Multimodal Data Fusion in High-Dimensional Heterogeneous Datasets via Generative Models

Yasin Yilmaz*, Mehmet Aktukmak**, and Alfred O. Hero**

***Abstract*—The commonly used latent space embedding techniques, such as Principal Component Analysis, Factor Analysis, and manifold learning techniques, are typically used for learning effective representations of homogeneous data. However, they do not readily extend to heterogeneous data that are a combination of numerical and categorical variables, e.g., arising from linked GPS and text data. In this paper, we are interested in learning probabilistic generative models from high-dimensional heterogeneous data in an unsupervised fashion. The learned generative model provides latent unified representations that capture the factors common to the multiple dimensions of the data, and thus enable fusing multimodal data for various machine learning tasks. Following a Bayesian approach, we propose a general framework that combines disparate data types through the natural parameterization of the exponential family of distributions. To scale the model inference to millions of instances with thousands of features, we use the Laplace-Bernstein approximation for posterior computations involving nonlinear link functions. The proposed algorithm is presented in detail for the commonly encountered heterogeneous datasets with real-valued (Gaussian) and categorical (multinomial) features. Experiments on two high-dimensional and heterogeneous datasets (NYC Taxi and MovieLens-10M) demonstrate the scalability and competitive performance of the proposed algorithm on different machine learning tasks such as anomaly detection, data imputation, and recommender systems.**

***Index Terms*—heterogeneous data integration, latent variable models, variational inference, factor analysis, exponential family of distributions**

I. Introduction

Finding lower-dimensional latent space representations of high-dimensional datasets is an important unsupervised learning problem for several objectives such as dimensionality reduction, visualization, exploratory data analysis, and data fusion. PCA is the most commonly used latent space embedding technique. It finds a succinct representation of the data points in terms of a smaller number of low-dimensional features obtained by linearly mixing the original features in such a way as to maximize variance. Such linear mixing coefficients are given by the eigenvectors of the covariance matrix of the feature variables that correspond to the K largest eigenvalues, where K is the desired number of new features ($K \leq P$). For heterogeneous data, e.g., consisting of a numerical and a non-numerical variable, the standard sample covariance, called the Pearson covariance, is not directly applicable. For ordinal data Pearson introduced a modified covariance, called the polychoric correlation, that estimates the association between several ordinal variables by modeling them as quantized bivariate Gaussian random variables [1]. On the other hand, for mixed continuous and ordinal data, the polyserial correlation estimates association between the variables where, again, the ordinal data is modeled as quantized Gaussian [2]. PCA based on polyserial and polychoric correlations is often used for dimensionality reduction in heterogeneous datasets [3], [4]. However, both polyserial and polychoric correlations are designed for categorical variables that are ordinal, i.e., their values are linearly ordered [2]. Following a probabilistic approach PCA can also be generalized to exponential family for homogeneous datasets [5], [6].

Factor analysis (FA) is another well known latent space embedding technique, which includes PCA as a special case [7]. The introduction of factor analysis is often attributed to Charles Spearman's work in 1904 [8], yet its roots can be traced to the earlier works of Francis Galton [9]. Factor analysis is a latent variable model that decomposes a data matrix into low dimensional explanatory variables, called factors. Similar to PCA, factor analysis does not readily extend to heterogeneous data [10, Ch. 5]. In [11], a mixture of factor analyzers is presented for heterogeneous data consisting of continuous and categorical variables. Similar to classical factor analysis, in [11], instances (rows of the data matrix) are modeled independently with latent factor loading coefficients, whereas the features (columns of the data matrix) of an instance are linear combinations of factor loadings where the weights are called factor scores. Probabilistic Canonical Correlation Analysis (PCCA) [12] similarly models a pair of Gaussian feature vectors using a weighted sum of latent factors, called canonical components. Independent Component Analysis (ICA) and its extension Independent Vector Analysis (IVA), which is also a generalization of Canonical Correlation Analysis (CCA), are also used for multimodal data fusion [13], [14]. Regarding multimodal data, similar to factor analysis and PCA, manifold learning methods, such as Laplacian eigenmaps [15] and Isomap [16], also require homogeneity among data points. Typically, such manifold learning methods require computing an affinity matrix, but it is not clear how to define a unified similarity or distance metric for disparate features (e.g., numerical and categorical). There is also a large literature on multi-view learning (e.g., [17], [18]) which aims at performing specific machine learning tasks using heterogeneous datasets. Some of these methods, such as parallel ICA, e.g., [19], [20], have been applied to categorical data. However, in a heterogeneous data setting, these existing techniques [12]–[18]

*Electrical Engineering Department, University of South Florida, Tampa, FL 33620.

**Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109. The research in this paper was partially supported by a grant from the US Army Research Office under contract W911NF-15-1-0479 and a grant from the US National Science Foundation under award 2040572.

treat the categorical data in the same way as continuous-valued data, which is a strictly suboptimal approach.

There are also methods which address specific heterogeneous data applications using latent factor models, e.g., [21], [22]. In [23], a generic method based on generalized linear models is proposed to build graphical models from the exponential family of distributions. However, a joint analysis of the aggregated exponential family is not discussed in [23]. While the mixtures of factor analyzers methods proposed in [24], [25] jointly analyze heterogeneous Gaussian data, they do not allow for heterogeneous data types, such as numerical and categorical, as the proposed mixture is based on classical factor analyzers.

This paper develops a general approach to joint factor analysis for heterogeneous data, called multimodal factor analysis (MMFA). MMFA, originally introduced in [26], is a Bayesian approach that models different types of data using latent factor loadings specific to each data type and latent factor scores that are common to the data types. It was applied to event detection in Twitter [27] in order to fuse categorical and spherical data that are modeled by multinomial and von Mises-Fisher distributions, respectively.

In this paper, we present MMFA as a comprehensive unsupervised learning tool for learning generative models in high-dimensional and heterogeneous datasets explainable by exponential family of distributions. Specifically, our contributions can be summarized as follows.

- As opposed to the preliminary work [26], [27], the proposed generalized MMFA model provides a tractable unified framework for jointly analyzing heterogeneous features from the exponential family of distributions through linking their natural parameters with a common factor score vector.
- Motivated by the Bernstein-von Mises theorem [35] MMFA finds Gaussian approximations to the posterior distribution of latent factor loadings for the data types whose natural parameters require nonlinear link functions, e.g., multinomial distribution (a.k.a. Laplace-Bernstein approximation).
- MMFA easily scales to high-dimensional datasets with many heterogeneous features and instances, as demonstrated by the experimental results, thanks to the Laplace-Bernstein approximations.

The problem formulation is given in Section II, and the proposed generalized MMFA model is introduced in Section III. In Section IV, a variational learning algorithm for the proposed model is presented. In Section V, the MMFA algorithm is illustrated for a heterogeneous dataset consisting of Gaussian and multinomial components, and an analysis of computational complexity and mean-squared-error (MSE) performance is presented. We also demonstrate the usage of MMFA in unsupervised learning problems using real datasets (Section VI). Finally, the paper is concluded in Section VII.

II. Problem Formulation

Consider a heterogenous random data structure composed of M different data types, called modalities, from the same source. Observed are P realizations of this data structure, called instances. Assume that the data from each modality can be modeled with a probability distribution from the *exponential dispersion family* (e.g., Gaussian, Poisson, multinomial), which is a generalization of the exponential family [28]. For each modality m and each instance i, if the modality corresponds to a continuous random variable of dimension D_m then it can be represented as a data vector $\boldsymbol{x}_{im} = [x_{im}^1 \dots x_{im}^{D_m}]^T$ with probability density function (pdf) given by

$$f_m(\boldsymbol{x}_{im}|\boldsymbol{\eta}_m, \tau_m) = h_m(\boldsymbol{x}_{im}, \tau_m) \\ \exp\left\{\tau_m \left[\boldsymbol{\eta}_m^T \; \boldsymbol{s}_m(\boldsymbol{x}_{im}) - a_m(\boldsymbol{\eta}_m)\right]\right\}; \tag{1}$$

if $\boldsymbol{x}_{im}$ is discrete-valued, its probability mass function (pmf) is given by

$$f_m(\boldsymbol{x}_{im}|\boldsymbol{\eta}_m, \tau_m) = h_m(\boldsymbol{x}_{im}, \tau_m) \\ \exp\left\{\boldsymbol{\eta}_m^T \; \boldsymbol{s}_m(\boldsymbol{x}_{im}) - \tau_m a_m(\boldsymbol{\eta}_m)\right\}. \tag{2}$$

In (1) and (2), for modality m, $\boldsymbol{\eta}_m$ is a vector of natural parameters, $\boldsymbol{s}_m(\boldsymbol{x}_{im})$ is a vector of sufficient statistics, $a_m(\boldsymbol{\eta}_m)$ is the log-partition (i.e., log-normalization) function, $\tau_m \geq 0$ is the dispersion parameter, $(\cdot)^T$ denotes the transpose, and $i = 1, \dots, P, \; m = 1, \dots, M$. Note that the dimension D_m of each modality can be different, and $D = \sum_{m=1}^{M} D_m$ gives the total number of dimensions, i.e., features, in the dataset.

Table I gives examples of some popular probability distributions from the exponential family in terms of the representations in (1) or (2). The first three and the last three rows of Table I refer to continuous cases and discrete cases, i.e., (1) and (2), respectively.

III. Proposed Generative Latent Variable Model

Given a data structure of M modalities, D dimensions, and P instances, the MMFA model summarizes the data with a small number of K latent factors where $K \ll \min\{D, P\}$. This generative model is illustrated in Fig. 1 where the latent factors $\{e_{(k)}\}$ and the P instances are shown. Fig. 1 is a Markov graph in the sense that, conditioned on the $e_{(k)}$'s, the instances are independent and modalities are distributed according to different exponential family distributions of the form (1) and (2).

Each instance i is assumed to follow an exponential model of the form (1) or (2) with a natural parameter vector $\boldsymbol{\eta}_i = [\boldsymbol{\eta}_{i1}^T \cdots \boldsymbol{\eta}_{im}^T]^T \in \mathbb{R}^{\widetilde{M}}$ composed of an instance-wide latent matrix $\boldsymbol{E}$ and an instance-specific score vector $\boldsymbol{c}_i \in \mathbb{R}^K$:

$$\boldsymbol{\eta}_i = \boldsymbol{E}^T \boldsymbol{c}_i,$$

where $\boldsymbol{E} = [\boldsymbol{e}_1 \cdots \boldsymbol{e}_{\widetilde{M}}]$, $\boldsymbol{e}_m \in \mathbb{R}^K$ are latent vectors, $\widetilde{M}$ is the total number of natural parameters used to model the multimodal data [2]. The latent vectors $\{\boldsymbol{e}_{(1)}, \dots, \boldsymbol{e}_{(K)}\}$ in Fig. 1 are defined as the rows of the matrix $\boldsymbol{E}$. In analogy with other factor analysis methods, the latent vectors $\{\boldsymbol{e}_m\}$ are called factor loading vectors and $\{\boldsymbol{c}_i\}$ are called factor score vectors. In the proposed model, each instance i is characterized by the score vector $\boldsymbol{c}_i$. The same $\boldsymbol{c}_i$ is used to linearly model

[2] If every modality in the observation model has a single natural parameter, then $\widetilde{M} = M$.

TABLE I
EXAMPLES OF EXPONENTIAL DISPERSION FAMILY.

	pdf / pmf	$\boldsymbol{\eta}$	τ	$\boldsymbol{s}(\boldsymbol{x})$	$a(\boldsymbol{\eta})$	$h(\boldsymbol{x}, \tau)$
Gaussian (μ, σ^2)	$\frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$	μ	$1/\sigma^2$	x	$\eta^2/2$	$\frac{\exp(-x^2\tau/2)}{\sqrt{2\pi/\tau}}$
Exponential (λ)	$\lambda \exp(-\lambda x)$	$-\lambda$	1	x	$-\log(-\eta)$	1
von Mises-Fisher $(\boldsymbol{\mu}^{d\times 1}, \kappa)^1$	$C_d(\kappa) \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x})$	$\boldsymbol{\mu}$	κ	$\boldsymbol{x}$	0	$C_d(\tau)$
Poisson (λ)	$\lambda^x \exp(-\lambda)/x!$	$\log \lambda$	1	x	$\exp(\eta)$	$1/x!$
Binomial (n, p)	$\frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$	$\log \frac{p}{1-p}$	n	x	$\log(1+e^\eta)$	$\frac{\tau!}{x!(\tau-x)!}$
Multinomial $(n, \boldsymbol{p}^{d\times 1})$	$\frac{n!}{x_1!\cdots x_d!} p_1^{x_1} \cdots p_d^{x_d}$	$\begin{bmatrix} \log \frac{p_1}{p_d} \\ \vdots \\ \log \frac{p_{d-1}}{p_d} \\ 0 \end{bmatrix}$	n	$\boldsymbol{x}$	$\log \left(\sum_{j=1}^{d} e^{\eta_j} \right)$	$\frac{\tau!}{x_1!\cdots x_d!}$

[1] The von Mises-Fisher distribution is an extension of the Gaussian distribution to spherical data. For the d-dimensional von Mises-Fisher distribution, $C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$, where $I_{d/2-1}$ is the modified Bessel function of the first kind at order $d/2-1$, e.g., $C_3(\kappa) = \frac{\kappa}{2\pi(e^{\kappa}-e^{-\kappa})}$.

