Bottom-up Cell Suppression that Preserves the Missing-at-random Condition

Yoshitaka Kameya and Kentaro Hayashi

Dept. of Information Engineering, Meijo University
1-501 Shiogama-guchi, Tenpaku-ku, Nagoya 468-8502, Japan
ykameya@meijo-u.ac.jp

Abstract. This paper proposes a cell-suppression based k-anonymization method which keeps minimal the loss of utility. The proposed method uses the Kullback-Leibler (KL) divergence as a utility measure derived from the notions developed in the literature of incomplete data analysis, including the missing-at-random (MAR) condition. To be more specific, we plug the KL divergence into an bottom-up, greedy procedure for a local recoding k-anonymization as a cost function which is efficiently computed. We focus on classification datasets and experimental results exhibit that the proposed method yields a small degradation of classification performance when combined with naive Bayes classifiers.

Keywords: k-anonymity, cell suppression, missing-at-random condition

1 Introduction

Generally, in data mining, fine-grained datasets tend to produce sharper, and accordingly, more useful results. However, when the datasets are human-related, such fineness may lead to re-identification of a person and disclosure of his/her privacy. Re-identification is not only possible from explicit identifiers but from a combination of common personal attributes e.g. age and gender. Such attributes are called quasi-identifiers or QIDs. In privacy-preserving data publishing [1, 5, 22], we often modify QIDs so that both the risk of re-identification and the loss of utility of the dataset are kept minimal at the same time.

k-Anonymity [16, 18] is a well-known privacy requirement on a tabular dataset that, for every combination of QIDs existing in a tuple, at least $k-1$ other tuples must have the same combination of QIDs. Under k-anonymity with a sufficiently large k, the risk of re-identification of a person will be small, since its probability is at most $1/k$. Modifying QIDs in the original dataset so that k-anonymity is satisfied is called k-anonymization. k-Anonymity is attractive in its simplicity and intuitiveness, but it is often quite costly in k-anonymization to fully minimize the loss of the utility of the dataset. For instance, minimizing the number of suppressed cells under k-anonymity is NP-hard [14].

Despite such a discouraging formal result, dozens of practical k-anonymization methods have been proposed. One grouping criterion among these methods is the range to which an anonymization operator is applied. In global recoding [2, 13,

16, 19], we replace all occurrences of a value with another general value, while in local recoding [7, 12, 21], we just replace an occurrence of a value independently of other occurrences. Cell suppression is a typical local recoding operator in which we replace a cell value with a null value. One advantage of suppressing cell values over generalizing them is that the former requires no hierarchical knowledge, and another advantage is that there have been statistical tools including classifiers that can work with suppressed (i.e. missing) data.

In this paper, we propose a cell-suppression based k-anonymization method which keeps minimal the loss of utility using the notion from incomplete data analysis, including the missing-at-random (MAR) condition [15, 17]. Kifer and Gehrke [11] formulated anonymized datasets in a probabilistic setting and introduced as a utility measure the Kullback-Leibler (KL) divergence between two empirical distributions, one from the original dataset and the other from the anonymized one. One contribution of this paper is to justify their utility measure from the viewpoint of preserving the MAR condition. An underlying key observation here is that anonymization is an artificial, explicit process that forces the original dataset to be ambiguous or incomplete for avoiding re-identification, whereas traditional incomplete data analysis deals with incomplete datasets as they are, assuming a hidden generation process of the datasets [17]. Another contribution is that we plug the KL divergence into an bottom-up, greedy procedure for a local recoding k-anonymization [7, 21] as a cost function which is efficiently computed. We focus on classification datasets where different anonymizations are clearly compared from the viewpoint of utility, though the proposed method can also deal with non-classification datasets.

The rest of this paper is outlined as follows. First, we introduce several background notions and notations in Section 2. Then, Section 3 describes the proposed method in detail. Experimental results are presented in Section 4. Section 5 concludes the paper with some discussions on open problems and related work.

2 Background

2.1 Preliminaries

We begin by introducing some background notions and notations used in the paper. The dataset we consider is a tabular classification dataset of size N with M attributes. We also consider a null value $\perp_j$ at the j-th attribute. In addition, $\mathcal{C}$ is a set of pre-defined classes, and $\mathcal{V}_j$ is a set of discrete non-null values of the j-th attribute. Then, a tuple is comprised of M attribute values and a class label from $\mathcal{C}$, i.e. it is an element of $\mathcal{V}'_1 \times \mathcal{V}'_2 \times \cdots \times \mathcal{V}'_M \times \mathcal{C}$, where $\mathcal{V}'_j = \mathcal{V}_j \cup \{\perp_j\}$. A tuple t is written as $(\boldsymbol{y}, c)$, where $\boldsymbol{y}$ of a vector $(y_1, y_2, \ldots, y_M)$ of attribute values. A dataset $\mathcal{D}$ is a multiset $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$ of tuples. Throughout the paper, i indicates the index of a tuple in a dataset ($1 \leq i \leq N$), and j indicates the index of an attribute ($1 \leq j \leq M$).

Suppressing a non-null attribute value is to replace it with a null value $\perp_j$. It is obvious that suppression is exactly a generalization along a two-level hierarchy

where the top-level corresponds to $\perp_j$, and the bottom-level only includes raw values from $\mathcal{V}_j$. In incomplete data analysis [17], null or suppressed values are called *missing values*. A tuple $t = (\boldsymbol{x}, c)$ is complete if it contains no missing values, i.e. is an element of $\mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_M \times \mathcal{C}$. A dataset is complete if it has no incomplete tuples. For non-classification datasets, it is sufficient to consider that $\mathcal{C}$ only contains one dummy class.

