 3000 users. These are used to infer a set of roles, using our method described in Section 5.3. From the remaining users, we randomly choose a fraction κ of permissions (e.g. 10%). These permissions, together with the business information, is then used to choose the best-matching role set. Formally, we compute the total risk (12), where, to compute the log-likelihood costs (10), we only sum over the revealed permissions j_κ. Each remaining user i is then assigned to the set of roles $\hat{\mathcal{L}}_i$ with minimal risk, i.e.

$$\hat{\mathcal{L}}_i := \arg\min_{\mathcal{L}\in\mathbb{L}} R_{i,\mathcal{L}} \,. \tag{24}$$

The set of roles is then used to predict all permissions of user i. Let $\hat{z}_{i\cdot}$ be the assignment to the best-matching set of roles. The estimated permissions of user i are then computed as $\hat{x}_{i\cdot} = \hat{z}_{i\cdot} \otimes \hat{\mathbf{u}}$, where $\hat{\mathbf{u}}$ is the role-permission relation found in the role mining step. Finally, the relative Hamming distance between the estimated and true permissions of user i, $\|\hat{x}_{i\cdot} - x_{i\cdot}\|/D$, i.e. the fraction of wrongly predicted permissions, is the measure for the generalization error of the inferred roles.

Interpretability.

We formulate the second measure, the business relevance of the role assignments, by the conditional entropy of the role set $\mathcal{L}_i$ of a user i, given his business information s_i, i.e. $h(\mathcal{L}_i|s_i)$. This captures the requirement that all users with the same business attribute should obtain the same set of roles. Thus, the knowledge of the business attribute should, ideally, determine the roles an employee is assigned to. A set of roles, however, might be shared by users with different business attributes. Note that this measure resembles the relevance analysis for business information that we introduced in Section 4. There, we required that the given business information should have a high mutual information with the permissions (recall $I(X_j; S) = h(X_j) - h(X_j|S)$) in order to agree with the permission structure (and therefore be useful for role mining). Following the same line of reasoning, we require roles to agree with the business information. Role decompositions that fulfill this requirement are easy to interpret from the business perspective.

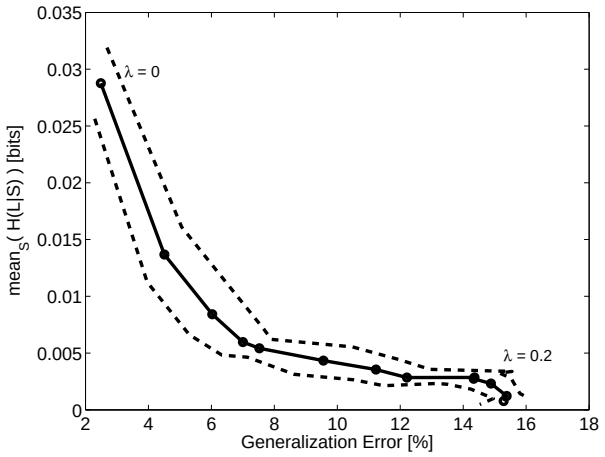


Figure 5: Generalization error versus conditional role entropy for the experiment using organization units. The dashed lines represent the standard deviation over ten repeated experiments.

Results.

We carried out hybrid role mining experiments for two different types of business information. The first experiment uses the organization units as business information, as suggested by our analysis in Section 6.1. Our second experiment uses the users' job-codes. In Figure 4, we plot the two described measures for both kinds of business information (OE left, JC right). For the OE-experiment, we directly compare these two measures, as displayed in Figure 5.

Figure 4 (left) compares, for different weighting factors λ, the two measures on the role decompositions that were discovered. The case with $\lambda = 0$ corresponds to pure bottom-up role mining without business information. While the generalization error slightly increases with λ, the mean conditional role set entropy given the business information decreases drastically if λ is increased from zero to a small value. Hence, the correspondence between the user-role assignment and the users' organizational unit substantially improves as business information is taken into account. Even for small values of λ, the roles can be better interpreted as business roles. Since this gain in the business meaning of the user-role assignment comes at the expense of only a small decrease in the generalization ability of the roles, it is a price worth paying. For $\lambda > 0.04$, the entropy $h(\mathcal{L}_i|s_i)$ does not substantially further improve whereas the prediction error rises. In this interval, the two parts of our objective function are antagonistic and hence a further increase of λ is not desirable.

The right part of Figure 4 displays the results obtained by using the job-codes as business information. In order to compare the two experiments, Figure 4 shows both results for axes with the same scale. Note that the two trends of the conditioned role entropy (blue diamonds) cannot be directly compared since they are computed with respect to the two different types of business information. However, one can reason about the generalization error, which is in both cases computed with respect to the same user-permission assignment. While for the job-codes (right graph) the generalization error converges exponentially to the maximum with λ, it converges only linearly for the organization units. For low λ, it is possible to substantially improve on the interpretability of the role decomposition while preserving good generalization ability.

For hybrid role mining with job-codes this is not possible. The generalization ability increases immediately for even small λ. This result confirms the findings of our analysis of these two types of business information carried out in Section 6.1. The job-codes do not agree as well with the direct user-permission assignment as the organizational units do. Hence, using job-codes, it is only possible to trade off generalization ability for business interpretability. However, with organization units, one can improve the business interpretability without substantially increasing the generalization error.