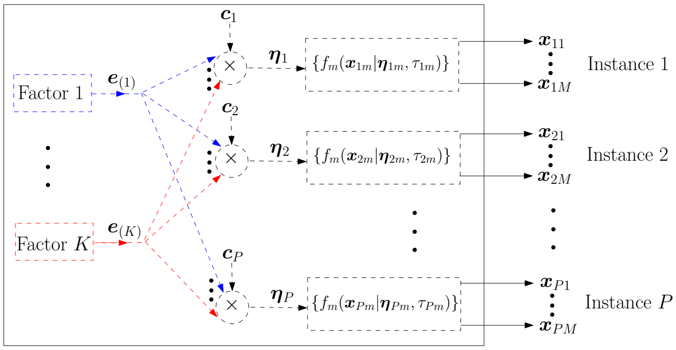


Fig. 1. Operational diagram of the considered system model. The outer rectangle represents the entire system; and the dashed part represents the proposed generative factor-based model, characterized by the latent vectors $\{\boldsymbol{e}_{(1)}, \ldots, \boldsymbol{e}_{(K)}\}$, $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_P\}$, and the exponential family distributions $\{f_1, \ldots, f_M\}$.

each natural parameter η_{im} associated with instance i for all of the M data modalities, e.g., (3)-(5).

For example, if the first three modalities are Gaussian, Poisson, and binomial, respectively, then from Table I, the natural parameter $\boldsymbol{\eta}_i$ is composed of three elements:

$$\mu_i = \boldsymbol{e}_1^T \boldsymbol{c}_i, \tag{3}$$

$$\log \lambda_i = \boldsymbol{e}_2^T \boldsymbol{c}_i, \tag{4}$$

$$\log \frac{p_i}{1-p_i} = \boldsymbol{e}_3^T \boldsymbol{c}_i. \tag{5}$$

The Gaussian case in (3) corresponds to the classical factor analysis model. Factor analysis, in its original form, is used to model the mean of continuous data through a linear combination of continuous latent variables [10], as in the Gaussian case given by (3). The classical factor analysis model does not provide a good fit for discrete data (e.g., Poisson data) or categorical data (e.g., binomial data) [29]. There are several latent variable models that extend factor analysis to non-Gaussian data, such as the general linear latent variable model [29], Poisson factor analysis [30], and latent Dirichlet allocation [31]. Different than those works, here we provide a joint model to deal with different data modalities together. In our proposed model, instead of the mean – e.g., λ for Poisson and p for binomial (see Table I) – we linearly model the natural parameter – e.g., $\log \lambda$ for Poisson (4) and $\log \frac{p}{1-p}$ for binomial (5) – which provides a general framework for the exponential family of distributions. This mapping is similar to the general linear latent variable model [29] and the generalized mixture of factor analyzers model [11], albeit with some important differences.

Firstly, we present a generic model for the joint analysis of exponential family where either the factor loadings $\{\boldsymbol{e}_m\}$ or the factor scores $\{\boldsymbol{c}_i\}$ can be modeled as latent variables, whereas they are strictly modeled as parameters and latent variables, respectively, in the existing factor analysis models including [29] and [11]. The proposed model can bring about significant tractability for high-dimensional heterogeneous datasets since computing the posterior of $\boldsymbol{c}_i$ involves all data modalities, whereas computing the posterior of $\boldsymbol{e}_m$ only requires data from modality m. We also provide a general scalable framework based on the Laplace-Bernstein approximation for fitting the proposed model to high-dimensional and heterogeneous datasets.

The proposed model is indeed a nonparametric model. Although a parametric model is used for each data vector $\boldsymbol{x}_{im}$ of instance i and modality m (see (1), (2)), we have a nonparametric model for the entire dataset $\{\boldsymbol{x}_{im}\}$ representing the collection of instances. This is because the number of parameters $\{\boldsymbol{c}_i\}$ linearly increases with the number of instances, and thus is asymptotically infinite.

IV. Learning the Model Parameters

We use the expectation-maximization (EM) approach to compute, in an iterative manner, the maximum likelihood estimates of the model parameters θ, which includes the score vectors $\{\boldsymbol{c}_i\}$, the hyperparameters of the prior distribution for $\{\boldsymbol{e}_m\}$, and the dispersion parameters $\{\tau_m\}$. The number of factors K can be selected in different ways, including Bayesian information criterion (BIC), Monte Carlo integration over a uniform on K, birth-death models, or evaluation of a knee in the scree plot of goodness of fit, e.g., as measured by the negative log-likelihood.

A. The EM Algorithm

In the original EM algorithm [33] for modality m, at each iteration n, in the expectation step (E-step), the expectation of the complete-data log-likelihood is computed with respect to the posterior distribution $g_{m,n-1}\big(\boldsymbol{e}_m|\{\boldsymbol{x}_{im}\}_i, \theta_{m,n-1}\big)$ of each latent vector $\boldsymbol{e}_m$, i.e., $Q_n(\theta_m) = \mathsf{E}_{g_{m,n-1}}\big[\log f_m\big(\{\boldsymbol{x}_{im}\}_i, \boldsymbol{e}_m|\theta_m\big)\big]$. Then, in the maximization step (M-step), the parameters $\theta_{m,n}$ are estimated by maximizing $Q_n(\theta_m)$ over the parameter space Θ_m, i.e., $\theta_{m,n} = \arg\max_{\theta_m \in \Theta_m} Q_n(\theta_m)$.

With the EM approach, the biggest challenge is to compute the posterior distribution in the E-step at each iteration n since the link function relating the natural parameters to latent vectors is in general nonlinear, and the posterior is multivariate. Markov chain Monte Carlo (MCMC) methods, such as the Gibbs sampler, can be used to sample from the posterior $g_{m,n-1}(\boldsymbol{e}_m|\{\boldsymbol{x}_{im}\}_i, \theta_{m,n-1})$. We might then use these samples to estimate $Q_n(\theta_m)$ for a given θ_m, and search over the parameter space to find the θ_m that maximizes $Q_n(\theta_m)$. Although the MCMC approach can enable the use of exact EM algorithm for inferring the model parameters, high-dimensional parameter space with thousands of features (large D) and millions of instances (large P) could be problematic for the convergence rate and computational complexity. We propose a variational EM approach to address this problem.

B. The Proposed Variational EM Algorithm

In the variational EM approach, a tractable probability distribution $\widetilde{g}_{m,n-1}(\boldsymbol{e}_m|\{\boldsymbol{x}_{im}\}_i, \theta_{m,n-1})$ is used to approximate the posterior. The objective is to select, from a tractable family of distributions, the distribution which is closest to the actual posterior in the KL-divergence sense, i.e., $\widetilde{g}_{m,n-1} = \arg\min_q \mathrm{KL}(q||g_{m,n-1})$. The design challenge here is to determine the tractable family of distributions such that $\widetilde{g}_{m,n-1}$ will be close to $g_{m,n-1}$. To this end we utilize the Bernstein-von Mises theorem, which states that the posterior is asymptotically, as the number of instances P increases, well approximated by a Gaussian distribution when the likelihood model is from exponential family and the prior is Lipschitz continuous, e.g., Gaussian, von Mises-Fisher, etc. [35], [36]. In the problem of interest with exponential family models, Gaussian priors, and large number of instance-feature interactions (e.g., $P \times D$ is on the order of millions), the Bernstein-von Mises theorem provides theoretical motivation for using Gaussian approximation to the posterior. Note that $P \times D$ gives the number of observations for finding the posterior of the K-dimensional latent vectors. Hence, following a variational EM approach, we approximate $g_{m,n-1}$ with a Gaussian $\widetilde{g}_{m,n-1}$.

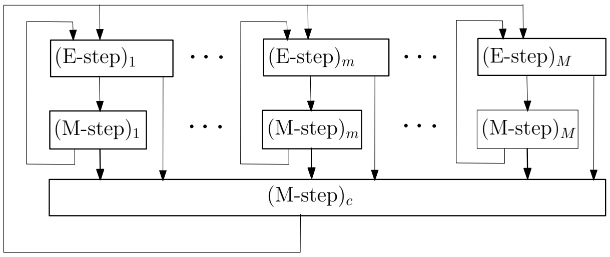


Fig. 2. Structure of the proposed variational EM algorithm. In each iteration, E-M steps of all modalities run in parallel. They are synchronized by a final M step for the score vector of each instance.

The Laplace technique approximates the posterior with the Gaussian $\mathcal{N}(\boldsymbol{\ell}, -\boldsymbol{H}(\boldsymbol{\ell})^{-1})$, where $\boldsymbol{\ell}$ is the mode of the posterior, and $\boldsymbol{H}(\boldsymbol{\ell})$ is the Hessian matrix (i.e., second-order derivative of $g_{m,n-1}$ with respect to $\boldsymbol{e}_m$) evaluated at the mode $\boldsymbol{\ell}$. The posterior mode typically does not have a closed form due to the nonlinear link function, and thus calls for iterative computation through a numerical optimization technique. The computation of the posterior mode and the Hessian matrix at each EM iteration, with a high-dimensional dataset, may be prohibitive. Similarly, higher-order variational inference techniques such as expectation propagation (EP) may incur significant computational complexity in high-dimensional datasets (large P, large D). Through moment matching EP will need to iteratively compute the mean vector and covariance matrix of the actual posterior $g_{m,n-1}$.

Hence, for scalability to large datasets, we resort to variational – in particular, quadratic – lower bounds, which significantly reduces the computational complexity compared to the Laplace approximation and EP by fixing the covariance matrix. Moreover, as noted in [7, p. 498], the variational lower bound method has additional flexibility, which leads to improved accuracy, compared to the Laplace method. This motivates us to approximate the log-partition function $a(\eta)$, which is the problematic term in the complete-data log-likelihood $\log f_m\big(\{\boldsymbol{x}_{im}\}_i, \boldsymbol{e}_m|\theta_m\big)$ with a quadratic term to obtain the Gaussian approximation $\widetilde{g}_{m,n-1}$.

For instance, we approximate $\log(-\eta)$ and e^{η} in the exponential and Poisson likelihoods (see Table I) using the second-order Taylor series expansion around 1 and 0, respectively. Specifically, we obtain an evidence lower bound (ELBO) by using $\min\{-1, -x_i^2\}$ and $\max\{1, x_i\}$ for the second-order derivative term, where x_i approximates $1/\lambda$ and λ in the exponential and Poisson case, respectively. In the next section, we will explain this procedure in detail for the multinomial likelihood, which is commonly used in the real-world datasets for categorical features. A Gaussian prior $\pi(\boldsymbol{e}_m)$ is assumed for each modality to facilitate the Gaussian approximation for the posterior. Note that no approximation to the posterior is needed for the Gaussian likelihood since the posterior is already Gaussian with a conjugate prior. For the von Mises-Fisher distribution, which is an extension of Gaussian distribution to spherical data, we use a von Mises-Fisher prior, which is conjugate to the likelihood. However, in this case, an approximation is still needed due to the constraint that the mean vector is a unit-length vector (see [27] for details).

In the proposed variational EM algorithm, for all modalities EM steps are run in parallel, which is followed by the M-step for the score vectors $\{\boldsymbol{c}_i\}$, as shown in Fig. 2. Since $\boldsymbol{c}_i$ is common to all modalities of instance i, it is updated by receiving related information from all EM steps for different modalities. Each $\boldsymbol{c}_i$ can be updated in parallel.

V. Example: Gaussian and Multinomial

To illustrate the proposed model and variational EM algorithm, in this section, we consider a bimodal dataset $\boldsymbol{X} = [\boldsymbol{Y}\boldsymbol{Z}]$ from the same source consisting of a real-valued matrix $\boldsymbol{Y} \in \mathbb{R}^{P\times D_1}$ and a categorical data matrix $\boldsymbol{Z} \in \mathbb{Z}_+^{P\times D_2}$ with P, D_1, and D_2 denoting the number of instances, number of real-valued features, and the number of categories, respectively. $\mathbb{Z}_+$ denotes the set of nonnegative integers. Assume the entries y_{ij} of $\boldsymbol{Y}$ and the rows $\boldsymbol{z}_{(i)}$ of $\boldsymbol{Z}$ are well modeled using the Gaussian $\mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ and the multinomial $\mathcal{M}(N_i, \boldsymbol{p}_i)$ models (where N_i is the number of experiments and $\boldsymbol{p}_i \in [0,1]^{D_2}$ is the probability vector), respectively.