Given a complete dataset $\mathcal{D}$, we may use statistics such as $N(\boldsymbol{y}, c) = |\{t^{(i)} \in \mathcal{D} \mid t^{(i)} = (\boldsymbol{y}, c)\}|$, $N(c) = |\{t^{(i)} \in \mathcal{D} \mid t^{(i)} = (\cdot, c)\}|$, $N(\boldsymbol{y}) = |\{t^{(i)} \in \mathcal{D} \mid t^{(i)} = (\boldsymbol{y}, \cdot)\}|$, $N(y_j, c) = |\{t^{(i)} \in \mathcal{D} \mid y_j \text{ is the } j\text{-th attribute value of } t^{(i)} = (\cdot, c)\}|$ and so on. In a probabilistic setting, we introduce a probability distribution $p(\boldsymbol{x}, c)$ over complete tuples $(\boldsymbol{x}, c) = (x_1, x_2, \ldots, x_M, c)$ and compute empirical probabilities $\hat{p}(c) = (N(c) + \alpha)/(N + \alpha|\mathcal{C}|)$ and $\hat{p}(x_j \mid c) = (N(x_j, c) + \alpha)/(N(\neg\perp_j, c) + \alpha|\mathcal{V}_j|)$ for each class c and non-null value x_j. Here, α is non-negative number called the pseudo count, and α prevents the empirical probabilities from being zero when $\alpha > 0$. Throughout the paper, we configure $\alpha = 1$, which results in so-called Laplace smoothing. On the other hand, $N(\neg\perp_j, c)$ denotes the sum of the occurrences of non-null values together with class c, i.e. we have $N(\neg\perp_j, c) = \sum_{x \in \mathcal{V}_j, x \neq \perp_j} N(x, c) = N(c) - N(\perp_j, c)$. Furthermore, a null or suppressed value $\perp_j$ means taking any value in $\mathcal{V}_j$, so its (conditional) probability should always be one. Specifically, we have $p(\perp_j \mid c) = \hat{p}(\perp_j \mid c) = 1$.

2.2 *k*-Anonymity

Here, we describe k-anonymity formally with some additional notations. First, for simplicity, we assume that all attributes $\boldsymbol{y}$ except the class label c in a tuple t are QIDs and focus on reducing the risk of re-identification of a person from QIDs. Whereas $\mathcal{D}$ was defined as a *multiset*, it is often convenient to transform $\mathcal{D}$ into a pair of a tuple set $\mathcal{S}$ and a count table $\mathcal{N}$. $\mathcal{S}$ is defined as $\{\boldsymbol{y} \mid (\boldsymbol{y}, c) \in \mathcal{D}\}$, i.e. an *ordinary* set of distinct tuples. The count table $\mathcal{N}$, on the other hand, stores $N(\boldsymbol{y}, c)$, $N(c)$, $N(\boldsymbol{y})$ and $N(y_j, c)$ in the previous section when needed. It is straightforward to generate an equivalent dataset from $\mathcal{S}$ and $\mathcal{N}$. From the settings above, k-anonymity of a dataset $\mathcal{D}$ is restated as $\min_{(\boldsymbol{y}, \cdot) \in \mathcal{S}} N(\boldsymbol{y}) \geq k$.

2.3 Bottom-up Cell Suppression

In this paper, we adopt a bottom-up, greedy algorithm for local recoding k-anonymization algorithm, which is a simplified adaptation of the one used in [7, 21] into the case of cell-suppression in classification datasets. The algorithm, shown in Algorithm 1, 2 and 3, resembles agglomerative clustering.[1]

The ANONYMIZE procedure is the main routine of the algorithm. The procedure takes as input the anonymity threshold k and the original dataset $\mathcal{D}$ and returns a k-anonymized version of $\mathcal{D}$. The tuple set $\mathcal{S}$ and the count table $\mathcal{N}$

[1] Agglomerative clustering is a typical hierachical clustering in which we start with initial clusters containing single tuples and merge the closest pair of clusters in a bottom-up manner [10].

Algorithm 1 ANONYMIZE$(k, \mathcal{D})$

Require: k: the anonymity to achieve, $\mathcal{D}$: the original dataset
1: Construct the tuple set $\mathcal{S}$ and the count table $\mathcal{N}$ from $\mathcal{D}$
2: Obtain the empirical probability function $\hat{p}$ from $\mathcal{S}$ and $\mathcal{N}$
3: **while** $\min_{(\boldsymbol{y},\cdot)\in\mathcal{S}} N(\boldsymbol{y}) < k$ **do**
4: Pick up $t = (\boldsymbol{y}, c)$ such that $N(\boldsymbol{y}) < k$ randomly from $\mathcal{S}$
5: $t^* := \mathrm{argmin}_{t'=(\boldsymbol{y}',c)\in\mathcal{S}} \mathit{\Gamma}(t, t', \hat{p}, \mathcal{N})$
6: $u :=$ SUPPRESS(t, t^*)
7: UPDATE$(u, t, t^*, \mathcal{S}, \mathcal{N})$
8: **end while**
9: Construct $\mathcal{D}'$ from $\mathcal{S}$ and $\mathcal{N}$
10: **return** $\mathcal{D}'$

Algorithm 2 SUPPRESS(t, t')

Require: t, t': tuples of the same class c to be suppressed
1: Let t be $(y_1, y_2, \ldots, y_M, c)$ and t' be $(y'_1, y'_2, \ldots, y'_M, c)$
2: **return** $u = (u_1, u_2, \ldots, u_M, c)$ s.t. $u_j = y_j$ (if $y_j = y'_j$) or $u_j = \perp_j$ (if $y_j \neq y'_j$)

of $\mathcal{D}$ are used inside the procedure (Line 1). Empirical probability function $\hat{p}$ w.r.t. the original dataset $\mathcal{D}$, which will be referred to in computing the suppression cost, is then obtained from $\mathcal{S}$ and $\mathcal{N}$ (Line 2). The procedure repeatedly chooses two tuples and merges them by suppression until no tuple violates the k-anonymity requirement (Lines 3–8). Specifically, we randomly pick up a tuple t from violating tuples (Line 4) and choose the best counterpart t^* of the same class (Line 5) that minimizes the suppression cost $\mathit{\Gamma}$ in the case of t and t^* being suppressed and merged. The suppression is actually done by the SUPPRESS procedure (Line 6). Then, the UPDATE procedure replaces two chosen tuples (t and t^*) in $\mathcal{S}$ with the merged one (u) and updates the count table $\mathcal{N}$ (Line 7).