We directly compare the two quality measures with each other for different values of λ in Figure 5 for the experiment with the organizational units. The graph shows that it is possible to improve the results of role mining by using our unified objective function to incorporate business information into the role mining process: Changing λ along the straighter parts of the curve gives improved solutions, whereas the more curved parts mark the solutions that are Pareto-optimal with respect to the two measures. In a concrete application, the trade-off between generalization and interpretability must be chosen such that the side conditions are met. For example, one might require that no more than some given percentage of permissions of a new employee are wrongly predicted by the solution. Viewed more generally, optimizing generalization performance and interpretability is a multi-objective optimization problem.

7. CONCLUSION

We have divided the hybrid role mining problem into two parts and provided solutions for them: determining the relevance of business information for role mining, and incorporating this information into a hybrid role mining algorithm. We solved the first problem with an entropy-based measure of relevance and the second by deriving an objective function that combines a probabilistic model of RBAC with business information.

To validate our solutions, we carried out experiments using actual enterprise data. The results show that our approach finds roles with the following properties: they generalize well, they are easy to interpret (i.e., intuitively understandable) from the business perspective, and they have high predictability in that they approximate closely the given user-permission assignment. All of these properties are desirable as they have direct, positive consequences for the administration and maintenance of RBAC-based systems. Generalization facilitates the maintenance of RBAC since new users can be easily equipped with needed permissions without creating new roles. Interpretable roles simplify both the role's life-cycle management and adding new users to the system. Finally, predictability leads to increased security, since predictive roles implement closely the authorization policy given by the original user-permission assignment.

As future work, we will investigate an adaptive weighting scheme of the business information. Namely, not every value of a business attribute might be equally descriptive for the permissions received by a user with this attribute. Adaptive weighting might enable an even more adaptive inclusion of specific business attributes. Furthermore, we will explore

extensions of our approach for analyzing and merging two given RBAC systems while preserving their inherent business semantic. The quantitative methods of our approach could also be a starting point to learn about the relationship between the access control data and the business structure of both domains.

Acknowledgments.

This work was partially supported by the Zurich Information Security Center. It represents the views of the authors. The work of AS was in part funded by CTI grant Nr. 8539.2;2 EPSS-ES.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[2] J. Buhmann and H. Kühnel. Vector quantization with complexity costs. In *IEEE Trans on Information Theory*, volume 39, pages 1133–1145, 1993.

[3] A. Colantonio, R. Di Pietro, and A. Ocello. A cost-driven approach to role engineering. In *Proceedings of the 23rd ACM Symposium on Applied Computing, SAC '08*, volume 3, pages 2129–2136, Fortaleza, Ceará, Brazil, 2008.

[4] A. Colantonio, R. Di Pietro, A. Ocello, and N. V. Verde. A formal framework to elicit roles with business meaning in RBAC systems. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09*, 2009.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[6] E. J. Coyne. Role engineering. In *RBAC '95: Proceedings of the 1st ACM Workshop on Role-based access control*, page 4, New York, NY, USA, 1996.

[7] R. Crook, D. Ince, and B. Nuseibeh. Towards an analytical role modelling framework for security requirements. In *Proc. of the 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)*, pages 9–10, 2002.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[9] P. Epstein and R. Sandhu. Engineering of role/permission assignments. In *ACSAC '01: Proceedings of the 17th Annual Computer Security Applications Conference*, page 127, Washington, DC, USA, 2001. IEEE Computer Society.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[11] M. Frank, D. Basin, and J. 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.

[12] B. Ganter, G. Stumme, and R. Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.

[13] M. Kuhlmann, D. Shohat, and G. Schimpf. Role mining – revealing business roles for security administration using data mining technology. In *SACMAT '03: Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 179–186, New York, NY, USA, 2003. ACM.

[14] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. Calo, and J. Lobo. Mining roles with semantic meanings. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 21–30, New York, NY, USA, 2008. ACM.

[15] G. Neumann and M. Strembeck. A scenario-driven role engineering process for functional RBAC roles. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 33–42, New York, NY, USA, 2002. ACM.

[16] Q. Ni, J. Lobo, S. Calo, P. Rohatgi, and E. Bertino. Automating role-based provisioning by learning from examples. In *SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 75–84, New York, NY, USA, 2009. ACM.

[17] H. Roeckle, G. Schimpf, and R. Weidinger. Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 103–110, New York, NY, USA, 2000. ACM.

[18] K. Rose, E. Gurewitz, and G. Fox. Vector quantization by deterministic annealing. In *IEEE Trans on Information Theory*, volume 38, pages 2210–2239, 1992.

[19] J. Schlegelmilch and U. Steffens. Role mining with ORCA. In *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 168–176, New York, NY, USA, 2005. ACM.

[20] A. P. Streich, M. Frank, D. Basin, and J. M. Buhmann. Multi-assignment clustering for boolean data. In *Proceedings of the 26th International Conference on Machine Learning*, pages 969–976, Montreal, June 2009. Omnipress.

[21] J. Vaidya, V. Atluri, and Q. Guo. The Role Mining Problem: Finding a minimal descriptive set of roles. In *The Twelth ACM Symposium on Access Control Models and Technologies*, pages 175–184, Sophia Antipolis, France, 2007. ACM.

[22] J. Vaidya, V. Atluri, and J. Warner. Roleminer: Mining roles using subset enumeration. In *CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, New York, NY, USA, 2006. ACM.

[23] D. Zhang, K. Ramamohanarao, and T. Ebringer. Role engineering using graph optimisation. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 139–144, New York, NY, USA, 2007. ACM.