As in (3) and (5), assuming K generative factors that are characterized by the latent vectors

$$
\begin{aligned}
&\{\boldsymbol{e}_{(k)} : \boldsymbol{e}_{(k)} \in \mathbb{R}^{D_1+D_2-1}, k = 1, \dots, K\},\\
\boldsymbol{E} = [\boldsymbol{U}\boldsymbol{V}] &= [\boldsymbol{u}_1 \cdots \boldsymbol{u}_{D_1} \boldsymbol{v}_1 \cdots \boldsymbol{v}_{D_2-1}] = [\boldsymbol{e}_{(1)} \cdots \boldsymbol{e}_{(K)}]^T,
\end{aligned}
$$

we linearly model the natural parameters of the Gaussian and the multinomial distributions, i.e.,

$$
\begin{aligned}
\mu_{ij} &= \boldsymbol{u}_j^T \boldsymbol{c}_i, \;\; j = 1, \dots, D_1\\
\log \frac{p_{ij}}{p_{iD_2}} &= \boldsymbol{v}_j^T \boldsymbol{c}_i, \;\; j = 1, \dots, D_2 - 1.
\end{aligned}
$$

The coefficient vector $\boldsymbol{c}_i$ represents each instance i in terms of the K latent factors. In the multinomial distribution, the last category is selected as pivot since one of the probabilities is fully determined by the other $D_2 - 1$ probabilities, i.e., the degree of freedom is one less than the number of categories (cf. Table I). Note that multinomial distribution covers as a special case categorical distribution ($N_i = 1, \forall i$), binomial distribution ($D_2 = 2$), and Bernoulli distribution ($D_2 = 2, N_i = 1, \forall i$). The two modalities considered in this section are the most common data types found in real-world datasets [39].

A. Model Parameters

1) Gaussian Parameters: For the Gaussian model, assuming the conjugate prior $\mathcal{N}(\boldsymbol{0}_K, \boldsymbol{I}_K)$ for $\boldsymbol{u}_j, j = 1, \dots, D_1$, where $\boldsymbol{0}_K$ and $\boldsymbol{I}_K$ denote the K-dimensional zero vector and identity matrix, we have the exact EM algorithm, i.e., at each iteration n

E-step: compute the posterior

$$
g_{j,n-1}\big(\boldsymbol{u}_j | \boldsymbol{y}_j, \theta_{j,n-1}\big) = \frac{f_1\big(\boldsymbol{y}_j, \boldsymbol{u}_j | \theta_{j,n-1}, \boldsymbol{c}_{i,n-1}\big)}{f_1\big(\boldsymbol{y}_j | \theta_{j,n-1}, \boldsymbol{c}_{i,n-1}\big)}
$$

M-step: estimate the parameters

$$
\theta_{j,n} = \arg\max_{\theta_j \in \Theta_j} \mathsf{E}_{g_{j,n-1}}\Big[\log f_1\big(\boldsymbol{y}_j, \boldsymbol{u}_j | \theta_j, \boldsymbol{c}_{i,n-1}\big)\Big],
$$

where $\theta_j = \{\sigma_{ij}^2\}_{i=1,\dots,P}$. From the classical factor analysis [10], [38], we have $\mathcal{N}(\boldsymbol{a}_{j,n}, \boldsymbol{B}_{j,n})$ as the posterior of $\boldsymbol{u}_j$ at iteration n, where

$$
\begin{aligned}
\boldsymbol{B}_{j,n} &= \Big(\boldsymbol{C}_{n-1} \boldsymbol{\Sigma}_{j,n-1}^{-1} \boldsymbol{C}_{n-1}^T + \boldsymbol{I}_K\Big)^{-1},\\
\boldsymbol{a}_{j,n} &= \boldsymbol{B}_{j,n} \boldsymbol{C}_{n-1} \boldsymbol{\Sigma}_{j,n-1}^{-1} \boldsymbol{y}_j, \qquad (6)\\
\boldsymbol{C}_{n-1} &= \Big[\boldsymbol{c}_{1,n-1} \cdots \boldsymbol{c}_{P,n-1}\Big],\\
\boldsymbol{\Sigma}_{j,n-1} &= \operatorname{diag}\Big(\sigma_{1j,n-1}^2 \cdots \sigma_{Pj,n-1}^2\Big),
\end{aligned}
$$

and the parameter update

$$
\sigma_{ij,n}^2 = \big(y_{ij} - \boldsymbol{a}_{j,n}^T \boldsymbol{c}_{i,n-1}\big)^2 + \boldsymbol{c}_{i,n-1}^T \boldsymbol{B}_{j,n} \boldsymbol{c}_{i,n-1}.
$$

Assuming an inverse-gamma distribution $InvGam(\alpha, \beta)$, which is the conjugate prior for the Gaussian variance, for σ_{ij}^2 it is straightforward to show that the update becomes

$$
\sigma_{ij,n}^2 = \frac{\big(y_{ij} - \boldsymbol{a}_{j,n}^T \boldsymbol{c}_{i,n-1}\big)^2 + \boldsymbol{c}_{i,n-1}^T \boldsymbol{B}_{j,n} \boldsymbol{c}_{i,n-1} + 2/\beta}{2(\alpha+1)+1}. \qquad (7)
$$

Combining the inputs from all features (Gaussian and multinomial) the coefficient vector $\boldsymbol{c}_i$ of instance i is updated as

$$
\begin{aligned}
\boldsymbol{c}_{i,n} = \arg\max_{\boldsymbol{c}_i} &-\frac{1}{2} \boldsymbol{c}_i^T \left(\sum_{j=1}^{D_1} \frac{\boldsymbol{B}_{j,n} + \boldsymbol{a}_{j,n} \boldsymbol{a}_{j,n}^T}{\sigma_{ij,n}^2}\right) \boldsymbol{c}_i\\
&+ \boldsymbol{c}_i^T \sum_{j=1}^{D_1} \frac{y_{ij} \boldsymbol{a}_{j,n}}{\sigma_{ij,n}^2} + \zeta_n(\boldsymbol{c}_i), \qquad (8)
\end{aligned}
$$

where the input from the multinomial features $\zeta(\boldsymbol{c}_i)$ will be derived next.

2) Multinomial Parameters: In the multinomial likelihood,

$$
\begin{aligned}
f_2\big(\{\boldsymbol{z}_{(i)}\} | \{\boldsymbol{v}_j\}, \boldsymbol{c}_i, N_i\big) &= \prod_{i=1}^{P} \frac{N_i!}{z_{i1}! \cdots z_{iD_2}!} \prod_{j=1}^{D_2} p_{ij}^{z_{ij}}\\
&= \prod_{i=1}^{P} \frac{N_i!}{z_{i1}! \cdots z_{iD_2}!} \frac{\prod_{j=1}^{D_2-1} e^{\boldsymbol{v}_j^T \boldsymbol{c}_i z_{ij}}}{\Big(1 + \sum_{j'=1}^{D_2-1} e^{\boldsymbol{v}_{j'}^T \boldsymbol{c}_i}\Big)^{N_i}},
\end{aligned}
$$

the dependency of the sum-of-exponentials (sum-exp) term in the denominator on all latent vectors $\{\boldsymbol{v}_j\}$ complicates the analysis considerably. First of all, we need to consider $\{\boldsymbol{v}_j\}$ together and obtain the posterior distribution of the combined latent vector $\boldsymbol{v} = [\boldsymbol{v}_1^T \cdots \boldsymbol{v}_{D_2-1}^T]^T$, which is $K(D_2-1)$-dimensional. More importantly, finding the exact posterior is not tractable due to the presence of $\{\boldsymbol{v}_j\}$ in the sum-exp function. We rewrite the above likelihood expression in a more compact form

$$
\begin{aligned}
&f_2\big(\{\boldsymbol{z}_{(i)}\} | \{\boldsymbol{v}_j\}, \boldsymbol{c}_i, N_i\big)\\
&\qquad = \prod_{i=1}^{P} \frac{N_i!}{z_{i1}! \cdots z_{iD_2}!} e^{\sum_{j=1}^{D_2-1} \boldsymbol{v}_j^T \boldsymbol{c}_i z_{ij} - N_i \operatorname{lse}(\boldsymbol{\eta}_i)},\\
&\qquad = \prod_{i=1}^{P} \frac{N_i!}{z_{i1}! \cdots z_{iD_2}!} e^{\boldsymbol{v}^T \boldsymbol{C}_i \boldsymbol{z}_i - N_i \operatorname{lse}(\boldsymbol{\eta}_i)},
\end{aligned}
$$

where $\mathrm{lse}(\boldsymbol{\eta}_i) = \log\left(1 + \sum_{j'=1}^{D_2-1} e^{\boldsymbol{v}_{j'}^T \boldsymbol{c}_i}\right)$ is the log-sum-exp function, $\boldsymbol{\eta}_i = [\boldsymbol{v}_1^T \boldsymbol{c}_i \cdots \boldsymbol{v}_{D_2-1}^T \boldsymbol{c}_i]^T$, $\boldsymbol{C}_i = \boldsymbol{I}_{D_2-1} \otimes \boldsymbol{c}_i$ with $\otimes$ being the Kronecker product, and $\boldsymbol{z}_i = [z_{i1} \cdots z_{iD_2-1}]^T$.

To obtain an approximate posterior $\widetilde{g}_{2,n}(\boldsymbol{v}|\{\boldsymbol{z}_i, \boldsymbol{c}_i, N_i\})$, as outlined in Section IV-B, we derive a lower bound for the likelihood, and accordingly for the complete-data log-likelihood, through the second-order Taylor series expansion of $\mathrm{lse}(\boldsymbol{\eta}_i)$ around a fixed point $\boldsymbol{\psi}_i$,

$$\begin{aligned}\mathrm{lse}(\boldsymbol{\eta}_i) &= \mathrm{lse}(\boldsymbol{\psi}_i) + (\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T \nabla \mathrm{lse}(\boldsymbol{\psi}_i) \\ &\quad + \frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T \nabla^2 \mathrm{lse}(\boldsymbol{\psi}_i + \epsilon(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i))(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i) \\ &\le \mathrm{lse}(\boldsymbol{\psi}_i) + (\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T \nabla \mathrm{lse}(\boldsymbol{\psi}_i) \\ &\quad + \frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T \boldsymbol{A}(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i), \end{aligned} \tag{9}$$

where $\epsilon \in [0,1]$, and $\boldsymbol{A} = \frac{1}{2}\left(\boldsymbol{I}_{D_2-1} - \frac{\boldsymbol{1}\boldsymbol{1}^T}{D_2}\right)$ from [40]. Note that we defined a new variable $\boldsymbol{\psi}_i$ for each instance i. We will show how to update it in Proposition 1. The gradient is given by the probability vector induced by $\boldsymbol{\psi}_i$, i.e., $\nabla \mathrm{lse}(\boldsymbol{\psi}_i) = \frac{e^{\boldsymbol{\psi}_i}}{1+\sum_{j=1}^{D_2-1} e^{\psi_{ij}}} = \boldsymbol{p}_{\boldsymbol{\psi}_i}$. Replacing the lse function with this quadratic upper bound we obtain the following lower bound for the likelihood

$$\begin{aligned} f_2(\{\boldsymbol{z}_{(i)}\}|\boldsymbol{v},\boldsymbol{c}_i, N_i) \ge \widetilde{f}_2(\{\boldsymbol{z}_{(i)}\}|\boldsymbol{v},\boldsymbol{c}_i, N_i) = \\ e^{\boldsymbol{v}^T\left(\sum_{i=1}^P \boldsymbol{C}_i \widetilde{\boldsymbol{z}}_i\right) - \boldsymbol{v}^T\left(\sum_{i=1}^P \frac{N_i}{2} \boldsymbol{C}_i \boldsymbol{A} \boldsymbol{C}_i^T\right)\boldsymbol{v} + const.} \end{aligned} \tag{10}$$

where $\widetilde{\boldsymbol{z}}_i = \boldsymbol{z}_i - N_i\left(\boldsymbol{p}_{\boldsymbol{\psi}_i} - \boldsymbol{A}\boldsymbol{\psi}_i\right)$, and $const.$ denotes the constant terms with respect to $\boldsymbol{v}$. Assuming standard multivariate Gaussian $\mathcal{N}(\boldsymbol{0}_{(D_2-1)K}, \boldsymbol{I}_{(D_2-1)K})$ prior for $\boldsymbol{v}$ the lower bound for the complete-data likelihood is given by

