Relational Mixture of Experts: Explainable Demographics Prediction with Behavioral Data

Masafumi Oyamada
NEC Corporation
Kawasaki, Japan
Email: m-oyamada@cq.jp.nec.com

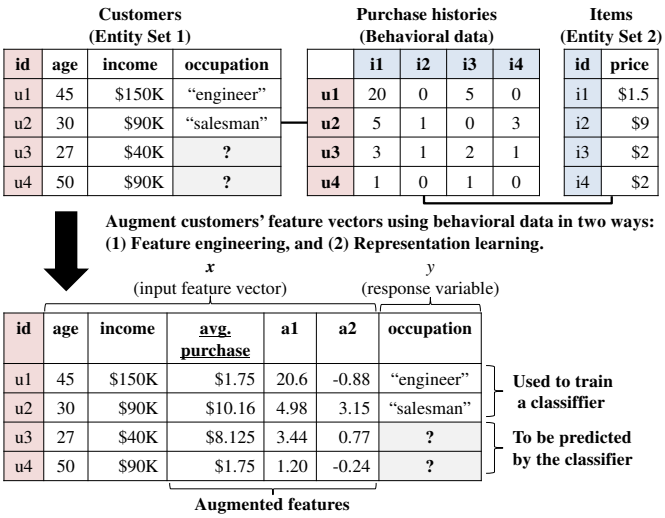
Shinji Nakadai
NEC Corporation
Kawasaki, Japan
Email: s-nakadai@az.jp.nec.com

***Abstract*—Given a collection of basic customer demographics (e.g., age and gender) and their behavioral data (e.g., item purchase histories), how can we predict sensitive demographics (e.g., income and occupation) that not every customer makes available?**

This demographics prediction problem is modeled as a classification task in which a customer's sensitive demographic y is predicted from his feature vector $\boldsymbol{x}$. So far, two lines of work have tried to produce a "good" feature vector $\boldsymbol{x}$ from the customer's behavioral data: (1) application-specific feature engineering using behavioral data and (2) representation learning (such as singular value decomposition or neural embedding) on behavioral data. Although these approaches successfully improve the predictive performance, (1) designing a good feature requires domain experts to make a great effort and (2) features obtained from representation learning are hard to interpret.

To overcome these problems, we present a *Relational Infinite Support Vector Machine* (R-iSVM), a mixture-of-experts model that can leverage behavioral data. Instead of augmenting the feature vectors of customers, R-iSVM uses behavioral data to find out behaviorally similar customer clusters and constructs a local prediction model at each customer cluster. In doing so, R-iSVM successfully improves the predictive performance without requiring application-specific feature designing and hard-to-interpret representations.

Experimental results on three real-world datasets demonstrate the predictive performance and interpretability of R-iSVM. Furthermore, R-iSVM can co-exist with previous demographics prediction methods to further improve their predictive performance.



id	age	income	occupation
u1	45	$150K	"engineer"
u2	30	$90K	"salesman"
u3	27	$40K	?
u4	50	$90K	?

	i1	i2	i3	i4
u1	20	0	5	0
u2	5	1	0	3
u3	3	1	2	1
u4	1	0	1	0

id	price
i1	$1.5
i2	$9
i3	$2
i4	$2

id	age	income	avg. purchase	a1	a2	occupation
u1	45	$150K	$1.75	20.6	-0.88	"engineer"
u2	30	$90K	$10.16	4.98	3.15	"salesman"
u3	27	$40K	$8.125	3.44	0.77	?
u4	50	$90K	$1.75	1.20	-0.24	?

Fig. 1. Conventional demographics prediction methods augment customer's feature vectors using "behavioral data" to improve the predictive performance in two ways: (1) application-specific feature engineering, in which domain experts aggregate behavioral data to produce a meaningful feature (average purchased item price is effective for predicting "occupation"), and (2) representation learning, such as Singular Value Decomposition, finds good feature vectors automatically (a1 and a2). Feature engineering requires considerable effort, and representation learning lacks interpretability. Our proposed model avoids these problems while achieving high predictive performance.

I. Introduction

Customer demographics, such as age and income, are essential information in various tasks including product planning, advertisement, and item recommendation [1]. Since not every customer makes available sensitive demographics like occupation, *customer demographics prediction* has received notable attention from both industry and academia [2], [3], [4], [5], [6], [7], [8].

Demographics prediction is conventionally formalized as a classification problem that predicts unknown demographics from known demographics [2], [3], [5], [7]. Because demographics have correlations like "older people tend to have large incomes (income and age positively correlate)," this formulation has achieved successes and been widely used. As shown in Figure 1, if "occupation" is unavailable for customers $u3$ and $u4$ but "age" and "income" are available for customers $u1$ to $u4$, we can learn a classifier using demographics of $u1$ and $u2$ as a training set. Then the classifier predicts "occupation" of customers $u3$ and $u4$ from their "age" and "income."

