Automatic Discovery of the Statistical Types of Variables in a Dataset

Isabel Valera [1] **Zoubin Ghahramani** [1 2]

Abstract

A common practice in statistics and machine learning is to assume that the statistical data types (*e.g.*, ordinal, categorical or real-valued) of variables, and usually also the likelihood model, is known. However, as the availability of real-world data increases, this assumption becomes too restrictive. Data are often heterogeneous, complex, and improperly or incompletely documented. Surprisingly, despite their practical importance, there is still a lack of tools to automatically discover the statistical types of, as well as appropriate likelihood (noise) models for, the variables in a dataset. In this paper, we fill this gap by proposing a Bayesian method, which accurately discovers the statistical data types in both synthetic and real data.

1. Introduction

Data analysis problems often involve pre-processing *raw* data, which is a tedious and time-demanding task due to several reasons: i) raw data is often unstructured and large-scale; ii) it contains errors and missing values; and iii) documentation may be incomplete or not available. As a consequence, as the availability of data increases, so does the interest of the data science community to automate this process. In particular, there are a growing body of work which focuses on automating the different stages of data pre-processing, including data cleaning (Hellerstein, 2008), data wrangling (Kandel et al., 2011) and data integration and fusion (Dong & Srivastava, 2013).

The outcome of data pre-processing is commonly a structured dataset, in which the *objects* are described by a set of *attributes*. However, before being able to proceed with the predictive analytics step of the data analysis process, the data scientist often needs to identify which kind of variables (*i.e.*, real-values, categorical, ordinal, etc.) these attributes represent. This labeling of the data is necessary to select the appropriate machine learning approach to explore, find patterns or make predictions on the data. As an example, a prediction task is solved differently depending on the kind of data to be predicted—*e.g.*, while prediction on categorical variables is usually formulated as a classification task, in the case of ordinal variables it is formulated as an ordinal regression problem (Agresti, 2010). Moreover, different data types should be pre-processed and input differently in the predictive tool—*e.g.*, categorical inputs are often transformed into as many binary inputs (which state whether the object belongs to a category or not) as number of categories; positive real inputs might be log-transformed, etc.

Information on the statistical data types in a dataset becomes particularly important in the context of statistical machine learning (Breiman, 2001), where the choice of a likelihood model appears as a main assumption. Although extensive work has focused on model selection (Ando, 2010; Burnham & Anderson, 2003), the likelihood model is usually assumed to be known and fixed. As an example, a common approach is to model continuous data as Gaussian variables, and discrete data as categorical variables. However, while extensive work has shown the advantages of capturing the statistical properties of the observed data in the likelihood model (Chu & Ghahramani, 2005a; Schmidt et al., 2009; Hilbe, 2011; Valera & Ghahramani, 2014), there still exists a lack of tools to automatically perform likelihood model selection, or equivalently to discover the most plausible statistical type of the variables in the data, directly from the data.

In this work, we aim to fill this gap by proposing a general and scalable Bayesian method to solve this task. The proposed method exploits the latent structure in the data to automatically distinguish among real-valued, positive real-valued and interval data as types of continuous variables, and among categorical, ordinal and count data as types of discrete variables. The proposed method is based on probabilistic modeling and exploits the following key ideas:

i) There exists a latent structure in the data that capture the statistical dependencies among the different objects and attributes in the dataset. Here, as in standard latent feature modeling, we assume that we can capture this structure by a low-rank representation, such that conditioning on it, the likelihood model factorizes for both number of objects and attributes.
ii) The observation model for each attribute can be ex-

[1] University of Cambridge, Cambridge, United Kingdom; [2] Uber AI Labs, San Francisco, California, USA. Correspondence to: Isabel Valera <miv24@cam.ac.uk>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017. JMLR: W&CP. Copyright 2017 by the author(s).

pressed as a mixture of likelihood models, one per each considered data type, where the inferred weight associated to a likelihood model captures the probability of the attribute belonging to the corresponding data type.

We derive an efficient MCMC inference algorithm to jointly infer both the low-rank representation and the weight of each likelihood model for each attribute in the observed data. Our experimental results show that the proposed method accurately discovers the true data type of the variables in a dataset, and by doing so, it fits the data substantially better than modeling continuous data as Gaussian variables and discrete data as categorical variables.

2. Problem Statement

As stated above, the outcome of the pre-processing step of data analysis is a structured dataset, in which a set of objects are defined by a set of attributes, and our objective is to automatically discover which type of variables these attributes correspond to. In order to distinguish between discrete and continuous variables, we can apply simple logic rules, *e.g.*. count the number of unique values that the attribute takes and how many times we observe these attributes. Moreover, binary variables are invariant to the labeling of the categories, and therefore, both categorical and ordinal models are equivalent in this case. However, distinguishing among different types of discrete and continuous variables cannot be easily solved using simple heuristics.

In the context of continuous variables, given the finite size of observed datasets, it is complicated to identify whether a variable may take values in the entire real line, or only on an interval of it, *e.g.*, $(0, \infty)$ or (θ_L, θ_H). In other words, due to the finite observation sample, we cannot distinguish whether the data distribution has an infinite tail that we have not observed, or its support is limited to an interval. As an illustrative example, Figures 2(d)&(f) in Section 4 show two data distributions that, although at a first sight look similar, correspond respectively to a Beta variable, which therefore takes values in the interval $(0, 1)$, and a gamma variable, which takes values in $(0, \infty)$.

In the context of discrete data, it is impossible to tell the difference between categorical and ordinal variables in isolation. The presence of an order in the data only makes sense given a context. As an example, while colors in M&Ms usually do not present an order, colors in a traffic light clearly do. Similarly, we cannot easily distinguish between ordinal data (which take values in a finite ordered set) and count data (which take values in an infinite ordered set with equidistant values) due to two main reasons. First, similarly to continuous variables, since datasets contain a finite number of examples, it is difficult to tell whether we have observed the finite set of possible values of a variable, or simply a finite subsample of an infinite set. Second, we would need access to exact information on whether its consecutive values are equidistant or not, however, this information depends on how the data have been gathered. For example, an attribute that collects information on “frequency of an action” will correspond to an ordinal variable if its categories belong to, *e.g.*, {“never”, “sometimes”, “usually”, “often”}, and to a count variable if it takes values in {‘‘0 times per week”, “1 time per week”, . . . }.