$$f_2(\{\boldsymbol{z}_i\}, \boldsymbol{v}|\boldsymbol{c}_i, N_i) \ge e^{-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\omega})^T \boldsymbol{\Omega}^{-1}(\boldsymbol{v}-\boldsymbol{\omega}) + const.}, \tag{11}$$

where $\boldsymbol{\Omega} = (\sum_{i=1}^P N_i \boldsymbol{C}_i \boldsymbol{A} \boldsymbol{C}_i^T + \boldsymbol{I}_{(D_2-1)K})^{-1}$ and $\boldsymbol{\omega} = \boldsymbol{\Omega} \sum_{i=1}^P \boldsymbol{C}_i \widetilde{\boldsymbol{z}}_i$. From that lower bound we obtain an approximate posterior $\widetilde{g}_{2,n}(\boldsymbol{v}|\{\boldsymbol{z}_i, \boldsymbol{c}_i, N_i\}) = \mathcal{N}(\boldsymbol{\omega}, \boldsymbol{\Omega})$. In the M-step, for computational efficiency, following the approach in [37] we update the parameters by maximizing the expected lower bound $\mathsf{E}_{\widetilde{g}_{2,n}}\left[\log \widetilde{f}_2(\{\boldsymbol{z}_i\}, \boldsymbol{v}|\boldsymbol{c}_i, N_i)\right]$ instead of maximizing $\mathsf{E}_{\widetilde{g}_{2,n}}\left[\log f_2(\{\boldsymbol{z}_i\}, \boldsymbol{v}|\boldsymbol{c}_i, N_i)\right]$. It is shown in [37] that maximizing the former is asypmtotically equivalent to, and computationally more efficient than maximizing the latter. Using this approximate posterior the M-step (i.e., parameter updates) of the variational EM algorithm can be derived as in the Gaussian case; however, the computational complexity might be prohibitive due to high dimensionality, e.g., $\boldsymbol{\Omega}$ requires inverting a $(D_2-1)K \times (D_2-1)K$ matrix. Typically, the number of factors K gets small values, whereas the number of categories may be large, e.g., a dictionary of words in topic modeling [31]. We next present a result that shows, indeed, the underlying dimensionality is $K \times K$.

Proposition 1. *At iteration n, the multinomial parameters $\boldsymbol{\psi}_i$ and the coefficient vector $\mathbf{c}_i$ can be updated as follows*

$$\boldsymbol{\psi}_{i,n} = \boldsymbol{\Phi}_n^T \boldsymbol{c}_{i,n-1} \tag{12}$$

$$\boldsymbol{c}_{i,n} = \arg\max_{\boldsymbol{c}_i} -\frac{1}{2}\boldsymbol{c}_i^T \boldsymbol{H}_{i,n} \boldsymbol{c}_i + \boldsymbol{c}_i^T \boldsymbol{\rho}_{i,n}, \tag{13}$$

$$\begin{aligned}\boldsymbol{H}_{i,n} &= N_i\Big(\frac{(D_2-1)^2}{2D_2}\boldsymbol{F}_n^{-1} + \frac{D_2-1}{2D_2}\boldsymbol{\Delta}_n + \frac{1}{2}\boldsymbol{\Phi}_n\boldsymbol{\Phi}_n^T \\ &\quad - \frac{1}{2D_2}(\boldsymbol{\Phi}_n \boldsymbol{1}_{D_2-1})(\boldsymbol{\Phi}_n \boldsymbol{1}_{D_2-1})^T\Big) \\ &\quad + \sum_{j=1}^{D_1} \frac{\boldsymbol{B}_{j,n} + \boldsymbol{a}_{j,n}\boldsymbol{a}_{j,n}^T}{\sigma_{ij,n}^2} \end{aligned}$$

$$\boldsymbol{\rho}_{i,n} = \boldsymbol{\Phi}_n \widetilde{\boldsymbol{z}}_{i,n} + \sum_{j=1}^{D_1} \frac{y_{ij}\boldsymbol{a}_{j,n}}{\sigma_{ij,n}^2},$$

using the matrices

$$\begin{aligned}\boldsymbol{F}_n &= \frac{1}{2}\boldsymbol{C}_{n-1}\boldsymbol{N}\boldsymbol{C}_{n-1}^T + \boldsymbol{I}_K \\ \boldsymbol{\Delta}_n &= \frac{\boldsymbol{I}_K - \boldsymbol{F}_n^{-1}}{D_2}\left[\boldsymbol{F}_n^{-1} + \left(\frac{\boldsymbol{F}_n}{D_2-1} + \boldsymbol{I}_K\right)^{-1}(\boldsymbol{I}_K - \boldsymbol{F}_n^{-1})\right] \\ \boldsymbol{\Phi}_n &= \boldsymbol{F}_n^{-1}\boldsymbol{C}_{n-1}\widetilde{\boldsymbol{Z}}_n + \boldsymbol{\Delta}_n\boldsymbol{C}_{n-1}(\widetilde{\boldsymbol{Z}}_n \boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T) \end{aligned} \tag{14}$$

where $\boldsymbol{B}_{j,n}$ and $\boldsymbol{a}_{j,n}$ are given by (6)*, the $K \times (D_2-1)$ matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{D_2-1}]$ is a reorganized form of the approximate posterior mean $\boldsymbol{\omega} = [\boldsymbol{\phi}_1^T \cdots \boldsymbol{\phi}_{D_2-1}^T]^T$, $\boldsymbol{N} =$ diag$(N_1, \ldots, N_P)$, and $\widetilde{\boldsymbol{Z}} = [\widetilde{\boldsymbol{z}}_1 \cdots \widetilde{\boldsymbol{z}}_P]^T$, $\widetilde{\boldsymbol{z}}_{i,n} = \boldsymbol{z}_i - N_i\left(\boldsymbol{p}_{\boldsymbol{\psi}_{i,n}} - \boldsymbol{A}\boldsymbol{\psi}_{i,n}\right)$.*

Proof: See Appendix. ■

Although not directly used in the learning algorithm, the complete form of the approximate posterior covariance (cf. (11)) is given by $\boldsymbol{\Omega}_n = \boldsymbol{I}_{D_2-1} \otimes \boldsymbol{F}_n^{-1} + \boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{\Delta}_n$.

3) Coefficient Vector: As shown in (13), we have a quadratic programming problem for the coefficient vector $\boldsymbol{c}_i$, which is simply solved as

$$\boldsymbol{c}_{i,n} = \boldsymbol{H}_{i,n}^{-1}\boldsymbol{\rho}_{i,n} \tag{15}$$

unless there is a constraint on $\boldsymbol{c}_i$, such as $c_{ik} \ge 0, \forall i, k$. If a constraint is added to the problem, a standard solver can be used, e.g., interior point methods. Additionally, depending on the application a convenient regularization, such as L^1-norm (Lasso) and L^2-norm (ridge regression), can be used to solve (15).

4) Algorithm: The resulting variational EM algorithm is summarized in Algorithm 1. Note that the EM steps for Gaussian (lines 4 and 5) and multinomial (lines 6 and 7) can be run in parallel, as shown in Fig. 2. Moreover, the coefficient vectors $\{\boldsymbol{c}_i\}$ can be updated in parallel (line 8).

B. Computational Complexity

In the following theorem, we show that the computational complexity of Algorithm 1 scales linearly with each dimension of the problem (i.e., number of instances, real-valued features, and categorical features). As a result, the proposed algorithm

Algorithm 1 The proposed EM algorithm for the Gaussian-multinomial example

```
Input $\boldsymbol{Y}, \boldsymbol{Z}$
Initialize $\{\boldsymbol{c}_i^{K\times 1}, \sigma_{ij}, \boldsymbol{\psi}_i^{D_2-1\times 1}\}$, $i = 1, \dots, P$, $j = 1, \dots, D_1$,
while not converged do
    Compute Gaussian posterior parameters $\{\boldsymbol{B}_j, \boldsymbol{a}_j\}$ as in (6)
    Update Gaussian parameters $\{\sigma_{ij}^2\}$ as in (7)
    Compute multinomial posterior parameters $\boldsymbol{F}, \boldsymbol{\Delta}, \boldsymbol{\Phi}$ as in (14)
    Update multinomial parameters $\{\boldsymbol{\psi}_i\}$ as in (12)
    Update coefficients $\{\boldsymbol{c}_i\}$ by solving (15)
end while
```

can be efficiently used for large datasets, as demonstrated in Section VI.

Theorem 1. *At each iteration of the proposed EM algorithm, given by Algorithm 1, the computational complexity linearly scales with the number of instances P, the number of real-valued features D_1, and the number of categories D_2. Specifically, the complexity is given by $O(K^3P + K^2PD_1 + KPD_2)$, where K is the number of factors.*

Proof: See Appendix. ■

Note that the number of factors is not an input from data, but a design parameter typically chosen to be a small number compared to the number of instances, $K \ll P$. Furthermore, even in mildly complex datasets, it is also much smaller than the total number of features, $K \ll D_1 + D_2$. Hence, in fact, K can be dropped from the asymptotic complexity notation, which yields the following result.

Corollary 1. *Algorithm 1 scales with the data size as $O(PD)$, where P is the number of instances and $D = D_1 + D_2$ is the total number of features.*

C. MSE Performance

In this section, assuming the Gaussian and multinomial generative models are consistent with the observations (i.e., there is no model mismatch), we numerically compare the mean squared error (MSE), $\mathsf{E}[\|\boldsymbol{c}_i - \boldsymbol{c}_i\|^2]$, of Algorithm 1 with the Cramér-Rao lower bound (CRLB). With $\boldsymbol{u}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, the distribution of the Gaussian observations is $y_{ij} \sim \mathcal{N}(\boldsymbol{c}_i^T\boldsymbol{\mu}_j, \boldsymbol{c}_i^T\boldsymbol{\Sigma}_j\boldsymbol{c}_i + \sigma_{ij}^2)$. It is straightforward to show that the Fisher information matrix of the Gaussian model is given by [41, p. 47]

$$\mathcal{F}_g(\boldsymbol{c}_i) = \sum_{j=1}^{P} \frac{\boldsymbol{\mu}_j\boldsymbol{\mu}_j^T}{\boldsymbol{c}_i^T\boldsymbol{\Sigma}_j\boldsymbol{c}_i + \sigma_{ij}^2} + 2\frac{\boldsymbol{\Sigma}_j\boldsymbol{c}_i\boldsymbol{c}_i^T\boldsymbol{\Sigma}_j}{(\boldsymbol{c}_i^T\boldsymbol{\Sigma}_j\boldsymbol{c}_i + \sigma_{ij}^2)^2},$$

where the first and second terms are the contributions from the mean and variance, respectively. On the other hand, in the multinomial model, due to the sum-exp term in the denominator of the likelihood, the Fisher information matrix is not tractable. Thus, we resort to numerical computation via Monte Carlo simulations as described next.

Proposition 2. *The Fisher information matrix $\mathcal{F}_m(\boldsymbol{c}_i)$ for the multinomial model is given by*

$$\mathcal{F}_m(\boldsymbol{c}_i) = \mathsf{E}_{\boldsymbol{z}_i}\left[\frac{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p'(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}\left(\frac{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p'(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}\right)^T\right] \tag{16}$$

where

$$p'(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i) = \frac{\partial}{\partial \boldsymbol{c}_i} p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i) = p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i) \sum_{j=1}^{D_2-1} (z_{ij} - N_i p_{ij})\, \boldsymbol{v}_j,$$

and $p_{ij} = \frac{e^{\boldsymbol{c}_i^T\boldsymbol{v}_j}}{1+\sum_{j'=1}^{D_2-1} e^{\boldsymbol{c}_i^T\boldsymbol{v}_{j'}}}$ is the probability of category j for instance i.