The choice of the cell-suppression cost $\mathit{\Gamma}$ is crucial since it reflects the utility of the dataset which we wish to exploit. One simple cost function is $\mathit{\Gamma}_{\mathrm{ham}}$, the one based on the Hamming distance, which is computed as:

$$\mathit{\Gamma}_{\mathrm{ham}}(t, t', \hat{p}, \mathcal{N}) \stackrel{\mathrm{def}}{=} N(\boldsymbol{y}, c)H(\boldsymbol{y}, \boldsymbol{u}) + N(\boldsymbol{y}', c)H(\boldsymbol{y}', \boldsymbol{u}), \tag{1}$$

where $t = (\boldsymbol{y}, c)$, $t' = (\boldsymbol{y}', c)$, $u = (\boldsymbol{u}, c)$ is the tuple to be generated by SUPPRESS(t, t'), and $H(\boldsymbol{a}, \boldsymbol{b})$ is the number of conflicting elements between $\boldsymbol{a}$ and $\boldsymbol{b}$ (null values and non-null values are considered distinct). $\mathit{\Gamma}_{\mathrm{ham}}$ is exactly the total number of cells to be suppressed further and does neither use the empirical probabilities $\hat{p}$ in the original dataset nor the current counts from $\mathcal{N}$. We may also use a cost function $\mathit{\Gamma}_{\mathrm{info}}$, which is based on information loss [7]:[2]

$$\mathit{\Gamma}_{\mathrm{info}}(t, t', \hat{p}, \mathcal{N}) \stackrel{\mathrm{def}}{=} -\sum_{j:y_j\neq y'_j} \left(N(\boldsymbol{y}, c) \log \hat{p}(y_j \mid c) + N(\boldsymbol{y}', c) \log \hat{p}(y'_j \mid c) \right), \tag{2}$$

[2] To be precise, the original definition by Harada et al. [7] does not consider classification datasets.

Algorithm 3 UPDATE$(u, t, t', \mathcal{S}, \mathcal{N})$

Require: u: a new tuple, t and t': old tuples, $\mathcal{S}$: a tuple set, $\mathcal{N}$: a count table
1: Remove t and t from $\mathcal{S}$
2: Let u be $(\boldsymbol{u}, c)$, t be $(\boldsymbol{y}, c)$ and t' be $(\boldsymbol{y}', c)$
3: **if** $u \in \mathcal{S}$ **then**
4: $\quad N(\boldsymbol{u}, c) := N(\boldsymbol{u}, c) + N(\boldsymbol{y}, c) + N(\boldsymbol{y}', c)$
5: **else**
6: $\quad N(\boldsymbol{u}, c) := N(\boldsymbol{y}, c) + N(\boldsymbol{y}', c)$
7: $\quad \mathcal{S} := \mathcal{S} \cup \{u\}$
8: **end if**
9: Remove all entries of $\mathcal{N}$ concerning t and t'

where $t = (y_1, y_2, \ldots, y_M, c)$ and $t' = (y'_1, y'_2, \ldots, y'_M, c)$. Γ_{info} uses empirical probabilities $\hat{p}(y_j \mid c)$ (y_j is a non-null value x_j or a null value $\perp_j$) computed from the original dataset as shown in Section 2.1. The term $-\log \hat{p}(y_j \mid c)$ is the self-information of the j-th attribute taking y_j. Since the self-information of the j-th attribute taking $\perp_j$ is $-\log \hat{p}(\perp_j \mid c) = -\log 1 = 0$, replacing a non-null value x_j with $\perp_j$ loses the information $-\log \hat{p}(x_j \mid c)$. As a result, $\Gamma_{\text{info}}(t, t', \hat{p}, \mathcal{N})$ measures the total amount of information loss in suppressing and merging t and t'. Obviously, the k-anonymization procedure in Section 2.3 tends to suppress frequent attribute values when combined with Γ_{info}.

3 The Proposed Method

As said before, we propose a cell-suppression based k-anonymization method which keeps minimal the loss of utility using the notion from incomplete data analysis. In this method, we consider that anonymization is an artificial process that forces the original dataset $\mathcal{D}$ to be ambiguous so as to avoid re-identification of persons. It is then desirable to control such an anonymization process for ensuring the soundness of later statistical inferences such as classification. In the literature of incomplete data analysis, it is proved that, under the missing-at-random (MAR) condition [15, 17], the process where some portion of the original dataset $\mathcal{D}$ turns to be missing is ignorable in the inference related to the empirical probability distribution of $\mathcal{D}$. In our context, the MAR condition allows us to obtain empirical probabilities from an anonymized dataset ignoring the anonymization process without distortion.

From the observations above, our k-anonymization method attempts to preserve the MAR condition as well as possible. More precisely, we present a cell-suppression cost function reflecting the deviation from the MAR condition and use it in the k-anonymization procedure introduced in Section 2.3. To measure the deviation from the MAR condition, we consider the Kullback-Leibler (KL) divergence in naive Bayes classifiers. In the rest of this section, we will describe these relevant notions in turn.

3.1 Naive Bayes Classification

In classification, we use Naive Bayes [20] as a primary classifier. Naive Bayes assumes that attributes in a classification dataset are conditionally independent of each other, given the class. Despite its strong independence assumption, naive Bayes often works surprisingly well in classifying real datasets [4]. Formally, it is assumed that the probability that a complete tuple $t = (\boldsymbol{x}, c) = (x_1, x_2, \ldots, x_M, c)$ occurs is simplified as $p(t) = p(\boldsymbol{x}, c) = p(c) \prod_j p(x_j \mid c)$. Typically, classification is performed in two steps: we first learn the empirical probabilities $\hat{p}(c)$ and $\hat{p}(x_j \mid c)$ from the complete training dataset, and then predict the most plausible class $c^* = \mathrm{argmax}_{c \in \mathcal{C}}\ \hat{p}(c \mid \boldsymbol{x}) = \mathrm{argmax}_c\ \hat{p}(c) \prod_j\ \hat{p}(x_j \mid c)$ for an unseen data having attribute values $\boldsymbol{x}$.

The independence assumption in naive Bayes also makes it simple to handle incomplete data. That is, noting that $p(\perp_j \mid c) = 1$, the probability that an incomplete tuple $(y_1, y_2, \ldots, y_M, c)$ occurs, where y_j is a non-null value from $\mathcal{V}_j$ or a null value $\perp_j$, is obtained as $p(c) \prod_{j: y_j \neq \perp_j} p(y_j \mid c)$, where null values are all ignored. Similarly, one may learn the empirical probabilities $\hat{p}(x_j \mid c)$ as described in Section 2.1 for a non-null value x_j, as if there are no missing values from the beginning. This is a standard way of learning called maximum likelihood (ML) estimation,[3] which is also applicable to anonymized datasets. However, in general, justifying ML estimation requires some extra condition on the process how missing data are generated. The MAR condition explained next is one of such conditions.

3.2 The Missing-at-random Condition

The Process of Anonymization. As said before, under the MAR condition, a standard learning procedure of naive Bayes classifiers is justified even with anonymized datasets. Conversely, to obtain a naive Bayes without distortion brought by anonymization, it is reasonable to anonymize the original dataset so that the MAR condition is preserved.