Previous work (Hernandez-Lobato et al., 2014) proposed to distinguish between categorical and ordinal data by comparing the model evidence and the predictive test log-likelihood of ordinal and categorical models. However, this approach can be only used to distinguish between ordinal and categorical data, and it does so by assuming that it has access to a real-valued variable that contains information about the presence of an ordering in the observed discrete (ordinal or categorical) variable. As a consequence, it cannot be easily generalizable to label the data type of all the variables (or attributes) in a dataset. In contrast, in this paper we proposed a general method that allows us to distinguish among real-valued, positive real-valued and interval data as types of continuous variables, and among categorical, ordinal and count data as types of discrete variables. Moreover, the general framework we present can be readily extended to other data types as needed.

3. Methodology

In this section, we introduce a Bayesian method to determine the statistical type of variable that corresponds to each of the attributes describing the objects in an observation matrix $\mathbf{X}$. In particular, we propose a probabilistic model, in which we assume that there exists a low-rank representation of the data that captures its latent structure, and therefore, the statistical dependencies among its objects and attributes. In detail, we consider that each observation x_n^d can be explained by a K-length vector of latent variables $\mathbf{z}_n = [z_{n1}, \ldots, z_{nK}]$ associated to the n-th object and a weighting vector $\boldsymbol{b}^d = [b_1^d, \ldots, b_K^d]$ (with K being the number of latent variables), whose elements b_k^d weight the contribution of k-th the latent feature to the d-th attribute in $\mathbf{X}$. Then, given the latent low-rank representation of the data, the attributes describing the objects in a dataset are assumed to be independent, *i.e.*,

$$p(\mathbf{X}|\mathbf{Z}, \{\boldsymbol{b}^d\}_{d=1}^D) = \prod_{d=1}^D p(\mathbf{x}^d|\mathbf{Z}, \boldsymbol{b}^d),$$

where we gather the latent feature vectors $\mathbf{z}_n$ in a $N \times K$ matrix $\mathbf{Z}$. For convenience, here $\mathbf{z}_n$ is a K-length row vector, while $\boldsymbol{b}^d$ is a K-length column vector. The above model resembles standard latent feature models (Salakhutdinov & Mnih, 2007; Griffiths & Ghahramani, 2011), which assume known and fixed likelihood models $p(\mathbf{x}^d|\mathbf{Z}, \boldsymbol{b}^d)$. In contrast, in this paper we aim to *infer* the statistical data type (or equivalently, the likelihood model) that better captures the distribution of each attribute in $\mathbf{X}$. To this end, here we assume that the likelihood model of the d-th attribute in $\mathbf{X}$ is a mixture of likelihood functions such that

$$p(\mathbf{x}^d|\mathbf{Z}, \{\boldsymbol{b}_\ell^d\}_{\ell\in\mathcal{L}^d}) = \sum_{\ell\in\mathcal{L}^d} w_\ell^d \, p_\ell(\mathbf{x}^d|\mathbf{Z}, \boldsymbol{b}_\ell^d),$$

where $\mathcal{L}^d$ is the set of possible types of variables (or equivalently, likelihood models) to be considered for this attribute, and the weight w_ℓ^d captures the probability of the likelihood function ℓ in the d-th attribute of the observation matrix $\mathbf{X}$. Note that, the above expression is a valid likelihood model as long as $\sum_{\ell\in\mathcal{L}^d} w_\ell^d = 1$ and each $p_\ell(\mathbf{x}^d|\mathbf{Z}, \boldsymbol{b}_\ell^d, \Psi_\ell^d)$ is a normalized probability density function or probability mass function for, respectively, continuous and discrete variables. Hence, under the proposed model, which is is illustrated in Figure 1a, the likelihood factorizes as

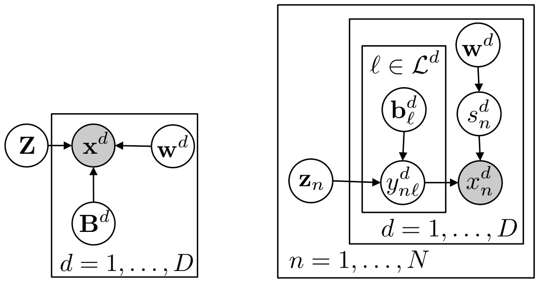
$$p(\mathbf{X}|\mathbf{Z}, \{\boldsymbol{b}_\ell^d\}) = \prod_{d=1}^{D} \sum_{\ell\in\mathcal{L}^d} w_\ell^d \, p_\ell(\mathbf{x}^d|\mathbf{Z}, \boldsymbol{b}_\ell^d). \tag{1}$$

We place a Dirichlet prior distribution on the likelihood weights $\mathbf{w}^d = [w_\ell^d]_{\ell\in\mathcal{L}^d}$, and similarly to (Salakhutdinov & Mnih, 2007), assume that both the latent feature vectors $\mathbf{z}_n$ and the weighting vectors $\boldsymbol{b}_j^d$ are Gaussian distributed with zero mean and covariance matrices $\sigma_z^2\mathbf{I}$ and $\sigma_b^2\mathbf{I}$, respectively. Here, $\mathbf{I}$ denotes the identity matrix of size equal to the number of latent features K.

Moreover, we consider the following types of data for, respectively, continuous and discrete variables:

- Continuous variables:
  1. Real-valued data, which takes values in the real line, *i.e.*, $x_n^d \in \Re$.
  2. Positive real-valued data, which takes values in the positive real line, *i.e.*, $x_n^d \in \Re^+$.
  3. Interval data, which takes values in an interval of the real line, *i.e.*, $x_n^d \in (\theta_L, \theta_H)$, where $\theta_L, \theta_H \in \Re$ and $\theta_L \leq \theta_H$.
- Discrete variables:
  1. Categorical data, which takes values in a finite unordered set, *e.g.*, $x_n^d \in \{$'blue', 'red', 'black'$\}$.
  2. Ordinal data, which takes values in a finite ordered set, *e.g.*, $x_n^d \in \{$'never', 'sometimes', 'often', 'usually', 'always'$\}$.
  3. Count data, which takes values in the natural numbers, *i.e.*, $x_n^d \in \{0, \ldots, \infty\}$.