Proof: The nontrivial part in the proof is justifying changing the order of expectation and differentiation. From the definition of Fisher information we have

$$\begin{aligned}\mathcal{F}_m(\boldsymbol{c}_i) &= \mathsf{E}_{\boldsymbol{z}_i}\left[\frac{\partial}{\partial \boldsymbol{c}_i}\log p(\boldsymbol{z}_i|\boldsymbol{c}_i)\left(\frac{\partial}{\partial \boldsymbol{c}_i}\log p(\boldsymbol{z}_i|\boldsymbol{c}_i)\right)^T\right] \\ &= \mathsf{E}_{\boldsymbol{z}_i}\left[\frac{\frac{\partial}{\partial \boldsymbol{c}_i}p(\boldsymbol{z}_i|\boldsymbol{c}_i)}{p(\boldsymbol{z}_i|\boldsymbol{c}_i)}\left(\frac{\frac{\partial}{\partial \boldsymbol{c}_i}p(\boldsymbol{z}_i|\boldsymbol{c}_i)}{p(\boldsymbol{z}_i|\boldsymbol{c}_i)}\right)^T\right] \\ &= \mathsf{E}_{\boldsymbol{z}_i}\left[\frac{\frac{\partial}{\partial \boldsymbol{c}_i}\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}\right. \\ &\qquad \left.\left(\frac{\frac{\partial}{\partial \boldsymbol{c}_i}\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}{\mathsf{E}_{\{\boldsymbol{v}_j\}}\left[p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)\right]}\right)^T\right],\end{aligned}$$

where the differentiation can be brought into the expectation due to the Dominated Convergence Theorem [42, p. 53] since expectation and differentiation are both limits, and $p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)$ is a probability dominated by 1. The derivative $p'(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i)$ directly follows from the likelihood

$$p(\boldsymbol{z}_i|\{\boldsymbol{v}_j\}, \boldsymbol{c}_i) = \frac{N_i!}{z_{i1}!\cdots z_{iD_2}!}\frac{e^{\boldsymbol{c}_i^T\sum_{j=1}^{D_2-1} z_{ij}\boldsymbol{v}_j}}{\left(1+\sum_{j'=1}^{D_2-1} e^{\boldsymbol{c}_i^T\boldsymbol{v}_{j'}}\right)^{N_i}}.$$

■

The Fisher information expression given in (16) can be efficiently computed through Monte Carlo simulations, as shown in Algorithm 2. In Algorithm 2, a simplified notation is used by dropping some indices: $\boldsymbol{c}$ is the coefficient vector to be estimated, N is the number of multinomial experiments, D is the number of categories, R is the number of realizations to be averaged over, and K is the number of factors.

The overall Fisher information of the multimodal model and CRLB are given by

$$\mathcal{F}(\boldsymbol{c}_i) = \mathcal{F}_g(\boldsymbol{c}_i) + \mathcal{F}_m(\boldsymbol{c}_i)$$
$$\mathsf{E}[\|\boldsymbol{c}_i - \boldsymbol{c}_i\|^2] \geq \mathrm{trace}(\mathcal{F}(\boldsymbol{c}_i)^{-1}).$$

Algorithm 2 Monte Carlo simulations for multinomial Fisher information

```
Input $\boldsymbol{c}, N, D, R, K$
Generate factor score matrices $\{\boldsymbol{V}_r\}_{r=1,\ldots,R}$ with columns $\{\boldsymbol{v}_{rd} \sim \mathcal{N}(\boldsymbol{0}_K, \boldsymbol{I}_K)\}_{d=1,\ldots,D-1}$
Compute probability vectors $\left\{\boldsymbol{p}_r = \left[e^{\boldsymbol{c}^T\boldsymbol{v}_{r1}}/(1+\sum_{d=1}^{D-1} e^{\boldsymbol{c}^T\boldsymbol{v}_{rd}}) \cdots 1/(1+\sum_{d=1}^{D-1} e^{\boldsymbol{c}^T\boldsymbol{v}_{rd}})\right]\right\}_i$
for r=1,...,R do
    Generate observation vector $\boldsymbol{z}_r \sim \mathcal{M}(N, \boldsymbol{p}_r)$
    Compute likelihoods $\boldsymbol{\ell}_r = [\ell_{r1} \cdots \ell_{rR}]^T$ with $\ell_{rs} = \mathcal{M}(\boldsymbol{z}_r | N, \boldsymbol{p}_s)$
    Compute average likelihood $\bar{\ell}_r = (\ell_{r1} + \cdots + \ell_{rR})/R$
    Compute matrix $\boldsymbol{\Lambda}_r = [\boldsymbol{\lambda}_{r1} \cdots \boldsymbol{\lambda}_{rR}]$ where $\boldsymbol{\lambda}_{rs} = \boldsymbol{V}_s(\boldsymbol{z}_r - N\boldsymbol{p}_s)$
    Compute average derivative $\bar{\boldsymbol{\ell}}'_r = \boldsymbol{\Lambda}_r \boldsymbol{\ell}_r / R$
end for
Compute $\mathcal{F}_m = \frac{1}{R}\sum_{r=1}^{R} \frac{\bar{\boldsymbol{\ell}}'_r \bar{\boldsymbol{\ell}}'^T_r}{\bar{\ell}_r^2}$
```

We next present simulation results for the MSE performance. In the simulated data, the number of Gaussian features and the number of multinomial categories are $D_1 = D_2 = 5$, the number of multinomial experiments is $N_i = 40$, and the number of instances is $P = 100$. In the MMFA algorithm, the number of factors is $K = 3$, ridge regression is used for updating $\boldsymbol{c}_i$ (see (15)) with weight 10^{-6}, and the hyperparameters $\alpha = 1$ and $\beta = 0.1$ are used for the inverse-gamma prior of σ_{ij}^2 (see (7)). The statistical expectation in the multinomial Fisher information is computed by averaging over $R = 2000$ realizations (see Algorithm 2). Fig. 3 shows that MSE of the proposed MMFA algorithm converges close to the CRLB (red dashed line) as early as in 20 iterations. The CRLB for Gaussian data, trace($\mathcal{F}_g(\boldsymbol{c}_i)^{-1}$), and multinomial data, trace($\mathcal{F}_m(\boldsymbol{c}_i)^{-1}$), are also shown in the same figure with black dotted line and purple dashed line, respectively.

Using the same simulation setup we show in Fig. 4 that under the MMFA model the likelihood of both the training ($P = 100$) and the unseen test data ($P = 10$) increase with the iterations and appear to converge to a limit. On the vertical axis, the likelihood of the multimodal data under the estimated model normalized by the likelihood under the true model is shown, i.e., $\frac{p(\{\boldsymbol{y}_i\}|\{\boldsymbol{c}_i, \boldsymbol{a}_j, \hat{\sigma}_{ij}^2\}) p(\{\boldsymbol{z}_i\}|\{\boldsymbol{c}_i\}, \boldsymbol{\Phi})}{p(\{\boldsymbol{y}_i\}|\{\boldsymbol{c}_i, \boldsymbol{u}_j, \sigma_{ij}^2\}) p(\{\boldsymbol{z}_i\}|\{\boldsymbol{c}_i, \boldsymbol{v}_j\})}$.

VI. Experiments

In this section, we will demonstrate the power of MMFA in large datasets for different tasks, such as generalization to unseen data, anomaly detection, data imputation, and recommender systems. We start with the New York City (NYC) Taxi dataset [43], and conclude with the MovieLens dataset. The codes used to produce the results in this paper are publicly available [1].

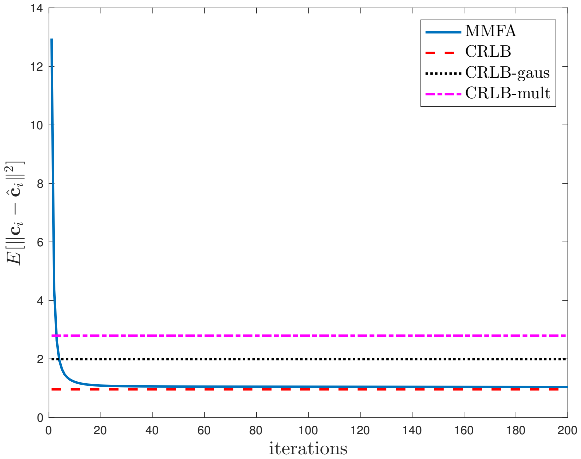


Fig. 3. MSE performance of the proposed variational EM algorithm. It converges to the multimodal bound CRLB quickly in 20 iterations. The Gaussian and multinomial components of CRLB are also shown.

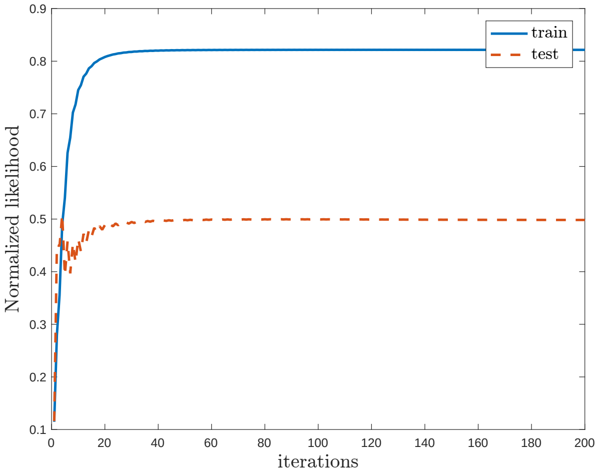
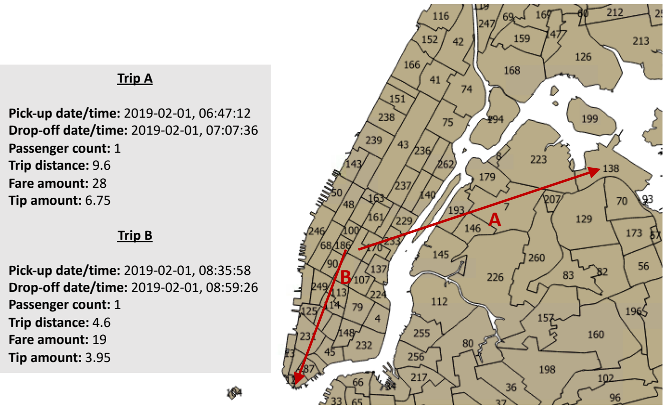


Fig. 4. Likelihood under the proposed model for training and test data normalized by the likelihood under the true model.

A. NYC Taxi Data

This dataset provides trip records for the yellow and green taxicabs, and the for-hire vehicles in NYC. Here we use the data from yellow taxis from February 2019 [43]. The dataset includes, for each recorded trip, the pick-up and drop-off dates, times and locations, trip distances, itemized fares, rate types, payment types, tip amounts, and passenger counts. The considered dataset has almost 7 million trip records. We first extracted a subset of the variables in the dataset, and filtered them to reduce bias. Specifically, we only considered trips with credit card payments since in most of the trips with cash payment the tip amount is unrealistically recorded as zero. We also disregarded trips that report fewer than 1 or more than 6 passengers (which is the legal limit). Location is reported in terms of the taxi zone id from 1 to 263. Unknown locations, denoted by the id 264 or 265, are ignored. Finally, we removed trips reporting a trip distance smaller than 0.1 mile or greater than 40 miles, and fare amounts less than

[1] https://github.com/maktukmak/MMFA



Trip A

Pick-up date/time: 2019-02-01, 06:47:12
Drop-off date/time: 2019-02-01, 07:07:36
Passenger count: 1
Trip distance: 9.6
Fare amount: 28
Tip amount: 6.75

Trip B

Pick-up date/time: 2019-02-01, 08:35:58
Drop-off date/time: 2019-02-01, 08:59:26
Passenger count: 1
Trip distance: 4.6
Fare amount: 19
Tip amount: 3.95

Fig. 5. Sample trips from the NYC Taxi dataset. Trip A is from Midtown Manhattan to LaGuardia Airport, and trip B is from Penn Station to Financial District.

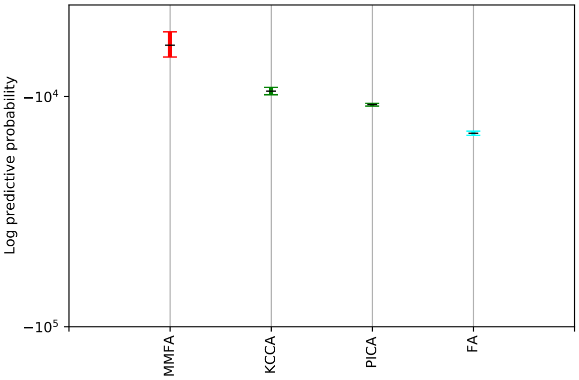


Fig. 6. Generalization performance comparison between the proposed MMFA algorithm and existing data fusion methods which treat categorical data as numerical. The mean and 95% confidence intervals are shown for log predictive likelihood. Higher values indicate higher generalization performance.

\$1 and more than \$200. After preprocessing, the data size decreased to around $P = 6.4 \times 10^6$. The considered features consist of numerical ones, namely the tip amount, fare amount, number of passengers, and trip distance, which are modeled using a four-dimensional Gaussian distribution ($D_1 = 4$), and categorical ones, namely the pick-up day ($D_{2,1} = 7$ choices), pick-up time ($D_{2,2} = 24$ choices), and location ($D_{2,3} = 263$ choices). Categorical distribution with one-of-K (a.k.a. one-hot) representation yields a total number of 298 features. For all categorical features, the number of trials is $N_i = 1, \forall i$.

Note that the numerical and categorical features arise from the same physical event (i.e., taxi trip), and hence they are dependent in general. For instance, it is seen that the tip amount, fare amount, and the number of passengers statistically depend on the pick-up and drop-off locations, e.g., trips from and to Manhattan statistically have higher tip percentages. Two sample trips are shown on the NYC map in Fig. 5.