First, let us model our anonymization process by an analogy to the process of generating missing data [17]. We focus on classification datasets where no class labels will be missing or suppressed. Given an original dataset $\mathcal{D}$ having a complete tuple $(\boldsymbol{x}, c) = (x_1, x_2, \ldots, x_M, c)$, we may anonymize it into an incomplete dataset having $(\boldsymbol{y}, c) = (y_1, y_2, \ldots, y_M, c)$ by suppressing some part of $\boldsymbol{x}$. A binary indicator $\boldsymbol{r} = (r_1, r_2, \ldots, r_M)$ says which part has been suppressed, i.e. $y_j = x_j$ iff $r_j = 1$, or $y_j = \perp_j$ iff $r_j = 0$. Note that, given an incomplete attribute values $\boldsymbol{y}$, the indicator $\boldsymbol{r}$ is uniquely determined. The joint probability of the whole anonymization process behind $\boldsymbol{y}$ is then introduced and we decompose

[3] To be precise, learning empirical probabilities using the pseudo count α, shown in Section 2.1, is called maximum a posteriori (MAP) estimation. ML estimation is a special case of MAP estimation where $\alpha = 0$. The following discussions can be easily extended to the case of MAP estimation.

it into two factors:[4]

$$p(\boldsymbol{r}, \boldsymbol{x}, c \mid \theta, \phi) = p(\boldsymbol{r} \mid \boldsymbol{x}, c, \phi) p(\boldsymbol{x}, c \mid \theta). \tag{3}$$

Here, $p(\boldsymbol{x}, c \mid \theta)$ is the probability that a complete, original tuple $(\boldsymbol{x}, c)$ occurs, and $p(\boldsymbol{r} \mid \boldsymbol{x}, c, \phi)$ is the probability that the suppressed pattern is $\boldsymbol{r}$ given such a complete tuple. The latter is called *the missing-data mechanism* and models our choice in anonymization. θ denotes the parameters of the probability distribution over complete tuples, and ϕ denotes the parameters for the missing-data mechanism. Since anonymization is an artificial operation subsequently performed after the original dataset has been obtained, it is natural to think that there is no overlap between θ and ϕ.

Learning under the MAR Condition. Given an incomplete values $\boldsymbol{y}$, we define $\boldsymbol{x}_{\text{obs}}$ as a collection of x_j's where $r_j = 1$, and $\boldsymbol{x}_{\text{mis}}$ as a collection of x_j's where $r_j = 0$. Thus, $\boldsymbol{x}_{\text{obs}}$ (resp. $\boldsymbol{x}_{\text{mis}}$) denotes the observed or non-suppressed (resp. missing or suppressed) part of $\boldsymbol{x}$. The probability that an incomplete tuple $(\boldsymbol{y}, c)$ occurs is then computed as $p(\boldsymbol{y}, c \mid \theta, \phi) = \sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{r}, \boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \theta, \phi)$ where $\boldsymbol{r}$ is compatible with $\boldsymbol{y}$, and $\boldsymbol{x} = (\boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}})$.

Next, let us consider the procedure for learning a naive Bayes classifier from the dataset $\{(\boldsymbol{y}, c)\}$ which contains only one tuple.[5] As said earlier, one standard learning procedure is ML estimation, where we attempt to maximize the likelihood of the whole process $L(\theta, \phi) = p(\boldsymbol{y}, c)$ by adjusting the parameters θ and ϕ. In other words, we obtain $(\hat{\theta}, \hat{\phi}) = \text{argmax}_{\theta,\phi} L(\theta, \phi)$. Now we assume that the MAR condition is satisfied. The MAR condition states that *the choice in suppression does not depend on the value to be suppressed itself.* This condition is formally written as $\forall \boldsymbol{x}, c\, p(\boldsymbol{r} \mid \boldsymbol{x}, c, \phi) = p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c, \phi) = p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, c, \phi)$. Then, the likelihood $L(\phi, \theta)$ is transformed as follows:

$$\begin{aligned}
L(\phi, \theta) &= p(\boldsymbol{y}, c \mid \phi, \theta) = \textstyle\sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{r}, \boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \phi, \theta) \\
&= \textstyle\sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c, \phi) p(\boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \theta) && (4) \\
&= \textstyle\sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, c, \phi) p(\boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \theta) && (5) \\
&= p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, c, \phi) \textstyle\sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \theta) \\
&= p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, c, \phi) L'(\theta), && (6)
\end{aligned}$$

where $L'(\theta) = \sum_{\boldsymbol{x}_{\text{mis}}} p(\boldsymbol{x}_{\text{obs}}, \boldsymbol{x}_{\text{mis}}, c \mid \theta) = p(\boldsymbol{x}_{\text{obs}}, c \mid \theta)$ is the likelihood of the anonymized dataset, ignoring the anonymization process. The MAR condition derives Eq. 5 from Eq. 4. Since $p(\boldsymbol{r} \mid \boldsymbol{x}_{\text{obs}}, \phi)$ is constant w.r.t. θ, Eq. 6 says that, for any ϕ, maximizing $L(\phi, \theta)$ and maximizing $L'(\theta)$ yield the same parameters $\hat{\theta}$, i.e. our anonymization is not influential on learning $\hat{\theta}$, as long as the MAR

[4] Joint distributions decomposed in this way are called selection models [17].

[5] Extending the discussion to the case with multiple i.i.d. (independent and identically distributed) tuples $\{(\boldsymbol{y}^{(1)}, c^{(1)}), (\boldsymbol{y}^{(2)}, c^{(2)}), \ldots, (\boldsymbol{y}^{(N)}, c^{(N)})\}$ is fairly straightforward, since the likelihood can be transformed as $L(\theta, \phi) = \prod_i p(\boldsymbol{y}^{(i)}, c^{(i)}) = (\prod_i p(\boldsymbol{r}^{(i)} \mid \boldsymbol{x}^{(i)}, c^{(i)}, \phi))(\prod_i p(\boldsymbol{x}^{(i)}, c^{(i)} \mid \theta))$, where $\boldsymbol{x}^{(i)}$ is the original of $\boldsymbol{y}^{(i)}$.

condition is preserved. The probability $p(\ldots \mid \hat{\theta})$ under the learned parameters $\hat{\theta}$ coincides with the empirical probability $\hat{p}(\ldots)$ used throughout the paper.