We remark that the main goal of this paper is to determine the types of variables that better capture each attribute in the observed matrix $\mathbf{X}$, which in our method translates to inferring the likelihood weights $\mathbf{w}^d$. However, solving this inference problem in an efficient way is a challenging task for several reasons. First, we need to jointly infer all the latent variables in the model, *i.e.*, the low-rank representation of the data (which includes the latent feature matrix $\mathbf{Z}$ and the corresponding weighting vectors $\{\boldsymbol{b}_\ell^d\}_{\{\ell\in\mathcal{L}_d|d=1,\ldots,D\}}$) and the likelihood weights $\{\mathbf{w}^d\}_{d=1}^D$. Second, we need to do so given a heterogeneous (and non-conjugate) observation model, which combines D different likelihood models, corresponding each of them to a mixture of likelihood functions and coupled through the latent feature matrix $\mathbf{Z}$. Additionally, these likelihood functions do not only correspond to either a probability density function or a probability mass function depending on whether we are dealing with a continuous or a discrete variable, but also each mixture combines likelihood functions with different supports. For example, while real-valued data lead to a likelihood function with the real line as support, interval data only accounts for a segment of the real line. Similarly, both categorical and ordinal data assume a finite support, while count data requires an infinite-support likelihood function.



(a) Proposed model (b) Alternative representation

Figure 1. Model illustration.

In order to allow for efficient inference, we exploit the key idea in (Valera & Ghahramani, 2014) to propose an alternative and equivalent model representation (shown in Figure 1b), which efficiently deal with heterogeneous likelihood functions. In this alternative model representation, we include for each observation x_n^d as many Gaussian variables (or *pseudo-observations*) $y_{n\ell}^d \sim \mathcal{N}(\mathbf{z}_n\boldsymbol{b}_\ell^d, \sigma_y^2)$ as the number of likelihood functions in $\mathcal{L}^d$, and assume that there exists a transformation function over the variables $y_{n\ell}^d$ which maps the real line $\Re$ into the support of the likelihood function ℓ, Ω_ℓ, *i.e.*,

$$\begin{array}{cccc} f_\ell : & \Re & \mapsto & \Omega_\ell \\ & y & \rightarrow & x \end{array}. \tag{2}$$

Note that, if we condition on the pseudo-observations the latent variable model behaves as a conjugate Gaussian model, allowing for efficient inference of the latent feature matrix $\mathbf{Z}$ and the weighting vectors $\{\boldsymbol{b}_\ell^d\}$. Additionally, we include a latent multinomial variable $s_n^d \sim Multinomial(\mathbf{w}^d)$ which indicates the type of variable (or likelihood function) that the observation x_n^d belongs to. Then, given s_n^d, we can obtain the observation x_n^d as

$$x_n^d = f_{s_n^d}(y_{ns_n^d}^d + u_n^d), \tag{3}$$

where $u_n^d \sim \mathcal{N}(0, \sigma_u^2)$ is a noise variable. We gather the likelihood assignments s_n^d in a $N \times D$ matrix $\mathbf{S}$.

3.1. Likelihood functions

In this section, we provide the set of transformations to map from the Gaussian pseudo-observations $y^d_{n\ell}$ into the types of data defined above, specifying also the six likelihood functions that our method will account for.

3.1.1. Continuous variables

In the case of continuous variables, we assume that the mapping functions f_ℓ are continuous invertible and differentiable functions, such that we can obtain corresponding likelihood function (after integrating out the pseudo-observation $y^d_{n\ell}$) as

$$p_\ell(x^d_n|\mathbf{z}_n, \boldsymbol{b}^d_\ell, s^d_n = \ell) = \frac{1}{\sqrt{2\pi(\sigma^2_y + \sigma^2_u)}} \left| \frac{d}{dx^d_n} f^{-1}_\ell(x^d_n) \right| \times \exp\left\{ -\frac{1}{2(\sigma^2_y + \sigma^2_u)} (f^{-1}_\ell(x^d_n) - \mathbf{z}_n \boldsymbol{b}^d_\ell)^2 \right\},$$

where f^{-1}_ℓ is the inverse function of the transformation $f_\ell(\cdot)$, *i.e.*, $f^{-1}_\ell(f_\ell(v)) = v$. Next, we provide examples of mapping functions that allow us to account for real-valued, positive real-valued, and interval data.

1. Real-valued Data. In order to obtain real-valued observations, *i.e.*, $x^d_n \in \Re$, we need a transformation over y^d_n that maps from the real numbers to the real numbers, *i.e.*, $f_\Re : \Re \rightarrow \Re$. The simplest case is to assume that $x = f_\Re(y + u) = y + u$, and therefore, each observation is distributed as $x^d_n \sim \mathcal{N}(\mathbf{z}_n \boldsymbol{b}^d_\Re, \sigma^2_y + \sigma^2_u)$. Nevertheless, other mapping functions can be used, *e.g.*, we will use in our experiments the transformation

$$x = f_\Re(y + u) = w(y + u) + \mu,$$

where w and μ are parameters allowing attribute rescaling, and tuneable by the user.

2. Positive Real-valued Data. As an example of a function that maps from the real numbers to the positive real numbers, *i.e.*, $f_{\Re_+} : \Re \mapsto \Re_+$, we consider

$$x = f_{\Re_+}(y + u) = \log(1 + \exp(w(y + u))).$$

where w allows attribute rescaling.

3. Interval Data. As an example of a function the maps from the real numbers into the interval (θ_L, θ_H), *i.e.*, $f_{\Re_+} : \Re \mapsto (\theta_L, \theta_H)$, we consider the transformation

$$x = f_{\text{Int}}(y + u) = \frac{\theta_H - \theta_L}{1 + \exp(-w(y + u))} + \theta_L,$$

where w, θ_L and θ_H are user hyperparameters.[1]

3.1.2. Discrete variables

1. Categorical Data. Now we account for categorical observations, i.e., each observation x^d_n can take values in the unordered index set $\{1, \ldots, R_d\}$. Hence, assuming a multinomial probit model, we can write

$$x = f_{\text{cat}}(\mathbf{y}) = \arg \max_{r \in \{1, \ldots, R_d\}} y(r),$$

where in this case there are as many pseudo-observations as number of categories and each pseudo-observation can be sampled as $y^d_{n\text{cat}}(r) \sim \mathcal{N}(\mathbf{z}_n \mathbf{b}^d_{\text{cat}}(r), \sigma^2_y)$ where $\mathbf{b}^d_{\text{cat}}(r)$ denotes the K-length weighting vector, which weights the influence the latent features for a categorical observation x^d_n taking value r. Note that, under this likelihood model, we need one pseudo-observation $y^d_{n\text{cat}}(r)$ and a weighting vector $\boldsymbol{b}^d_{\text{cat}}(r)$ for each possible value of the observation $r \in \{1, \ldots, R_d\}$.