Generalization performance: For generative models a common performance metric, the predictive log-likelihood value, is often used to assess the ability of the model to generalize to unseen data in training [31]. The predictive likelihood value is computed for an unseen data instance as follows. The trained MMFA model produces posterior distributions from estimated model parameters obtained using successive E and M steps $\{(\text{M-step})_m\}_{m=1}^{M}$ shown in Fig. 2. The final M-step, $(\text{M-step})_c$ in Fig. 2, uses these parameters to compute the factor score vector $\boldsymbol{c}_i$ associated with an unseen data instance $\boldsymbol{x}_i$, which specifies the likelihood function through the generative model shown in Fig. 1.

We compare the proposed MMFA algorithm with Kernel Canonical Correlation Analysis (KCCA) [56], Parallel Independent Component Analysis (PICA) [19], [20], and the standard factor analysis (FA) method [38] on the NYC Taxi dataset. As compared to standard CCA, KCCA algorithm enables fusion of data with multiple modalities by choosing proper kernels. Here, we use a linear kernel for the numerical modalities to form the symmetric Gram matrix. For the categorical data, we compute the Gram matrix by using the Hamming distances between observations, which is a proper similarity measure for the binary-coded categorical variables. Canonical components are then computed by projecting the Gram matrices to a lower dimensional latent space through demixing matrices for each modality. The KCCA objective maximizes the correlation of each of these canonical components across the modalities. This is usually performed by solving a generalized eigenvalue problem. On the other hand, PICA assumes non-Gaussian independent components for each modality. Here, we use a non-quadratic exponential decay function for the log of the probability distribution functions of the components as suggested in [57]. The modalities are correlated through the mixing matrices instead of the latent components. To implement the cross-modality optimization in PICA, a term is added to the PICA objective that encourages maximization of the correlation between the components of the matrices. Centering and whitening are applied in advance as a preprocessing step. Conversely, FA assumes spherical Gaussian components for the latent variables, i.e., factor loading coefficients. It is a generalization of PCA, where the noise covariance matrix is diagonal and has D free parameters. Also, the orthonormality constraint on the factor loading matrix is relaxed in FA models. The model is usually fit by using the EM algorithm, where the E-step computes posterior distributions of the latent variables, and the M-step computes point estimate for the factor loading matrix. We adopted the FA model as a representative of the class of fusion algorithms that treat all data modalities as numerical by concatenating them to form a long feature vector in a linear model.

MMFA achieves fusion by generating common latent factor scores to generate both modalities under a Bayesian graphical model. Under this model, the data modalities are conditionally independent given the factor loading coefficients, allowing modality-specific probabilistic models to be fused together. As generalized linear models for a large variety of data types and distributions are available (Table I), the MMFA can be applied to a wide range of data types beyond the Gaussian/multinomial case considered here. In contrast, Kernel CCA fuses data types by performing an eigendecomposition on a distance matrix or, equivalently, a similarity matrix. Instead of incorporating explicit probability distribution models for different data types, as in MMFA, KCCA accounts for different data types through transformation of the similarity matrix via kernelization, where

TABLE II
SOME ANOMALIES FOUND BY MMFA IN THE TEST SET.

Location	Day	Hour	Tip	Fare	Passenger
132	6	4	90.00	46.00	1
66	2	10	29.30	26.00	1
91	2	7	0.00	59.00	1
237	2	23	2.70	59.50	1
79	5	1	5.00	63.50	2

the type of kernel is selected to match the data type. As KCCA performs an eigendecomposition of a data similarity matrix its computational complexity is of order $O(P^3)$ as compared to only order $O(PD)$ for FA and MMFA. On the other hand, PICA jointly models the modalities by maximizing the correlation between the components of the mixing matrices. If we denote dimension of the two modalities D_1 and D_2, there are $D_1 \times D_2$ possible pairs to compute the correlation. Indeed, the algorithm chooses single pair at each iteration, which has the maximum correlation as compared to the other pairs. Hence, it is not straightforward how to extend this model to more than two modalities in an efficient way. Also, note that the categorical data is still modeled as numerical data.

We compute the log predictive marginal likelihood on the test set to compare the models. To this end, the latent variables are integrated out. Particularly, for MMFA, the Gaussian parameters $\mathbf{u}_j$ and multinomial parameters $\mathbf{v}_j$ are integrated out. For FA, PICA and KCCA, the latent variables are assigned per data point as opposed to MMFA, which are integrated out in a closed form. Note that KCCA likelihoods are computed on the higher dimensional kernel space instead of the observation space.

A train-validate-test split with $(0.6, 0.2, 0.2)$ ratio is applied to the data. Using BIC, the best number of latent dimensions is found to be 10 for MMFA ($K = 10$), 6 for FA, 8 for KCCA. The BIC value $k \log(P) - 2\log(p(\boldsymbol{X}))$ penalizes the negative log-likelihood score with the number of model parameters k, hence the K value with the smallest BIC value is selected for each algorithm. Figure 6 shows, for MMFA, FA, and KCCA, the mean and 95% confidence interval of the log predictive marginal likelihood values. The random train-validate-test split was repeated 20 times to compute the mean and confidence interval of the log predictive likelihoods. The mean values for MMFA, KCCA, PICA, and FA are -5982, -9471, -10833, -14429, respectively. By modeling the categorical data appropriately and fusing it with numerical data with probabilistic models, MMFA achieves much better generalization performance compared to other data fusion techniques that treat categorical data the same way as they do with numerical data.

Anomaly detection: We next demonstrate the anomaly detection performance of MMFA on the NYC taxi data. We first fit the MMFA model on the training set, and then sort the likelihoods of instances in the validation set with respect to the trained model. Finally, in the test, we compare the likelihood of each instance with the likelihoods of validation set, and declare anomalous if it is smaller than the $(1-\delta)\%$ of the validation likelihoods, where δ is a small number representing the statistical significance level. For $\delta = 0.05$, the top five anomalies with the smallest likelihoods are shown in Table II. The first three anomalies are obvious as the tip/fare ratio is unexpectedly high or low. However, the last two anomalies in the table can be considered as interesting findings of the MMFA model. The reason why they appear among the top anomalies is not only the tip/fare ratio, but in fact the inconsistency between the location and the tip percentage, 4.5% and 7.9%, respectively. Their locations are both in Manhattan with tip percentage mean and standard deviation of $(27.3\%, 10.1\%)$ for location 237 and $(25.2\%, 10.4\%)$ for location 79. While these trips are detected by MMFA as anomalous with tip percentages 4.5% and 7.9%, there are other trips with smaller tip percentages that are deemed nominal in locations with smaller mean tip percentages, e.g., 5.1% in location 132 (Queens) where mean and standard deviation is $(19\%, 10.7\%)$. Since there is no ground truth (i.e., nominal and anomalous labels) in the dataset, such comparative evaluation is useful in showing MMFA's success in anomaly detection. Note also that MMFA is a completely unsupervised algorithm.

B. MovieLens Data

Our next application is recommender systems, in which the objective is to learn user patterns and provide successful item recommendations to the users. While the commonly used collaborative filtering techniques in recommender systems typically use only the interactions between the users and items to learn the user patterns, there are also hybrid methods that combine user-item interactions with side information, such as user demographics and item features, for better recommendation performance [47]. However, the existing hybrid methods mainly convert the categorical side information, such as gender, occupation, item genre, country, etc., to numerical data for fusing multimodal data.

The MovieLens dataset, which is commonly used as a benchmark dataset in recommender system applications, has three different versions, 100K, 1M, and 10M, in terms of the number of interactions between the users and the movies, i.e., user ratings for items. To show the scalability of the proposed MMFA algorithm, we use the MovieLens-10M dataset, which has a little more than 10M ratings from 71567 users to 10681 items. The size and sparsity of this dataset, where 98.7% of 71567×10681 possible interactions is missing, brings significant challenges. In addition to the ratings, two item side information, release date and genre, are available in the dataset. We model release date using a univariate Gaussian distribution, and each of 21 genre categories using a Bernoulli (i.e., binary categorical) distribution since an item can have multiple genres.

We train the MMFA model on heterogeneous data from $P = 10681$ items, consisting of numerical ratings from $D_{1,1} = 71567$ users, release date ($D_{1,2} = 1$), and categorical genre information ($D_{2,j} = 2, j = 1, \dots, 21$ with $N_{i,j} = 1$ trial $\forall i, j$), to find the latent vector $\boldsymbol{c}_i$ for each item i. The number of factors K is chosen as 10 using BIC. In latent variable models for collaborative filtering, such as probabilistic matrix factorization (PMF), the rating r_{ij} of item i from user j is commonly modeled using a Gaussian distribution $\mathcal{N}(\boldsymbol{c}_i^T \boldsymbol{u}_j, \sigma^2)$,

TABLE III
MSE ON MOVIELENS-10M DATASET WITH WARM-START.

	SVDpp	NMF	PMF	BPMF	**MMFA**
MSE	0.659	0.766	0.691	0.671	**0.632**

where $\boldsymbol{u}_j$ is a latent factor score vector representing user j, and σ^2 is the variance parameter.

Following the recommender systems literature we consider two experiment setups called warm start and cold start. In warm start, each user or each item has at least one rating in the training set, whereas in cold start, for some users or items the recommender system has to completely rely on side information as there was no related rating in training. For warm start, $60\% - 20\% - 20\%$ training-validation-test split is used for the ratings of each item. On the other hand, for cold start, all the ratings of 20% of the items are used in the test set, and similarly 20% in the validation set.

Data imputation: We first evaluate MMFA for predicting ratings in terms of MSE, $\frac{1}{|\mathcal{O}|}\sum_{i,j\in\mathcal{O}}(r_{ij} - \boldsymbol{c}_{i,n}^T\boldsymbol{a}_{j,n})^2$, where $\boldsymbol{a}_{j,n}$ is the posterior mean of $\boldsymbol{u}_j$ (see (6)), $\mathcal{O}$ is the set of observed ratings in the test and $|\mathcal{O}|$ is its cardinality. MMFA is compared with several collaborative filtering algorithms, namely SVDpp, NMF, PMF, and BPMF. Specifically, the SVDpp method [49] performs a matrix factorization on the rating matrix including implicit ratings to find the user and item matrices. The nonnegative matrix factorization (NMF) algorithm, similar to the singular value decomposition (SVD), computes a matrix factorization, but by enforcing both user and item matrices to be nonnegative [50]. PMF also applies a matrix factorization on the rating matrix by assuming Gaussian latent variables for both users and items [48]. In Bayesian PMF (BPMF), additional prior distributions are assumed for the hyperparameters of user and item latent variables [51]. All of the four benchmark models use only the rating matrix without any side information. Leveraging item side information MMFA has a clear advantage over them in the cold-start setting, hence we only compare the algorithms in the warm-start setting by averaging over 10 experiments with random training-validation-test split. As shown in Table III, MMFA achieves the best MSE performance also in the warm-start setting by utilizing the item side information available in the dataset.

Recommendation accuracy: Finally, we evaluate MMFA's performance in terms of the accuracy of recommended movies to the users. For movie datasets, recall (i.e., true positive rate) is computed as the ratio "number of movies user liked in the recommendations / total number of movies user liked". In Table IV, for 10 recommendations, we report the recall averaged over all test users and 10 different experiments with random splits in both warm- and cold-start settings. Here we compare the proposed MMFA approach with state-of-the-art recommender systems that are capable of utilizing side information. Among the considered state-of-the-art methods, LCE [52], DecRec [53], and KMF [54] could not scale well to the MovieLens-10M dataset due to the memory limitations. These algorithms store similarity matrices for users and items based on the side information, causing $O(P^2{+}D^2)$ space complexity, where P and D are the number users and items. Moreover, the KMF algorithm inverts such matrices, increasing its space complexity to $O(P^3 + D^3)$. In the MovieLens-10M dataset, only items have side information, and $D = 10681$. On the other hand, LightFM [55], which incorporates the side information into the rating matrix and performs matrix factorization on the enhanced data in a non-probabilistic way, scales well to the MovieLens-10M data. The superior performance of MMFA can be attributed to its natural handling of data fusion through appropriate probabilistic models while LightFM embeds categorical features into numerical values for data fusion.

TABLE IV
RECALL WITH 10 RECOMMENDATIONS ON MOVIELENS-10M DATASET WITH WARM- AND COLD-START.