The KL Divergence for Examining the MAR Condition. The next question is how to preserve the MAR condition in anonymization. First, the MAR condition $\forall \boldsymbol{x}, c\ p(\boldsymbol{r} \mid \boldsymbol{x}_{\mathrm{obs}}, \boldsymbol{x}_{\mathrm{mis}}, c, \phi) = p(\boldsymbol{r} \mid \boldsymbol{x}_{\mathrm{obs}}, c, \phi)$ can always be rewritten as $\forall \boldsymbol{x}, c\ p(\boldsymbol{x}_{\mathrm{mis}} \mid \boldsymbol{x}_{\mathrm{obs}}, \boldsymbol{r}, c, \phi) = p(\boldsymbol{x}_{\mathrm{mis}} \mid \boldsymbol{x}_{\mathrm{obs}}, c, \phi)$. By the independence assumption in naive Bayes, this is simplified as $\forall \boldsymbol{x}_{\mathrm{mis}}, c\ p(\boldsymbol{x}_{\mathrm{mis}} \mid \boldsymbol{r}_{\mathrm{mis}} = \boldsymbol{0}, c, \phi) = p(\boldsymbol{x}_{\mathrm{mis}} \mid c, \phi)$, where $\boldsymbol{x}_{\mathrm{obs}}$ and $\boldsymbol{x}_{\mathrm{mis}}$ are independent given c, and $\boldsymbol{r}_{\mathrm{mis}}$ is the portion of $\boldsymbol{r}$ corresponding to $\boldsymbol{x}_{\mathrm{mis}}$ which is necessarily all zero. Furthermore, this statement is satisfied when $p(x_j \mid r_j = 0, c, \phi) = p(x_j \mid c, \phi)$ for all x_j such that $r_j = 0$ (y_j is a suppressed value), using naive Bayes's assumption again. The resulting statement says that the missing part of the j-th attribute must follow the same distribution as the one over all j-th attribute values of original tuples. Since the observed part and the missing part are mutually exclusive and collectively exhaustive, this statement must also apply to the observed part. Eventually we see that, when the empirical distribution from the original dataset is identical to those from an anonymized dataset, the MAR condition is preserved.

To measure the deviation from the MAR condition, we consider the Kullback-Leibler (KL) divergence, which was firstly introduced by Kifer and Gehrke [11] in the literature of anonymization. The KL divergence is defined and simplified under the independence assumption in naive Bayes:

$$\begin{aligned} \mathrm{KL}(\hat{p}, \hat{q}) &= \sum_{\boldsymbol{x}, c} \hat{p}(\boldsymbol{x}, c) \log \frac{\hat{p}(\boldsymbol{x}, c)}{\hat{q}(\boldsymbol{x}, c)} = \sum_{c} \hat{p}(c) \sum_{j} \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \qquad (7) \\ &= \sum_{c} \hat{p}(c) \sum_{j} \mathrm{KL}_{j,c}(\hat{p}, \hat{q}) \ \text{ where } \ \mathrm{KL}_{j,c}(\hat{p}, \hat{q}) = \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \end{aligned}$$

(the derivation is presented in the appendix). Here $\hat{p}$ is the empirical probability distribution from the original dataset, and $\hat{q}$ is the one from an anonymized dataset, which may be unfinished one in the ANONYMIZE procedure (Section 2.3). $\mathrm{KL}_{j,c}(\hat{p}, \hat{q})$ is the class- and attribute-wise version of the KL divergence. It is known that the KL divergence is non-negative, and hence making $\mathrm{KL}(\hat{p}, \hat{q})$ smaller implies making each $\mathrm{KL}_{j,c}(\hat{p}, \hat{q})$ smaller. This further implies making $\hat{p}(x_j \mid c)$ and $\hat{q}(x_j \mid c)$ closer, which leads to the preservation of the MAR condition. In addition, the summation in Eq. 7 is taken over all classes and distinct attribute values and so is costly to compute. Next, we plug the KL divergence above into the ANONYMIZE procedure as a new cost function which is efficiently computed.

3.3 Cell-suppression Cost for Preserving the MAR Condition

To introduce a light-weight cost function that reflects the MAR condition, we consider the difference between two KL divergences before and after a cell suppression in the ANONYMIZE procedure. Cell suppression is a local operator, so the difference between these two quantities will also be rather limited.

More formally, let $\mathcal{D}^{(\ell)}$ be the dataset obtained at the end of the ℓ-th loop in the ANONYMIZE procedure. We apply a cell suppression once to the dataset at each loop. The difference is then written as $\Delta\mathrm{KL} = \mathrm{KL}(\hat{p}, \hat{q}') - \mathrm{KL}(\hat{p}, \hat{q})$, where $\hat{p}$ is the empirical distribution from the original dataset $\mathcal{D} = \mathcal{D}^{(0)}$, $\hat{q}$ is the one from $\mathcal{D}^{(\ell)}$, and $\hat{q}'$ is the one from $\mathcal{D}^{(\ell+1)}$ ($\ell \geq 0$). Here we easily have:

$$\Delta\mathrm{KL} = \sum_c \hat{p}(c) \sum_j \Delta\mathrm{KL}_{j,c} \text{ where } \Delta\mathrm{KL}_{j,c} = \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{q}(x_j \mid c)}{\hat{q}'(x_j \mid c)}. \quad (8)$$

Let us consider next a more specific case in which the j-th non-null attribute value x_j of a tuple $t = (\boldsymbol{y}, c)$ is suppressed in $\mathcal{D}^{(\ell)}$. Also suppose that $\hat{q}(x_j \mid c)$ has been obtained from $\mathcal{D}^{(\ell)}$ as $(N(x_j, c)+\alpha)/(N(\neg\perp_j, c)+\alpha|\mathcal{V}_j|)$. Then, $\hat{q}'(x_j \mid c)$ is obtained from $\mathcal{D}^{(\ell+1)}$ as $(N(x_j, c) - N(\boldsymbol{y}, c) + \alpha)/(N(\neg\perp_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|)$, in which the count of the suppressed non-null value is decreased by $N(\boldsymbol{y}, c)$. Substituting these empirical probabilities into Eq. 8 results in:

$$\Delta\mathrm{KL}_{j,c} = \hat{p}(x_j \mid c) \log \frac{N(x_j, c) + \alpha}{N(x_j, c) - N(\boldsymbol{y}, c) + \alpha} + \log \frac{N(\neg\perp_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(\neg\perp_j, c) + \alpha|\mathcal{V}_j|} \quad (9)$$

(the derivation is presented in the appendix).