Under the multinomial probit model, we can obtain the probability of x^d_n taking value $r \in \{1, \ldots, R_d\}$ as (Girolami & Rogers, 2005)

$$p_{\text{cat}}(x = r|\mathbf{z}_n, \boldsymbol{b}^d_{\text{cat}}, s^d_n = \text{cat}) = \mathbb{E}_{p(u)}\left[\prod_{\substack{r'=1 \\ r' \neq r}}^{R_d} \Phi\Big(u + \mathbf{z}_n(\mathbf{b}^d_{\text{cat}}(r) - \mathbf{b}^d_{\text{cat}}(r'))\Big) \right],$$

where $\Phi(\cdot)$ denotes the cumulative density function of the standard normal distribution and $\mathbb{E}_{p(u)}[\cdot]$ denotes expectation with respect to the distribution $p(u) = \mathcal{N}(0, \sigma^2_y)$.

2. Ordinal Data. Consider ordinal data, in which each element x^d_n takes values in the ordered index set $\{1, \ldots, R_d\}$. Then, assuming an ordered probit model, we can write

$$x^d_n = f_{\text{ord}}(y^d_{\text{nord}}) = \begin{cases} 1 & \text{if } y^d_{\text{nord}} \leq \theta^d_1 \\ 2 & \text{if } \theta^d_1 < y^d_{\text{nord}} \leq \theta^d_2 \\ & \vdots \\ R_d & \text{if } \theta^d_{R_d - 1} < y^d_{\text{nord}} \end{cases}$$

where again y^d_{nord} is Gaussian distributed with mean $\mathbf{z}_n \boldsymbol{b}^d_{\text{ord}}$ and variance σ^2_y, and θ^d_r for $r \in \{1, \ldots, R_d - 1\}$ are the thresholds that divide the real line into R_d regions. We assume the thresholds θ^d_r are sequentially generated from the truncated Gaussian distribution $\theta^d_r \sim \mathcal{TN}(0, \sigma^2_\theta, \theta^d_{r-1}, \infty)$, where $\theta^d_0 = -\infty$ and $\theta^d_{R_d} = +\infty$. As opposed to the categorical case, now we have a unique weighting vector $\boldsymbol{b}^d_{\text{ord}}$ and a unique Gaussian variable y^d_{nord} for each observation x^d_n, and the value of x^d_n is determined by the region in which y^d_{nord} falls.

Under the ordered probit model (Chu & Ghahramani, 2005b), the probability of each element x^d_n taking value $r \in \{1, \ldots, R_d\}$ can be written as

$$p_{\text{ord}}(x^d_n = r|\mathbf{z}_n, \boldsymbol{b}^d_{\text{ord}}, s^d_n = \text{ord}) = \Phi\left(\frac{\theta^d_r - \mathbf{z}_n \boldsymbol{b}^d_{\text{ord}}}{\sigma_y}\right) - \Phi\left(\frac{\theta^d_{r-1} - \mathbf{z}_n \boldsymbol{b}^d_{\text{ord}}}{\sigma_y}\right).$$

[1] In our experiments, we assume $\theta_L = \arg\min_n(x^d_n) - \epsilon$ and $\theta_H = \arg\max_n(x^d_n) + \epsilon$, where $\epsilon \rightarrow 0$ is a user hyper-parameter. We set the rescaling parameter $w = 2/\max(\boldsymbol{x}^d)$ for the three continuous data types.

3. Count Data. In count data each observation x_n^d takes non-negative integer values, i.e., $x_n^d \in \{0, \ldots, \infty\}$. Then, we assume

$$x_n^d = f_{\text{count}}(y_n^d) = \lfloor g(y_n^d) \rfloor,$$

where $\lfloor v \rfloor$ returns the floor of v, that is the largest integer that does not exceed v, and $g : \Re \to \Re^+$ is a monotonic differentiable function, in our experiments $g(y) = \log(1 + \exp(wy))$. We can thus write the likelihood function as

$$p_{\text{count}}(x_n^d | \mathbf{z}_n, \boldsymbol{b}_{\text{ord}}^d, s_n^d = \text{count}) =$$
$$\Phi\left(\frac{g^{-1}(x_n^d + 1) - \mathbf{z}_n \boldsymbol{b}_{\text{count}}^d}{\sigma_y}\right) - \Phi\left(\frac{g^{-1}(x_n^d) - \mathbf{z}_n \boldsymbol{b}_{\text{count}}^d}{\sigma_y}\right)$$

where $g^{-1} : \Re^+ \to \Re$ is the inverse function of the transformation $g(\cdot)$.

3.2. Inference Algorithm

Here, we exploit the model representation in Figure 1b to derive an efficient inference algorithm that allows us to infer all the latent variables in the model, providing as output the likelihood weights $\mathbf{w}^d$, which determine the probability of the d-th attribute in $\mathbf{X}$ belonging to each of the above data types. Algorithm 1 summarizes the inference.

Sampling low-rank decomposition. In order to sample the latent feature matrix $\mathbf{Z}$ and the associated weighting vectors $\{\boldsymbol{b}_\ell^d\}$, we condition on the pseudo-observations such that we can efficiently sample the feature vectors as $\mathbf{z}_n \sim \mathcal{N}\left(\boldsymbol{\mu}_z^d, \Sigma_z\right)$, where $\Sigma_z = \left(\sum_{d=1}^{d} \sum_{\ell \in \mathcal{L}^d} \boldsymbol{b}_\ell^d (\boldsymbol{b}_\ell^d)^\top + \sigma_z^{-2} \mathbf{I}\right)^{-1}$ and $\boldsymbol{\mu}_z = \Sigma_z \left(\sum_n^N \sum_{\ell \in \mathcal{L}^d} \boldsymbol{b}_\ell^d y_{n\ell}^d\right)$. Note that this step involves a matrix inversion of size K (the number of latent features) per iteration of the algorithm. Similarly, the weighting vectors can be sampled as $\boldsymbol{b}_\ell^d \sim \mathcal{N}\left(\boldsymbol{\mu}_\ell^d, \Sigma_b\right)$, where $\Sigma_b = \left(\sigma_y^{-2} \mathbf{Z}^\top \mathbf{Z} + \sigma_b^{-2} \mathbf{I}\right)^{-1}$ and $\boldsymbol{\mu}_\ell^d = \Sigma_b \left(\sum_n^N \mathbf{z}_n^\top y_{n\ell}^d\right)$. Since Σ_b is shared for all $\{\boldsymbol{b}_\ell^d\}$ with $\ell \in \mathcal{L}^d$ and $d = 1 \ldots, D$, this step also involves one matrix inversions of size K per iteration of the algorithm.