Recall	LightFM	**MMFA**
Warm-start	0.886	**0.902**
Cold-start	0.824	**0.886**

VII. CONCLUSION

A general unsupervised Bayesian framework based on the exponential family was proposed for the joint analysis of heterogeneous datasets. The proposed model, called Multimodal Factor Analysis (MMFA), uses the most appropriate probability distribution from the exponential family for each data modality, and fuses them by modeling their natural parameters through a common latent vector for each instance. To fit the model on large high-dimensional datasets, we proposed a computationally efficient variational Expectation-Maximization (EM) algorithm, which scales linearly with the number of features and the number of instances. On the common real-valued and categorical data combination, we showed that the algorithm quickly converges to the Cramer-Rao Lower Bound (CRLB) when there is no model mismatch. The proposed algorithm was also evaluated on two high-dimensional and heterogeneous datasets, the NYC Taxi dataset and the MovieLens-10M dataset, for various machine learning tasks. Specifically, the experiments demonstrated that the proposed MMFA model generalizes to unseen data better than the state-of-the-art data fusion techniques such as KCCA and PICA, provides meaningful anomaly detection results, predicts missing data better than the collaborative filtering techniques, and gives more accurate recommendations than the state-of-the-art recommender systems. We should note here that despite our efforts for a fair comparison between algorithms, the benchmark algorithms could possibly be further optimized to improve their performances. As future work, we plan to investigate (i) deep versions of MMFA which fuses different modalities in lower levels of hyper-parameters in a hierarchical Bayesian setup, (ii) links and comparisons with generative neural network models, such as Variational Autoencoders, Restricted Boltzmann Machines (RBM), Generative Adversarial Networks (GAN), and (iii) stochastic optimization methods for variational EM to improve further the memory complexity for extremely large datasets.

Appendix

Proof of Proposition 1:

For notational simplicity, we will drop the iteration index n in the E-step. Defining the diagonal matrix $\boldsymbol{N} = \text{diag}(N_1, \ldots, N_P)$ we start with manipulating $\boldsymbol{\Omega}$ using (9) and (11),

$$\begin{aligned}\boldsymbol{\Omega} &= \left(\sum_{i=1}^{P} N_i \boldsymbol{C}_i \boldsymbol{A} \boldsymbol{C}_i^T + \boldsymbol{I}_{(D_2-1)K}\right)^{-1} \\ &= \left(\boldsymbol{A} \otimes \boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^T + \boldsymbol{I}_{(D_2-1)K}\right)^{-1} \\ &= \Big[\boldsymbol{I}_{D_2-1} \otimes \Big(\frac{1}{2}\boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^T + \boldsymbol{I}_K\Big) \\ &\quad + \Big(\boldsymbol{1}_{D_2-1} \otimes \boldsymbol{C}\Big)\Big(-\frac{\boldsymbol{N}}{2D_2}\Big)\Big(\boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{C}^T\Big)\Big]^{-1}.\end{aligned}$$

Using the Matrix Inversion Lemma we can write

$$\begin{aligned}\boldsymbol{\Omega} &= \boldsymbol{I}_{D_2-1} \otimes \boldsymbol{F}^{-1} - \boldsymbol{1}_{D_2-1} \otimes \boldsymbol{F}^{-1}\boldsymbol{C} \\ &\quad \underbrace{\Big[-2D_2\boldsymbol{N}^{-1} + (D_2-1)\boldsymbol{C}^T\boldsymbol{F}^{-1}\boldsymbol{C}\Big]^{-1}}_{\boldsymbol{\Gamma}} \boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{C}^T\boldsymbol{F}^{-1} \\ &= \boldsymbol{I}_{D_2-1} \otimes \boldsymbol{F}^{-1} - \boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{F}^{-1}\boldsymbol{C}\boldsymbol{\Gamma}\boldsymbol{C}^T\boldsymbol{F}^{-1},\end{aligned}$$

where $\boldsymbol{F} = \frac{1}{2}\boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^T + \boldsymbol{I}_K$. Using again the Matrix Inversion Lemma for $\boldsymbol{\Gamma}$ we obtain

$$\boldsymbol{\Omega} = \boldsymbol{I}_{D_2-1} \otimes \boldsymbol{F}^{-1} + \boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{\Delta},$$

where $\boldsymbol{\Delta} = \boldsymbol{F}^{-1}\boldsymbol{G}\Big[\boldsymbol{I}_K + (D_2-1)\left(\boldsymbol{G} + \boldsymbol{I}_K\right)^{-1}\boldsymbol{G}\Big]\boldsymbol{F}^{-1}$ and $\boldsymbol{G} = \frac{1}{2D_2}\boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^T = \frac{\boldsymbol{F}-\boldsymbol{I}_K}{D_2}$.

We continue with putting the mean $\boldsymbol{\omega}$ of the approximate posterior in a compact and computationally efficient form.

$$\begin{aligned}\boldsymbol{\omega} &= \boldsymbol{\Omega}\sum_{i=1}^{P}\boldsymbol{C}_i\widetilde{\boldsymbol{z}}_i = \boldsymbol{\Omega}\left[\sum_{i=1}^{P}\widetilde{z}_{i1}\boldsymbol{c}_i^T \cdots \sum_{i=1}^{P}\widetilde{z}_{iD_2-1}\boldsymbol{c}_i^T\right]^T \\ &= \boldsymbol{\Omega}\left[(\boldsymbol{C}\boldsymbol{\xi}_1)^T \cdots (\boldsymbol{C}\boldsymbol{\xi}_{D_2-1})^T\right]^T \\ &= \left[\left(\boldsymbol{F}^{-1}\boldsymbol{C}\boldsymbol{\xi}_1 + \boldsymbol{\Delta}\boldsymbol{C}\sum_{j=1}^{D_2-1}\boldsymbol{\xi}_j\right)^T \cdots \right. \\ &\qquad\qquad \left.\left(\boldsymbol{F}^{-1}\boldsymbol{C}\boldsymbol{\xi}_{D_2-1} + \boldsymbol{\Delta}\boldsymbol{C}\sum_{j=1}^{D_2-1}\boldsymbol{\xi}_j\right)^T\right]^T,\end{aligned}$$

where $\boldsymbol{\xi}_j = [\widetilde{z}_{1j} \cdots \widetilde{z}_{Pj}]^T, j = 1, \ldots, D_2 - 1$, and we used the $\boldsymbol{\Omega}$ expression derived above. Reorganizing the vector $\boldsymbol{\omega} = [\boldsymbol{\phi}_1^T \cdots \boldsymbol{\phi}_{D_2-1}^T]^T$ as the matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{D_2-1}]$, in a more compact form, we can write

$$\boldsymbol{\Phi} = \boldsymbol{F}^{-1}\boldsymbol{C}\widetilde{\boldsymbol{Z}} + \boldsymbol{\Delta}\boldsymbol{C}(\widetilde{\boldsymbol{Z}}\boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T),$$

where $\widetilde{\boldsymbol{Z}} = [\boldsymbol{\xi}_1 \cdots \boldsymbol{\xi}_{D_2-1}] = [\widetilde{\boldsymbol{z}}_1 \cdots \widetilde{\boldsymbol{z}}_P]^T$.

In the M-step, the update equation for $\boldsymbol{\psi}_i$ at iteration n is found from (9) as,

$$\begin{aligned}\boldsymbol{\psi}_{i,n} &= \arg\max_{\boldsymbol{\psi}_i} \mathsf{E}_{\mathcal{N}(\boldsymbol{v}|\boldsymbol{\omega},\boldsymbol{\Omega})}\Big[-\frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T\boldsymbol{A}(\boldsymbol{\eta}_i - \boldsymbol{\psi}_i) \\ &\qquad - (\boldsymbol{\eta}_i - \boldsymbol{\psi}_i)^T\nabla\text{lse}(\boldsymbol{\psi}_i) - \text{lse}(\boldsymbol{\psi}_i)\Big] \\ 0 &= \mathsf{E}_{\mathcal{N}(\boldsymbol{v}|\boldsymbol{\omega},\boldsymbol{\Omega})}\Big[\boldsymbol{A}(\boldsymbol{V}^T\boldsymbol{c}_i - \boldsymbol{\psi}_{i,n}) + \nabla\text{lse}(\boldsymbol{\psi}_i) - \nabla\text{lse}(\boldsymbol{\psi}_i)\Big] \\ \boldsymbol{\psi}_{i,n} &= \boldsymbol{\Phi}_n^T\boldsymbol{c}_{i,n-1}\end{aligned}$$

For the update of the coefficient vector estimate $\boldsymbol{c}_{i,n}$ (cf. (8)), from (10), the part related to the multinomial model is given by

$$\begin{aligned}\zeta_n(\boldsymbol{c}_i) &= -\frac{N_i}{2}\mathsf{E}_{\mathcal{N}(\boldsymbol{v}|\boldsymbol{\omega},\boldsymbol{\Omega})}\Big[\boldsymbol{v}^T\boldsymbol{C}_i\boldsymbol{A}\boldsymbol{C}_i^T\boldsymbol{v}\Big] + \boldsymbol{\omega}_n^T\boldsymbol{C}_i\widetilde{\boldsymbol{z}}_i \\ &= -\frac{N_i}{2}\mathsf{Tr}\left(\boldsymbol{C}_i\boldsymbol{A}\boldsymbol{C}_i^T(\boldsymbol{\Omega}_n + \boldsymbol{\omega}_n\boldsymbol{\omega}_n^T)\right) + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i \\ &= -\frac{N_i}{2}\mathsf{Tr}\left((\boldsymbol{I}_{D_2-1} \otimes \boldsymbol{c}_i)\boldsymbol{A}(\boldsymbol{I}_{D_2-1} \otimes \boldsymbol{c}_i^T)(\boldsymbol{\Omega}_n + \boldsymbol{\omega}_n\boldsymbol{\omega}_n^T)\right) \\ &\quad + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i \\ &= -\frac{N_i}{2}\mathsf{Tr}\left((\boldsymbol{A} \otimes \boldsymbol{c}_i\boldsymbol{c}_i^T)(\boldsymbol{\Omega}_n + \boldsymbol{\omega}_n\boldsymbol{\omega}_n^T)\right) + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i \\ &= -\frac{N_i}{2}\mathsf{Tr}\left(\boldsymbol{A} \otimes \boldsymbol{c}_i\boldsymbol{c}_i^T\boldsymbol{F}_n^{-1} + \boldsymbol{A}\boldsymbol{1}_{D_2-1}\boldsymbol{1}_{D_2-1}^T \otimes \boldsymbol{c}_i\boldsymbol{c}_i^T\boldsymbol{\Delta}_n\right) \\ &\quad - \frac{N_i}{2}\boldsymbol{\omega}_n^T(\boldsymbol{A} \otimes \boldsymbol{c}_i\boldsymbol{c}_i^T)\boldsymbol{\omega}_n + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i \\ &= -\frac{N_i}{2}\boldsymbol{c}_i^T\left(\mathsf{Tr}(\boldsymbol{A})\boldsymbol{F}_n^{-1} + \mathsf{Tr}(\boldsymbol{A}\boldsymbol{1}\boldsymbol{1}^T)\boldsymbol{\Delta}_n\right)\boldsymbol{c}_i \\ &\quad - \frac{N_i}{2}\sum_{j=1}^{D_2-1}\boldsymbol{\omega}_{j,n}^T\boldsymbol{c}_i\boldsymbol{c}_i^T\boldsymbol{\omega}_{j,n} + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i \\ &\quad - \frac{N_i}{2D_2}\sum_{j=1}^{D_2-1}\sum_{j'=1}^{D_2-1}\boldsymbol{\omega}_{j,n}^T\boldsymbol{c}_i\boldsymbol{c}_i^T\boldsymbol{\omega}_{j',n} \\ &= -\frac{N_i}{2}\boldsymbol{c}_i^T\Big(\frac{(D_2-1)^2}{2D_2}\boldsymbol{F}_n^{-1} + \frac{D_2-1}{2D_2}\boldsymbol{\Delta}_n + \frac{1}{2}\boldsymbol{\Phi}_n\boldsymbol{\Phi}_n^T \\ &\quad - \frac{1}{2D_2}(\boldsymbol{\Phi}_n\boldsymbol{1})(\boldsymbol{\Phi}_n\boldsymbol{1})^T\Big)\boldsymbol{c}_i^T + \boldsymbol{c}_i^T\boldsymbol{\Phi}_n\widetilde{\boldsymbol{z}}_i,\end{aligned}$$

which concludes the proof.