Based on the above, consider an extended case where two incomplete tuples $t = (y_1, y_2, \ldots, y_M, c)$ and $t' = (y'_1, y'_2, \ldots, y'_M, c)$ are suppressed and merged. Suppressions occur at y_j and/or y'_j in the j-th attribute such that y_j and y'_j are distinct. An extension of Eq. 9 to this case is derived as:

$$\begin{aligned}\Delta\mathrm{KL}_{j,c} = \hat{p}(y_j \mid c) \log \frac{N(y_j, c) + \alpha}{N(y_j, c) - w_j(t) + \alpha} + \hat{p}(y'_j \mid c) \log \frac{N(y'_j, c) + \alpha}{N(y'_j, c) - w_j(t') + \alpha} \\ + \log \frac{N(\neg\perp_j, c) - (w_j(t) + w_j(t')) + \alpha|\mathcal{V}_j|}{N(\neg\perp_j, c) + \alpha|\mathcal{V}_j|}. \quad (10)\end{aligned}$$

Note here that $\hat{p}(\perp_j \mid c) = 1$ and we define $w_j(t) = N(\boldsymbol{y}, c)$ (if $y_j \neq \perp_j$) or $w_j(t) = 0$ (if $y_j = \perp_j$) for an incomplete tuple $t = (\boldsymbol{y}, c)$. Finally, a cost function Γ_{mar} which measures the deviation from the MAR condition is introduced as:

$$\Gamma_{\mathrm{mar}}(t, t', \hat{p}, \mathcal{N}) \stackrel{\mathrm{def}}{=} \Delta\mathrm{KL} = \textstyle\sum_c \hat{p}(c) \sum_j \Delta\mathrm{KL}_{j,c}, \quad (11)$$

where $\Delta\mathrm{KL}_{j,c}$ is the one defined in Eq. 10. One may find from Eq. 10 that we have only to refer to the quantities related to the suppressed attribute values and hence computing $\Delta\mathrm{KL}_{j,c}$ just requires a constant time. Lastly, we see from Eqs. 8 and 11 that Γ_{mar} can be negative. This case happens when the empirical distribution $\hat{q}'$ after the suppression gets closer than $\hat{q}$ to the original one $\hat{p}$.

4 Experimental Results

We tested the proposed method using the adult dataset available from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets/`

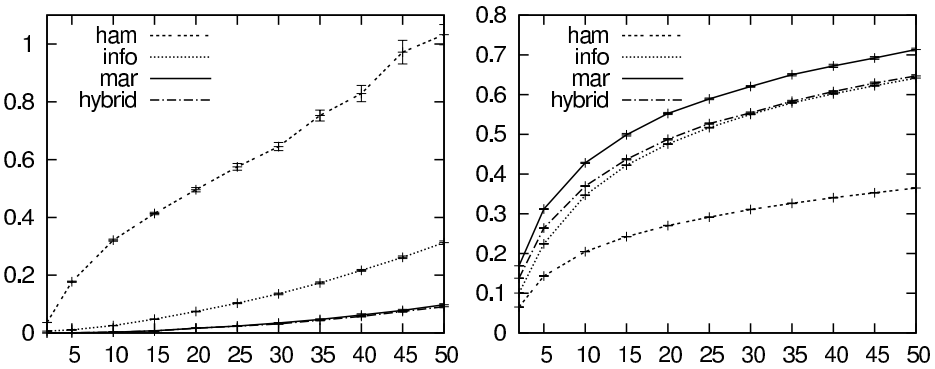

Fig. 1. The KL divergence (left) and the ratio of suppressed cells (right) in k-anonymized datasets, with various k (indicated by the x-axis) and cost functions. In the case of the KL divergence, lines mar and hybrid overlap almost entirely.

`Adult`). Specifically, we compare cost functions Γ_{ham}, Γ_{info} and Γ_{mar} plugged into the ANONYMIZE procedure in Section 2.3. We additionally introduced a cost function Γ_{hybrid} as a simple hybrid of Γ_{ham} and Γ_{mar}, defined as follows:

$$\Gamma_{\mathrm{hybrid}}(t,t',\hat{p},\mathcal{N}) \stackrel{\mathrm{def}}{=} \begin{cases} \Gamma_{\mathrm{mar}}(t,t',\hat{p},\mathcal{N})/\Gamma_{\mathrm{ham}}(t,t',\hat{p},\mathcal{N}) & (\Gamma_{\mathrm{mar}}(t,t',\hat{p},\mathcal{N}) \leq 0) \\ \Gamma_{\mathrm{mar}}(t,t',\hat{p},\mathcal{N})\Gamma_{\mathrm{ham}}(t,t',\hat{p},\mathcal{N}) & (\Gamma_{\mathrm{mar}}(t,t',\hat{p},\mathcal{N}) > 0). \end{cases} \tag{12}$$

In this hybrid function, Γ_{mar} works as a base cost function, and Γ_{ham} plays a role of a penalty function which increases the cost according to the Hamming distance, i.e. the total number of suppressed cells.

The adult dataset has two classes: salary above or below 50,000 dollars. Furthermore, following the previous work [2, 9, 19], we used eight attributes for a person: *age*, *work class*, *education*, *marital status*, *occupation*, *race*, *gender* and *native country*. All attribute except *age* are discrete, and we discretized the *age* attribute as $[15, 20)$, $[20, 25)$, $[25, 30)$, $\ldots$, $[70, 75)$, $[75, 80)$ and $[80, 95)$, where we first split the whole range into the ranges of five years and then merged the last three to ensure that each range includes more than 100 tuples.

The classifiers we used are naive Bayes classifiers and C4.5, implemented in Weka [20]. Each classifier is evaluated by average error rate in stratified 10-fold cross validation. Before evaluation, we first anonymized the original datasets, and in each fold of cross validation, we use the anonymized version for the training dataset and the original version for the test dataset. All classifiers were run under Weka's default setting. Since the ANONYMIZE procedure runs in a randomized way, the obtained results were averaged over 30 trials.