Sampling pseudo-observations. Given the low-rank decomposition and the likelihood assignments $\mathbf{S}$, we can sample each pseudo-observation $y_{n\ell}^d$ from its prior distribution if $s_n^d \neq \ell$, and from its posterior distribution if $s_n^d = \ell$.

In the case of continuous variables, the posterior distribution of the pseudo-observation can be obtained as

$$p(y_{n\ell}^d | x_n^d, \mathbf{z}_n, \boldsymbol{b}_\ell^d, s_n^d = \ell) = \mathcal{N}\left(y_n^d \middle| \hat{\mu}_y, \hat{\sigma}_y^2\right),$$

where $\hat{\mu}_y = \left(\frac{(\mathbf{z}_n \boldsymbol{b}_\ell^d)}{\sigma_y^2} + \frac{f_\ell^{-1}(x_n^d)}{\sigma_u^2}\right) \hat{\sigma}_y^2$, and $\hat{\sigma}_y^2 = \left(\frac{1}{\sigma_y^2} + \frac{1}{\sigma_u^2}\right)^{-1}$.

In the case of discrete variables, the posterior distribution of the pseudo-observation can be computed as follows.

Algorithm 1 Inference Algorithm.

```
Input: X
Initialize: S, $\{\boldsymbol{b}_\ell^d\}$ and $\{y_{n\ell}^d\}$
for each iteration do
    Update Z given $\{\boldsymbol{b}_\ell^d\}$ and $\{y_{n\ell}^d\}$.
    for $d = 1, \ldots, D$ do
        for $\ell \in \mathcal{L}^d$ do
            for $n = 1, \ldots, N$ do
                Sample $\{y_{n\ell}^d\}$ given $x_n^d$, Z, $\{\boldsymbol{b}_\ell^d\}$ and $s_n^d$.
            end for
            Sample $\{\boldsymbol{b}_\ell^d\}$ given Z and $\{y_{n\ell}^d\}$ .
            for $n = 1, \ldots, N$ do
                Sample $s_n^d$ given $x_n^d$, Z and $\{\boldsymbol{b}_\ell^d\}$.
            end for
        end for
        Sample $\mathbf{w}^\mathbf{d}$ given S.
    end for
end for
Output: Likelihood weights $\mathbf{w}^\mathbf{d}$.
```

1. For categorical observations:
$$p(y_{n\text{cat}}^d(r) | x_n^d = T, \mathbf{z}_n, \boldsymbol{b}_{\text{cat}}^d, s_n^d = \text{cat})$$
$$= \begin{cases} \mathcal{TN}(\mathbf{z}_n \boldsymbol{b}_{\text{cat}}^d(r), \sigma_y^2, \max_{j \neq r}(y_{n\text{cat}}^d(j)), \infty), \; r = T \\ \mathcal{TN}(\mathbf{z}_n \boldsymbol{b}_{\text{cat}}^d(r), \sigma_y^2, -\infty, y_{n\text{cat}}^d(T)), \; r \neq T \end{cases}$$

   In words, if $x_n^d = T = r$ we sample y_{nr}^d from a truncated Normal distribution with mean $\mathbf{z}_n \boldsymbol{b}_{\text{cat}}^d(r)$, variance σ_y^2 and truncated on the left by $\max_{j \neq r}(y_{n\text{cat}}^d(j))$. Otherwise, we sample from a truncated Gaussian (with same mean and variance) truncated on the right by $y_{n\text{cat}}^d(r)$ with $r = x_n^d$. Note that sampling from the variables $y_{n\text{cat}}^d(r)$ corresponds to solve a multinomial probit regression problem. Hence, to achieve identifiability we assume, without loss of generality, that the regression function $f_{R_d}(\mathbf{z}_n)$ is identically zero, and thus, we fix $\boldsymbol{b}_\ell^d(R_d) = \mathbf{0}$.

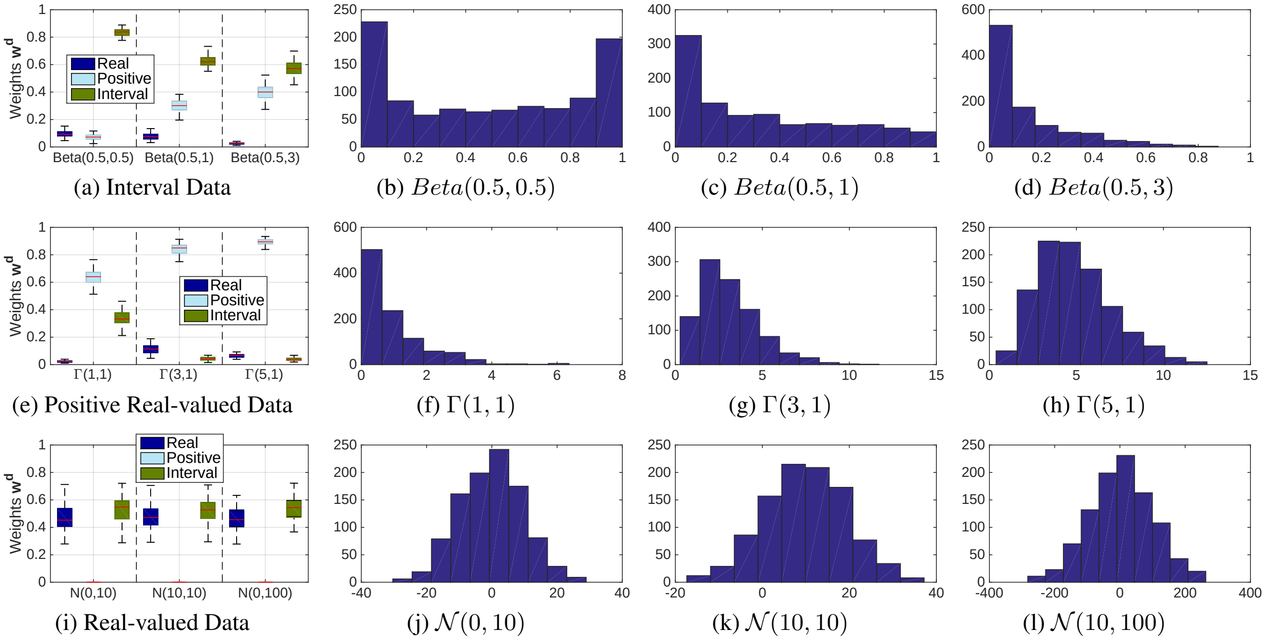
2. For ordinal observations:
$$p(y_{n\text{ord}}^d | x_n^d = r, \mathbf{z}_n, \boldsymbol{b}_{\text{ord}}^d, s_n^d = \text{ord})$$
$$= \mathcal{TN}(y_{n\text{ord}}^d | \mathbf{z}_n \mathbf{b}_{\text{ord}}^d, \sigma_y^2, \theta_{r-1}^d, \theta_r^d).$$