Proof of Theorem 1:

In the Gaussian E-step (line 4 in Algorithm 2), the most expensive computations are $\boldsymbol{C}\boldsymbol{\Sigma}_j^{-1}\boldsymbol{C}^T$ and $\boldsymbol{B}_j\boldsymbol{C}$ for each feature j, resulting in $O(K^2PD_1)$ computations. The matrix inversion for all j is $O(K^3D_1)$, but this is cheaper than $O(K^2PD_1)$ since $K \ll P$. Similarly in the Gaussian M-step (line 5), computing $\boldsymbol{c}_i^T\boldsymbol{B}_j$ for each i, j pair yields $O(K^2PD_1)$ complexity. In the multinomial model, both $\boldsymbol{C}\widetilde{\boldsymbol{Z}}$ in the E-step (line 6) and $\boldsymbol{\Phi}^T\boldsymbol{c}_i$, for $i = 1, \ldots, P$, in the M-step (line 7) have a complexity of $O(KPD_2)$, and the rest of the computations are cheaper. Finally, for the coefficients $\boldsymbol{c}_i$ (line 8) we solve a quadratic program for each i. In solving a possibly constrained quadratic program for each $\boldsymbol{c}_i$, the number of iterations, in practice, is bounded by a constant; and in each iteration, linear algebra operations in the K-dimensional space are performed. Hence, the overall complexity for solving the quadratic programs is $O(K^3P)$. Note also that each $\boldsymbol{c}_i$ can be updated in parallel. The computation of $\{\boldsymbol{H}_i\}$ and $\{\boldsymbol{\rho}_i\}$ have $O(K^2(K+P+D_2+PD_1))$ and $O(KP(D_1+D_2))$ complexity, respectively. Combining all the complexities we get $O(K^3P+K^2PD_1+KPD_2)$.

References

[1] K. Pearson, "Mathematical contributions to the theory of evolution. VIII. On the correlation of characters not quantitatively measurable", *Proceedings of the Royal Society of London*, vol. 66., no. 424-433, pp. 241–244, 1900.

[2] U. Olsson, F. Drasgow, and N.J. Dorans, "The Polyserial Correlation Coefficient", *Pcychometrika*, vol. 47, no. 3, pp. 337–347, Sep. 1982.

[3] S. Kolenikov, and G. Angeles, *The Use of Discrete Data in PCA: Theory, Simulations, and Applications to Socioeconomic Indices*, Chapel Hill: Carolina Population Center, University of North Carolina, 2004.

[4] S. Kolenikov, and G. Angeles, "Socioeconomic status measurement with discrete proxy variables: Is principal components analysis a reliable answer?", *The Review of Income and Wealth*, vol. 55, no. 1, pp. 128–165, Mar. 2009.

[5] M. Collins, S. Dasgupta, R.E. Schapire, "A generalization of principal components analysis to the exponential family", *Advances in neural information processing systems (NIPS)*, pp. 617–624, 2002.

[6] S. Mohamed, Z. Ghahramani, K.A. Heller, "Bayesian exponential family PCA", *Advances in neural information processing systems (NIPS)*, pp. 1089-1096, 2009.

[7] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, 2006.

[8] C. Spearman, ""General Intelligence," Objectively Determined and Measured" *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–292, Apr. 1904.

[9] F. Galton, *Hereditary genius: An inquiry into its laws and consequences*, Macmillan, vol. 27, 1869.

[10] R. Cudeck, and R.C. MacCallum, *Factor Analysis at 100: Historical Developments and Future Directions*, Lawrence Erlbaum Associates, Publishers, Mahwah, NJ, 2007.

[11] M.E. Khan, G. Bouchard, K.P. Murphy, and B.M. Marlin, "Variational bounds for mixed-data factor analysis", *Advances in neural information processing systems (NIPS)*, pp. 1108–1116, 2010.

[12] F.R. Bach, and M.I. Jordan, "A Probabilistic Interpretation of Canonical Correlation Analysis", *Technical Report, Department of Statistics, University of California, Berkeley*, 2005.

[13] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects", *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[14] T. Adali, Y. Levin-Schwartz, V.D. Calhoun, "Multimodal data fusion using source separation: Two effective models based on ICA and IVA and their properties", *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1478–1493, Sep. 2015.

[15] M. Belkin, and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", *Advances in neural information processing systems (NIPS)*, 2002.

[16] J.B. Tenenbaum, V. De Silva, J.C. De Silva, "A global geometric framework for nonlinear dimensionality reduction", *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[17] S. Sun, "A survey of multi-view machine learning", *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[18] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity", *International Conference on Machine Learning (ICML)*, pp. 352–360, 2013.

[19] J. Liu, G. Pearlson, A. Windemuth, G. Ruano, N.I. Perrone?Bizzozero, and V. Calhoun, "Combining fMRI and SNP data to investigate connections between brain function and genetics using parallel ICA", *Human brain mapping*, vol. 30, no. 1, pp. 241–255, Jan. 2009.

[20] G.D. Pearlson, J. Liu, and V.D. Calhoun, "An introductory review of parallel independent component analysis (p-ICA) and a guide to applying p-ICA to genetic data and imaging phenotypes to identify disease-associated biological pathways and systems in common complex disorders", *Frontiers in genetics*, vol. 6, no. 276, Sep. 2015.

[21] P. Ray, L. Zheng, J. Lucas, and L. Carin, "Bayesian joint analysis of heterogeneous genomics data", *Bioinformatics*, vol. 30, no. 10, pp. 1370–1376, 2014.

[22] E. Salazar, Y. Nikolova, W. Lian, P. Rai, A.L. Romer, A.R. Hariri, and L. Carin, "A Bayesian Framework for Multi-Modality Analysis of Mental Health", *submitted to Journal of the American Statistical Association (JASA).*

[23] E. Yang, G. Allen, Z. Liu, and P.K. Ravikumar, "Graphical models via generalized linear models", *Advances in neural information processing systems (NIPS)*, pp. 1358–1366, 2012.

[24] Z. Ghahramani, and G.E. Hinton, "The EM algorithm for mixtures of factor analyzers", *Technical Report CRG-TR-96-1, University of Toronto*, 1996.

[25] Z. Ghahramani, and M.J. Beal, "Variational inference for Bayesian mixtures of factor analysers", *Advances in neural information processing systems (NIPS)*, pp. 449–455, 2000.

[26] Y. Yilmaz, and A.O. Hero, "Multimodal Factor Analysis", *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015.

[27] Y. Yilmaz, and A.O. Hero, "Multimodal Event Detection in Twitter Hashtag Networks", *Journal of Signal Processing Systems*, vol. 90, no. 2, pp. 185-200, Feb. 2018.

[28] B. Jorgensen, "Exponential Dispersion Models", *Journal of the Royal Statistical Society B*, vol. 49, no. 2, pp. 127–162, 1987.

[29] D.J. Bartholomew, M. Knott, and I. Moustaki, *Latent Variable Models and Factor Analysis: A Unified Approach, 3rd Edition*, Wiley, Chichester, UK, 2011.

[30] M. Zhou, L.A. Hannah, D.B. Dunson, and L. Carin, "Beta-negative binomial process and Poisson factor analysis", *Journal of Machine Learning Research*, 2012.

[31] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation", *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan. 2003.

[32] P. McCullagh, and J.A. Nelder, *Generalized Linear Models, 2nd Edition*, CRC Press, Boca Raton, FL, 1989.

[33] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[34] C.F.J. Wu, "On the Convergence Properties of the EM Algorithm", *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, Mar. 1983.

[35] A.W. van der Vaart, *Asymptotic Statistics*, Cambridge University Press, Cambridge, UK, 2000.

[36] A. Belloni, and V. Chernozhukov, "Posterior inference in curved exponential families under increasing dimensions", *The Econometrics Journal.*, vol. 17, no. 2, pp. S75–S100, 2014.

[37] D. Böhning, and B.G. Lindsay, "Monotonicity of Quadratic-Approximation Algorithms", *Ann. Inst. Statist. Math.*, vol. 40, no. 4, pp. 641–663,1988.

[38] K.P.. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA, 2012.

[39] M. Lichman, *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]*, University of California, School of Information and Computer Science, Irvine, CA, 2013.

[40] D. Böhning, "Multinomial Logistic Regression Algorithm", *Ann. Inst. Statist. Math.*, vol. 44, no. 1, pp. 197–200,1992.

[41] S.M. Kay, *Fundamentals of statistical signal processing, volume i: Estimation theory*, PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.

[42] J. Jacod, and P. Protter, *Probability Essentials*, Springer-Verlag Berlin Heidelberg, 2004.

[43] NYC Taxi and Limousine Commission, *TLC Trip Data [http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml]*,

[44] V. D. Calhoun and T. Adali, "Feature-Based Fusion of Medical Imaging Data," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 711-720, Sept. 2009.

[45] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ICA to multivariate components" *International conference on independent component analysis and signal separation*, 2006, pp. 165–172.

[46] X.L. Li, and T. Adali, "Independent component analysis by entropy bound minimization" *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp.5151-5164, 2010.

[47] C. C. Aggarwal, et al., *Recommender systems*, Springer, 2016.

[48] A. Mnih, and R.R. Salakhutdinov, "Probabilistic matrix factorization", *Advances in neural information processing systems (NIPS)*, pp. 1257–1264, 2008.

[49] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model", *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434, 2008.

[50] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Inform.*, vol. 10, no. 2, pp. 1273–1284, 2014.

[51] R.R. Salakhutdinov, and A. Mnih, “Bayesian probabilistic matrix factorization using Markov chain Monte Carlo”, *International Conference on Machine Learning (ICML)*, pp. 880–887, 2008.

[52] M. Saveski, and A. Mantrach, “Item cold-start recommendations: learning local collective embeddings”, *ACM Conference on Recommender systems*, pp. 89–96, 2014.

[53] I. Barjasteh, R. Forsati, D. Ross, A.-H. Esfahanian, and H. Radha, “Cold-start recommendation with provable guarantees: a decoupled approach,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1462–1474, 2016.

[54] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, “Kernelized probabilistic matrix factorization: exploiting graphs and side information”, *SIAM International Conference on Data Mining*, pp. 403–414, 2012.

[55] M. Kula, “Metadata embeddings for user and item cold-start recommendations”, *CEUR Workshop Proceedings*, pp. 14–21, 2015.

[56] Bilenko, Natalia Y., and Jack L. Gallant, “Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging”, *Frontiers in neuroinformatics 10*, (2016): 49.

[57] Hyvrinen, Aapo, and Erkki Oja, “Independent component analysis: algorithms and applications”, *Neural networks 13.4-5*, (2000): 411-430.

Yasin Yilmaz (S’11-M’14-SM’20) received the Ph.D. degree in Electrical Engineering from Columbia University, New York, NY, in 2014. He is currently an Assistant Professor of Electrical Engineering at the University of South Florida, Tampa. His research intere ts include statistical signal processing, machine learning, and their applications to computer vision, cybersecurity, IoT networks, energy systems, transportation systems, and communication systems.

Mehmet Aktukmak received the B.S. degree in electrical and electronics engineering from Hacettepe University in 2009, the M.S. degree in electrical and electronics engineering from Middle East Technical University in 2012, and the Ph.D. degree in electrical engineering from University of South Florida in 2020. He is currently working as postdoctoral research fellow in electrical and computer engineering department at the University of Michigan. His research interests include multimodal-multitask learning, Bayesian modelling, variational inference, and their applications to image/video processing, matrix completion, meta learning, and recommender systems.

Alfred O. Hero III received the B.S. (summa cum laude) from Boston University (1980) and the Ph.D from Princeton University (1984), both in Electrical Engineering. Since 1984 he has been with the University of Michigan, Ann Arbor, where he is the John H. Holland Distinguished University Professor of Electrical Engineering and Computer Science and the R. Jamison and Betty Williams Professor of Engineering. His primary appointment is in the Department of Electrical Engineering and Computer Science and he also has appointments, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. He is a Section Editor of the SIAM Journal on Mathematics of Data Science and a Senior Editor of the IEEE Journal on Selected Topics in Signal Processing . He is on the editorial board of the Harvard Data Science Review (HDSR) and serves as moderator for the Electrical Engineering and Systems Science category of the arXiv . He was co-General Chair of the 2019 IEEE International Symposium on Information Theory (ISIT) and the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing. He was founding Co-Director of the Universitys Michigan Institute for Data Science (MIDAS) (2015-2018). From 2008-2013 he held the Digiteo Chaire dExcellence at the Ecole Superieure dElectricite, Gif-sur-Yvette, France. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the Society for Industrial and Applied Mathematics (SIAM). Several of his research articles have received best paper awards. Alfred Hero was awarded the University of Michigan Distinguished Faculty Achievement Award (2011), the Stephen S. Attwood Excellence in Engineering Award (2017), and the H. Scott Fogler Award for Professional Leadership and Service (2018). He received the IEEE Signal Processing Society Meritorious Service Award (1998), the IEEE Third Millenium Medal (2000), the IEEE Signal Processing Society Technical Achievement Award (2014), the Society Award from the IEEE Signal Processing Society (2015) and the Fourier Award from the IEEE (2020). Alfred Hero was President of the IEEE Signal Processing Society (2006-2008) and was on the IEEE Board of Directors (2009-2011) where he served as Director of Division IX (Signals and Applications). From 2011 to 2020 he was a member and Chair (2017-2020) of the Committee on Applied and Theoretical Statistics (CATS) of the US National Academies of Science. Alfred Heros recent research interests are in high dimensional spatio-temporal data, multi-modal data integration, statistical signal processing, and machine learning. Of particular interest are applications to social networks, network security and forensics, computer vision, and personalized health.