Fig. 1 (left) shows the KL divergence between the empirical distribution from the original dataset and the one from the datasets k-anonymized by the ANONYMIZE procedure with $k = 2, 5, 10, 15, \ldots$ and cost functions. Fig. 1 (right) shows the number of suppressed cells in the k-anonymized datasets. In all graphs presented in the paper, the lines labeled ham, info, mar and hybrid correspond to

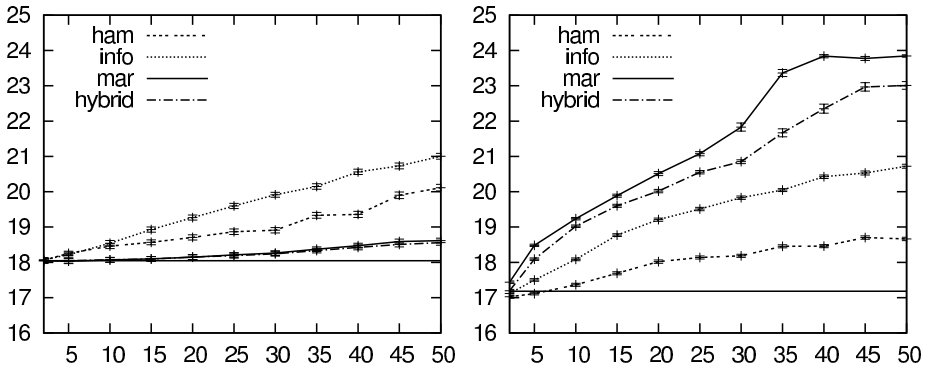

Fig. 2. The average error rate (%) of naive Bayes (left) and C4.5 (right) for k-anonymized datasets, with various k (the x-axis) and cost functions. In the case of naive Bayes, lines mar and hybrid overlap almost entirely.

the cases with Γ_{ham}, Γ_{info}, Γ_{mar} and Γ_{hybrid}, respectively. It is found in Fig. 1 (left) that, as expected, the KL divergence is smaller with Γ_{mar} and Γ_{hybrid}. Fig. 1 (right), on the other hand, exhibits a contrasting behavior that Γ_{ham} yields a smaller number of suppressed cells, which is also expected. In addition, the error bars in the graphs indicate the 95% confidence intervals. The error bars are narrow, so we can see that the ANONYMIZE procedure works stably.

Fig. 2 shows the average error rate (%) of naive Bayes (left) and C4.5 (right) for k-anonymized datasets. The horizontal line indicates the average error rate for the original dataset. In these graphs, Γ_{mar} and Γ_{hybrid} give the least degradation of error rate when combined with naive Bayes, as the theory suggests. However, C4.5 did not work well with Γ_{mar}. From the fact that Γ_{hybrid} which brings less suppressions reduces error rate, the number of suppressed cells seems to give a highly negative impact on the classification performance of C4.5.

5 Concluding Remarks

This paper proposed a cell-suppression based k-anonymization method which keeps minimal the loss of utility. The proposed method aims to preserve the MAR condition and uses the KL divergence as a utility measure. From the discussions and the experimental results presented in this paper, our approach is shown to be statistically promising in both formal and practical senses. On the other hand, there remain a couple of open problems. Here we conclude the paper by discussing such open problems and related work in the literature.

First, a newly introduced cost function Γ_{mar}, which is based on the KL divergence and the independence assumption in naive Bayes, only considers attributes individually. This also applies to most of the existing work, e.g. Γ_{info} used in [7], but some authors take multi-dimensional approaches, in which two or more attributes are jointly taken into account. For instance, given a classification dataset, kACTUS [12] performs cell suppression based on a decision tree

built in advance. Relaxing the independence assumption in naive Bayes would be one possible extension of the proposed method.

There have been several methods targeting classification datasets. Many of such methods [2, 6, 9, 19], as well as kACTUS above, exploit classification-centric heuristic scores such as information gain. Although out target is not limited to classification datasets, as a simple hybrid cost Γ_{hybrid} used in our experiment suggests, some classification-centric cost function would contribute to the improvement of classification performance. In addition, anonymization may be performed in big data environments consisting of, for example, data providers, data collectors and data users who have different requirements [22]. To balance several cost functions, multi-objective optimization techniques look attractive. Dewri et al. [3] explored an evolutionary multi-objective optimization to determine a suitable anonymity threshold k.

As mentioned earlier, one advantage of cell suppression is that it requires no hierarchical knowledge. If such knowledge is available, the coarsening-at-random (CAR) condition [8] would be a key notion since it is a generalization of the MAR condition considering partial information loss in each cell. In addition, Harada et al. proposed a way for automatically constructing hierarchical knowledge [7].

References

1. Aggarwal, C.C.: Data Mining: The Textbook. Springer (2015)
2. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: Proc. of ICDE-05. pp. 217–228 (2005)
3. Dewri, R., Ray, I., Ray, I., Whitley, D.: On the optimal selection of k in the k-anonymity problem. In: Proc. of ICDE-08. pp. 1364–1366 (2008)
4. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
5. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: a survey of recent developments. ACM Computing Surveys 42(4), 14:1–14:53 (2010)
6. Fung, B.C.M., Wang, K., Yu, P.S.: Anonymizing classification data for privacy preservation. IEEE Trans. on Knowledge and Data Engineering 19(5), 711–725 (2007)
7. Harada, K., Sato, Y., Togashi, Y.: Reducing amount of information loss in k-anonymization for secondary use of collected personal information. In: Proc. of the 2012 Service Research and Innovation Institute Global Conf. pp. 61–69 (2012)
8. Heitjan, D.F.: Ignorability and coarse data. The Annals of Statistics 19(4), 2244–2253 (1991)
9. Iyengar, V.: Transforming data to satisfy privacy constraints. In: Proc. of KDD-02. pp. 279–288 (2002)
10. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
11. Kifer, D., Gehrke, J.: Injecting utility into anonymized datasets. In: Proc. of SIGMOD-06. pp. 217–228 (2006)
12. Kisilevich, S., Rokach, L., Elovici, Y.: Efficient multidimensional suppresion for k-anonymity. IEEE Trans. on Knowledge and Data Engineering 22(3), 334–347 (2010)

13. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: Proc. of SIGMOD-05. pp. 49–60 (2005)
14. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: Proc. of PODS-04. pp. 223–228 (2004)
15. Rubin, D.B.: Inference and missing data. Biometrika 63, 581–592 (1976)
16. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Trans. on Knowledge and Data Engineering 13(6), 670–682 (2001)
17. Schafer, J.L., Graham, J.W.: Missing data: our view of the state of the art. Psychological Methods 7, 147–177 (2002)
18. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. Int'l J. of Uncertainty, Fuzziness and Knowledge-based Systems 10(5), 571–588 (2002)
19. Wang, K., Yu, P.S., Chakraborty, S.: Bottom-up generalization: a data mining solution to privacy protection. In: Proc. of ICDM-04. pp. 249–256 (2004)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2nd edn. (2005)
21. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.C.: Utility-based anonymization using local recoding. In: Proc. of KDD-06. pp. 785–790 (2006)
22. Xu, L., Jiang, C., Chen, Y., Wang, J., Ren, Y.: A framework for categorizing and applying privacy-preservation techniques in big data mining. Computer 49(2), 54–62 (2016)