   Note that in this case, we also need to sample the values for the thresholds θ_r^d with $r = 1, \ldots, R_d - 1$ as

$$p(\theta_r^d | y_{n\text{ord}}^d) = \mathcal{TN}(\theta_r^d | 0, \sigma_\theta^2, \theta_{\min}, \theta_{\max}),$$

   where $\theta_{\min} = \max(\theta_{r-1}^d, \max_n(y_{n\text{ord}}^d | x_n^d = r))$ and $\theta_{\max} = \min(\theta_r^d, \min_n(y_{n\text{ord}}^d | x_n^d = r + 1))$. In words, each θ_r^d is constrained to be between θ_{r-1}^d and θ_{r+1}^d, as well as to ensure that the pseudo-observations $y_{n\text{ord}}^d$ associated to the observations $x_n^d = r$ and $x_n^d = r + 1$ fall respectively at the left and at the right side of θ_r^d. Since in this ordinal regression problem the thresholds $\{\theta_r\}_{r=1}^{R_d}$ are unknown, we set θ_1 to a fixed value in order to achieve identifiability.

3. For count observations:
$$p(y_{n\text{count}}^d | x_n^d, \mathbf{z}_n, \boldsymbol{b}_{\text{count}}^d, s_n^d = \text{count})$$
$$= \mathcal{TN}(y_{n\text{count}}^d | \mathbf{z}_n \mathbf{b}_{\text{count}}^d, \sigma_y^2, g^{-1}(x_n^d), g^{-1}(x_n^d + 1)),$$



(a) Interval Data (b) $Beta(0.5, 0.5)$ (c) $Beta(0.5, 1)$ (d) $Beta(0.5, 3)$

(e) Positive Real-valued Data (f) $\Gamma(1, 1)$ (g) $\Gamma(3, 1)$ (h) $\Gamma(5, 1)$

(i) Real-valued Data (j) $\mathcal{N}(0, 10)$ (k) $\mathcal{N}(10, 10)$ (l) $\mathcal{N}(10, 100)$

Figure 2. [Synthetic Continuous Data] The first column shows the distribution of the inferred likelihood weights $\mathbf{w}^d$ when the ground truth data is (a) interval, (e) positive real-valued, and (i) real-valued. The remaining columns show example histograms of the datasets.

where $g^{-1} : \Re_+ \to \Re$ is the inverse function of g, i.e., $g^{-1}(g(y)) = y$. Therefore, $y^d_{n\text{count}}$ is sampled from a Gaussian truncated on the left by $g^{-1}(x^d_n)$ and on the right by $g^{-1}(x^d_n + 1)$.

Sampling likelihood assignments. In order to improve the mixing properties of the sampler, when sampling $\{s^d_n\}$ we integrate out the pseudo-observations $\{y^d_{n\ell}\}$. Then, the posterior probability of each observation being assigned to the likelihood model ℓ can be obtained as

$$p(s_n^d = \ell|\mathbf{w}^d, \mathbf{Z}, \{\boldsymbol{b}_\ell^d\}) = \frac{w_\ell^d\, p_\ell(x_n^d|\boldsymbol{z}_n, \boldsymbol{b}_\ell^d)}{\sum_{\ell'\in\mathcal{L}^d} w_{\ell'}^d\, p_{\ell'}(x_n^d|\boldsymbol{z}_n, \boldsymbol{b}_{\ell'}^d)}.$$

Sampling likelihood weights. We assume the prior distribution on the vector $\mathbf{w}^d$ to be a Dirichlet distribution with parameters $\{\alpha_\ell\}_{\ell\in\mathcal{L}^d}$. Then, by conjugacy, we can sample $\mathbf{w}^d$ given the likelihood assignments $\mathbf{S}$ from a Dirichlet distribution with parameters $\{\alpha_\ell + \sum_n \delta(s^d_n == \ell)\}_{\ell\in\mathcal{L}^d}$.

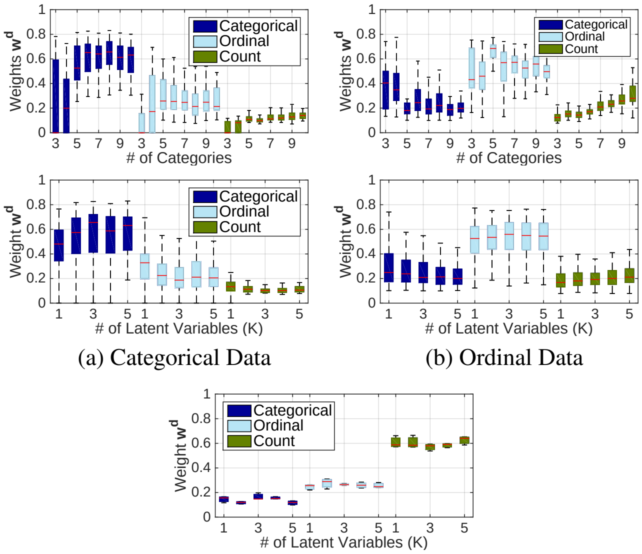
Scalability. The overall complexity of Algorithm 1 is $\mathcal{O}(NDL_{max} + K^3)$ per iteration, where N is the number of objects, D the number of attributes, L_{max} the maximum number of considered data types (or likelihood models) and K the size of the low-rank representation. In all of our experiments, we run the MCMC for 5000 iterations, which last 10-100 minutes depending on the dataset.

4. Evaluation

4.1. Experiments on synthetic data

In this section, we show that the proposed method is able to accurately discover the true statistical type of variables in synthetic datasets, where we have perfect knowledge of the distribution from which the data have been generated. First, we focus on continuous variables by generating univariate datasets with $1,000$ observations sampled from a known probability density function, which corresponds to i) a Gaussian distribution when considering real-valued data; ii) a Gamma distribution for positive real-valued data; and iii) a (scaled) Beta distribution for interval data lying in the interval $(0, \theta_L)$ where θ_L takes values 0.1, 1 or 100. Figure 2 shows the distribution, by means of a *boxplot*,[2] of the inferred likelihood weights $\mathbf{w}^d$ for 10 independent simulations of Algorithm 1 with 500 iterations on 10 independent datasets generated with the parameters detailed in the figure. Reassuringly we observe that the proposed method identifies interval data as the most likely type of data for the three considered Beta distributions; moreover, as the tail of the Beta distribution increases, so does the weight given to the positive real-valued variables. This effect can be explained by the finite size of the dataset, since it is hard to determine whether the variable is limited to values smaller than θ_L, or we simply have not observed them in the finite set of observations. A similar effect occurs when applying our method to data sampled from Gamma (Figure 2(e)-(h)) and Gaussian (Figure 2(i)-(l)) distributions. Here, we observe that in addition to, respectively, positive real-valued and real-valued data types, our model finds that the variable may also be of interval data type. This effect is larger for Gaussian variables, since in this example the Gaussian is a more heavy-tailed distribution than the Gamma.

[2] In a boxplot, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the 10th and 90th percentiles.



(a) Categorical Data

(b) Ordinal Data

(c) Count Data

Figure 3. [Synthetic Discrete Data] Distribution of the inferred likelihood weights $\mathbf{w}^d$ when the ground truth data is (a) categorical, (b) ordinal, and (c) count data. For categorical and ordinal data, we plot the likelihood weight distribution with respect to both the number of categories in the data and the model complexity K, and for count data with respect to K.

Next, we study whether the proposed model is able to disambiguate among different discrete types of variables, particularly, among categorical, ordinal and count data. To this end, we generate three types of datasets of size $1,000$. In the first type we account for categorical data by sampling a multinomial variable with R categories, where the probability of the categories is sampled from a Dirichlet distribution. Then, for each category we sample a multidimensional Gaussian centroid that corresponds to the mean of the multivariate Gaussian observations that complete the dataset. To account for ordinal observations, we first sample the first variable in our dataset from a uniform distribution in the interval $(0, R)$, which we randomly divide into R categories that correspond to the ordinal variable in our dataset. Finally, to account for count data we first generate a Gamma variable sampled from $\Gamma(\alpha, \alpha/4)$, and then generate the counting variable in the dataset by taking the floor of the Gamma variable. For both categorical and ordinal data, we generate 10 independent datasets for each value of the number of categories $R \in \{3, \dots, 10\}$, and for count data we generate another 10 datasets for each value of $\alpha \in \{2, \dots, 8\}$. Figure 3 summarizes the likelihood weights obtained for each type of datasets (*i.e.*, for each type of discrete variable) after running on each dataset 10 independent simulations of Algorithm 1 with 500 iterations for different model complexity values, *i.e.*, for different numbers of latent feature variables $K = 1, \dots, 10$. In this figure we can observe that we can accurately discover the true type of discrete variable robustly and independently of the assumed model complexity K. We also observe on the top row of Figure 3(a)-(b) that i) as the number of categories R in the discrete variable decreases, the harder is to distinguish between ordinal and categorical data, *i.e.*, to find out whether the data takes values in a ordered set or in an unordered set; and ii) as R in ordinal data increases, the ordinal variable is more likely to be identified as count data. Both of these effects are intuitively sensible.

Table 1. Information on real datasets

Dataset	N	D	# of Discrete	# of Binary
Abalone	$4,177$	9	2	0
Adult	$32,561$	15	12	2
Chess	$28,056$	7	7	0
Dermatology	366	35	35	0
German	1000	21	20	4
Student	395	33	33	13
Wine	177	14	2	0

4.2. Experiments on real data

In this section, we evaluate the performance of the proposed method on seven real datasets collected from the UCI machine learning repository (Lichman, 2013). Table 1 summarizes theses datasets by providing the number of objects and attributes in the dataset, as well as how many of these attributes are discrete.

In order to quantitatively evaluate the performance of the proposed method, we select at random 10% of the observations in each dataset as a held-out set and compare the predictive performance, in terms of average test log-likelihood per observation, of our method with a baseline method. The baseline method corresponds to a latent feature model in which all the continuous variables are modeled as real-valued data and the discrete variables as categorical data. Figure 4 shows the obtained results for our method (solid line) and the baseline (dashed line) for several values of the model complexity (*i.e.*, the number of latent features K) averaged over 10 independent runs of the corresponding inference algorithms. Here, we observe that i) both methods provide robust results with respect to the number of variables K; and ii) our method clearly outperforms the baseline in all the datasets, except for the Student dataset where the baseline performs slightly better. In other words, this figure shows that by taking into account the uncertainty in the statistical types of the variables, we provide a better fitting of the data.

Additionally, Table 2 shows the list of (non-binary) attributes in the Adult and the German datasets together with the data types with larger inferred likelihood weights,[3] *i.e.*, the discovered statistical data types. Here, the number in parenthesis corresponds to the observed number of categories in discrete data. The very heterogeneous nature of these datasets explains the substantial gain observed in Figure 4. Moreover, Table 2 shows some expected results, *e.g.*,

[3] In cases in which two data types present very similar likelihood weights ($< 10\%$ difference), we display both of them.

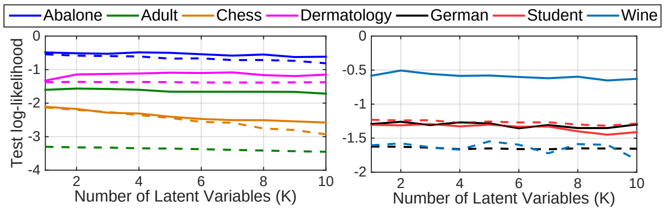


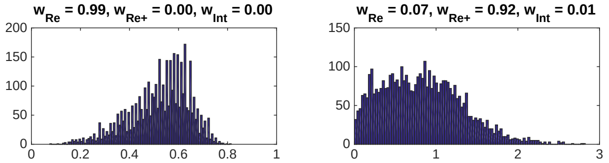
Figure 4. [Real Data] Comparison between our model (solid) and the baseline (dashed) in terms of average test log-likelihood per observation evaluated on a held-out set containing 10% of the observations in each dataset.