Appendix: Derivation of the Proposed Suppression Cost

Here, we complete the derivation of the cost function Γ_{mar} by showing how to obtain Eqs. 7 and 9. First, let us note that $\hat{p}(c) = \hat{q}(c)$ holds since the class label c is initially non-null and will be never suppressed. Eq. 7 is then derived as follows:

$$
\begin{aligned}
&\mathrm{KL}(\hat{p}, \hat{q}) \\
&= \sum_{\boldsymbol{x},c} \hat{p}(\boldsymbol{x}, c) \log \frac{\hat{p}(\boldsymbol{x}, c)}{\hat{q}(\boldsymbol{x}, c)} = \sum_{\boldsymbol{x},c} \Big(\hat{p}(c) \prod_{j'=1}^{M} \hat{p}(x_{j'} \mid c) \Big) \log \frac{\hat{p}(c) \prod_{j=1}^{M} \hat{p}(x_j \mid c)}{\hat{q}(c) \prod_{j=1}^{M} \hat{q}(x_j \mid c)} \\
&= \sum_{c} \hat{p}(c) \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} \Big(\prod_{j'=1}^{M} \hat{p}(x_{j'} \mid c) \Big) \sum_{j=1}^{M} \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \\
&= \sum_{c} \hat{p}(c) \sum_{j=1}^{M} \sum_{x_1} \cdots \sum_{x_{j-1}} \sum_{x_j} \sum_{x_{j+1}} \cdots \sum_{x_M} \\
&\qquad \Big(\prod_{j'=1}^{j-1} \hat{p}(x_{j'} \mid c) \Big) \hat{p}(x_j \mid c) \Big(\prod_{j'=j+1}^{M} \hat{p}(x_{j'} \mid c) \Big) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \qquad (13) \\
&= \sum_{c} \hat{p}(c) \sum_{j=1}^{M} \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \cdot \\
&\qquad \sum_{x_1} \cdots \sum_{x_{j-1}} \sum_{x_{j+1}} \cdots \sum_{x_M} \Big(\prod_{j'=1}^{j-1} \hat{p}(x_{j'} \mid c) \Big) \Big(\prod_{j'=j+1}^{M} \hat{p}(x_{j'} \mid c) \Big) \qquad (14)
\end{aligned}
$$

$$= \sum_c \hat{p}(c) \sum_{j=1}^{M} \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)} \Big(\prod_{j'=1}^{j-1} \sum_{x_{j'}} \hat{p}(x_{j'} \mid c) \Big) \Big(\prod_{j'=j+1}^{M} \sum_{x_{j'}} \hat{p}(x_{j'} \mid c) \Big)$$
$$= \sum_c \hat{p}(c) \sum_{j=1}^{M} \sum_{x_j} \hat{p}(x_j \mid c) \log \frac{\hat{p}(x_j \mid c)}{\hat{q}(x_j \mid c)}. \tag{15}$$

In Eqs. 13 and 14, we carefully reordered summations and moved irrelevant factors outside the summations wherever possible. Eq. 15 was finally derived using $\sum_{x_{j'}} \hat{p}(x_{j'} \mid c) = 1$ since $\hat{p}$ is a probability function.

On the other hand, for Eq. 9, we have been considering a specific case where the j-th non-null attribute value x_j of a tuple $t = (\boldsymbol{y}, c)$ is suppressed. We have $\hat{q}(x_j \mid c) = (N(x_j, c) + \alpha)/(N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|)$ and $\hat{q}'(x_j \mid c) = (N(x_j, c) - N(\boldsymbol{y}, c) + \alpha)/(N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|)$ as already mentioned, and additionally, for each value x'_j of j-th attribute which is not suppressed this time (i.e. $x'_j \neq x_j$), we have $\hat{q}'(x'_j \mid c) = (N(x'_j, c) + \alpha)/(N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|)$. Substituting these probabilities into Eq. 8 results in Eq. 9 as follows:

$$\begin{aligned}
&\Delta\mathrm{KL}_{j,c} \\
&= \hat{p}(x_j \mid c) \log \frac{\hat{q}(x_j \mid c)}{\hat{q}'(x_j \mid c)} + \sum_{x'_j : x'_j \neq x_j} \hat{p}(x'_j \mid c) \log \frac{\hat{q}(x'_j \mid c)}{\hat{q}'(x'_j \mid c)} \\
&= \hat{p}(x_j \mid c) \log \left(\frac{N(x_j, c) + \alpha}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|} \cdot \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(x_j, c) - N(\boldsymbol{y}, c) + \alpha} \right) + \\
&\qquad \sum_{x'_j : x'_j \neq x_j} \hat{p}(x'_j \mid c) \log \left(\frac{N(x'_j, c) + \alpha}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|} \cdot \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(x'_j, c) + \alpha} \right) \\
&= \hat{p}(x_j \mid c) \log \frac{N(x_j, c) + \alpha}{N(x_j, c) - N(\boldsymbol{y}, c) + \alpha} + \hat{p}(x_j \mid c) \log \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|} + \\
&\qquad \sum_{x'_j : x'_j \neq x_j} \hat{p}(x'_j \mid c) \log \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|} \\
&= \hat{p}(x_j \mid c) \log \frac{N(x_j, c) + \alpha}{N(x_j, c) - N(\boldsymbol{y}, c) + \alpha} + \\
&\qquad \log \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|} \left(\hat{p}(x_j \mid c) + \sum_{x'_j : x'_j \neq x_j} \hat{p}(x'_j \mid c) \right) \\
&= \hat{p}(x_j \mid c) \log \frac{N(x_j, c) + \alpha}{N(x_j, c) - N(\boldsymbol{y}, c) + \alpha} + \log \frac{N(\neg\bot_j, c) - N(\boldsymbol{y}, c) + \alpha|\mathcal{V}_j|}{N(\neg\bot_j, c) + \alpha|\mathcal{V}_j|}.
\end{aligned}$$