Table 2. Inferred data types.

Adult		German	
Attribute	Type	Attribute	Type
age (74)	ord.	status account (4)	cat.
workclass (8)	cat.	duration (69)	ord.
final weight	positive	credit hist. (5)	cat./ord.
education (16)	cat.	purpose (10)	cat./ord.
education num. (16)	cat.	amount	interval
marital status (7)	cat.	savings (5)	ord.
occupation (14)	cat./ord.	installment (5)	cat./ord.
relationship (6)	ord.	personal status (4)	cat.
race (5)	cat.	debtors (4)	ord.
sex (2)	binary	residence (3)	cat.
capital-gain	real	property (4)	cat./ord.
capital-loss	real	age (57)	count
hours per week (99)	cat./ord.	plans (3)	cat.
native-country (41)	ord.	housing (3)	ord.
		# credits (4)	ord.
		job (4)	ord.

marital status and *race* are identified as categorical, while the *age* is of count data type for both datasets. However, other results might seem surprising. For example, the *duration (in months)*, which one would expect it to be count data, is identified as ordinal; or the *a priori* categorical attributes *native country* and *job* are inferred to be ordinal.

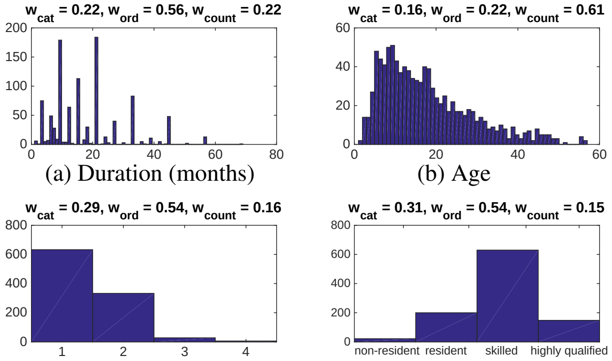
In order to better understand these results, we show the histograms of several variables in these datasets and the associated inferred likelihood weights. Figure 5 shows the histograms of two continuous variables, *length* and *weight* of the Abalone dataset, which take only positive real values, but are assigned to different data types (respectively, to real-valued and positive real-valued data). This can be explained by the fact that, while the distribution of the *length* presents large tails, the distribution of the *weight* is clearly truncated at zero. Additionally, Figure 6(a)-(b) shows two discrete variables, the *duration (in months)* and the *age* in German data, which based on the documentation are expected to be count data. However, our model assigns the duration to ordinal data. This result can be explained by the irregular distribution that this variable has. In count data the distance between every two consecutive values should be roughly the same (there is the same distance from "1 pen" to "2 pens" as from "2 pens" to "3 pens", that is 1 pen), resulting therefore in smooth probability mass functions. We found in Figure 6(c)-(d) that while the *number of credits* and the *job* variables can be a priori thought as respectively count and categorical data, they are both inferred to be ordinal data. In the case of the *number of credits*, this can be explained by the small (*finite*) number of values that the variable takes, while in the case of the *job*, this assignment can be explained by the labels of its categories, *i.e.*, {*unskilled non-resident, unskilled resident, skilled employee and highly qualified employee*}, which clearly represent an ordered set.



(a) Length (mm)

(b) Weight (grams)

Figure 5. [Abalone dataset]



(a) Duration (months)

(b) Age

(c) #of Credits

(d) Job

Figure 6. [German dataset]

From these results, we can conclude that i) our model accurately discovers the true statistical type of the data, which might not be easily extracted from its documentation; and by doing so, ii) it provides a better fit of the data. Moreover, apparent failures are in fact sensible when data histograms are carefully examined.

5. Conclusions

In this paper, we presented the first approach to automatically discover the statistical types of the variables in a dataset. Our experiments showed that the proposed approach accurately infers the data type, or equivalently likelihood model, that best fits the data.

Our work opens many interesting avenues for future work. For example, it would be interesting to extend the proposed method to account for other data types. We would like to include directional data, also called circular data, which arise in a multitude of data-modelling contexts ranging from robotics to the social sciences (Navarro et al., 2016). Moreover, since the proposed method can be seen as a likelihood selection method, it would be interesting to study how to incorporate our framework in any statistical machine learning tool, where the likelihood model, instead of being fixed *a priori*, would be inferred directly from the data jointly with the rest of the model parameters.

Acknowledgement

Isabel Valera acknowledges her Humboldt Research Fellowship for Postdoctoral Researchers, which funded this research during her stay at the Max Planck Institute for Software Systems.

References

Agresti, A. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

Ando, T. *Bayesian model selection and statistical modeling*. CRC Press, 2010.

Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

Burnham, K. P and Anderson, D. R. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.

Chu, W. and Ghahramani, A. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(Jul):1019–1041, 2005a.

Chu, W. and Ghahramani, Z. Gaussian processes for ordinal regression. *J. Mach. Learn. Res.*, 6:1019–1041, December 2005b. ISSN 1532-4435.

Dong, X. Luna and Srivastava, D. Big data integration. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pp. 1245–1248. IEEE, 2013.

Girolami, M. and Rogers, S. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18:2006, 2005.

Griffiths, T. L. and Ghahramani, Z. The Indian buffet process: an introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

Hellerstein, J. M. Quantitative data cleaning for large databases, 2008.

Hernandez-Lobato, J. M., Lloyd, J. R., Hernandez-Lobato, D., and Ghahramani, Z. Learning the semantics of discrete random variables: Ordinal or categorical? In *NIPS Workshop on Learning Semantics*, 2014.

Hilbe, J. M. *Negative binomial regression*. Cambridge University Press, 2011.

Kandel, S., Heer, J., Plaisant, C., Kennedy, J., van Ham, F., Riche, N. H., Weaver, C., Lee, B., Brodbeck, D., and Buono, P. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Navarro, A. KW, Frellsen, J., and Turner, R. E. The multivariate generalised von mises: Inference and applications. *arXiv preprint arXiv:1602.05003*, 2016.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 2007.

Schmidt, M. N, Winther, O., and Hansen, L. K. Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pp. 540–547. Springer, 2009.

Valera, I. and Ghahramani, Z. General table completion using a Bayesian nonparametric model. In *Advances in Neural Information Processing Systems 27*, 2014.