Customer's *behavioral data*, such as purchase histories and web browsing histories, has been actively used to predict demographics because just using demographic correlations sometimes results in poor predictive performance [2], [3], [4], [5], [6], [7], [8]. The underlying intuition of these studies is that behavioral data can describe demographics well, which is sometimes expressed in a more catchy phrase: "you are

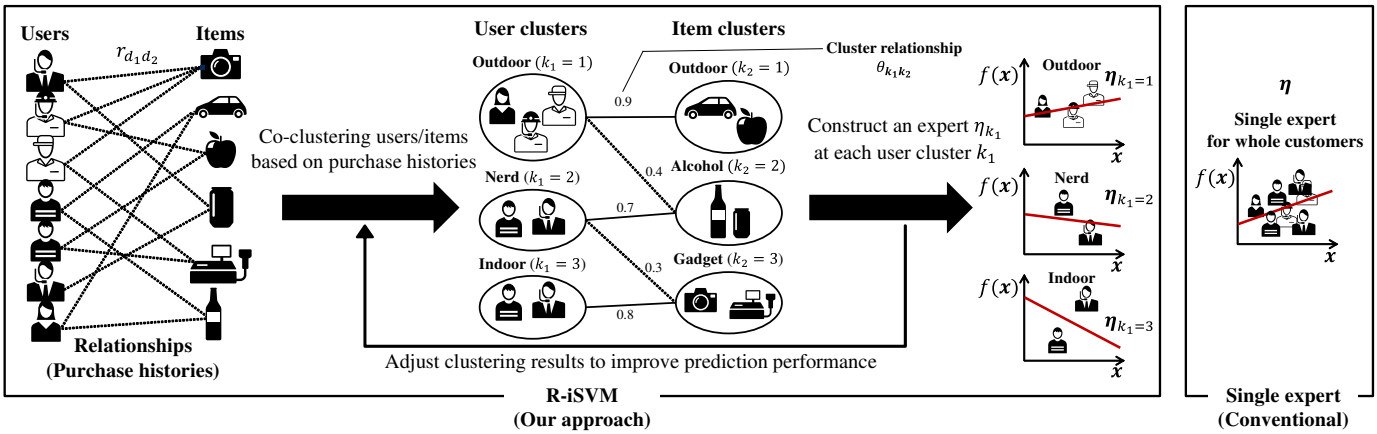


Fig. 2. Instead of using a prediction model for all customers (right), R-iSVM trains a local prediction model for each behaviorally similar customer cluster (left). R-iSVM first simultaneously groups users and items into clusters using behavioral data (purchase histories) in co-clustering fashion to identify behaviorally similar customers (E-Step), and identifies the strengths of cluster-cluster relationships between user- and item-clusters (M-Step), which later help interpretation of each cluster. Then R-iSVM trains a local demographics prediction model $\boldsymbol{\eta}_{k_1}$ at each user cluster (QP-Step). After that, R-iSVM adjusts the clustering results to improve predictive performance by checking training errors of experts (E-Step). By repeating these three steps in EM-fashion, R-iSVM constructs a mixture-of-experts model.

what you buy" [8]. These studies can be divided into two approaches: (1) *application-specific feature engineering* using behavioral data [6] and (2) *representation learning*, which learns feature vectors from behavioral data [2], [3], [5], [7]. In (1) application-specific feature engineering, as depicted in Figure 1, we can augment a customer's feature vector by gathering his behavioral data, such as the average price of items he has purchased or categories of books he has read, which may be a good indicator of his occupation. In this approach, how to design a good feature is an important problem that requires domain experts' knowledge. In (2) representation learning, *Singular Value Decomposition (SVD)* and *Tucker Decomposition* on behavioral data (e.g., user-item matrix or user-item-store tensor) has been actively used for producing low-dimensional feature vectors for customers [2], [3], [5]. A neural-embedding method has recently been applied to learn a more discriminative representation of a customer from his item purchase histories [7].

Although these demographics prediction methods achieved successes, we encountered several problems in deploying these methods to production. (1) Application-specific feature engineering requires domain-experts to make hard effort to find good features and is one of the most time-consuming tasks, involving significant trial and error [9]. Furthermore, application specific features limit the model's applicability to other domains. (2) Representation learning is apparently effective for improving predictive performance. However, automatically generated feature vectors (representations) rarely have meanings, and the prediction result would be difficult to interpret and explain to customers and colleagues (for example, "income and age positively correlate" is an intuitive explanation, but "income and auto-generated-feature-1 positively correlate" is hard to explain).

A. Design Goals

To overcome the aforementioned issues in conventional demographics prediction methods, we set **four design goals** for our model:

1) **(General)** It does not require application specific modeling, and is not even limited to demographics prediction.
2) **(Interpretable)** *It does not produce hard-to-interpret features.* Furthermore, it provides additional information that helps people interpret prediction results.
3) **(Accurate)** It achieves good predictive performance. Further, *it can co-exist with conventional feature engineering or representation learning methods* and helps to improve their predictive performances.
4) **(Scalable)** Its computational complexity is linear in the size of behavioral data. Further, the training algorithm can be fully parallelized.

In this paper, for a model that fulfills the four design goals, we present a *Relational Infinite Support Vector Machine* (R-iSVM), a mixture-of-experts model that can leverage behavioral data. As shown in Figure 2, instead of constructing a prediction model for all customers, R-iSVM finds behaviorally similar customers through co-clustering [10], [11] on behavioral data, and constructs a local demographics prediction model at each customer cluster. R-iSVM jointly models co-clustering and training of prediction models as a unified optimization problem, and thus those tasks affect each other to improve the model quality. Further, R-iSVM determines the characteristic of each customer demographics prediction model using the co-clustering results, which helps to increase interpretability. For a local demographics prediction model at each customer cluster, we use a multi-class kernel machine [12] that has better predictive performance than generalized linear models. To fill the gap between the co-clustering model (Bayesian generative model) and the multi-

TABLE I
LIST OF SYNONYMS. DEPENDING ON THE CONTEXT, WE SOMETIMES USE THESE WORDS INTERCHANGEABLY.

Entity-Relationship Data	Demographics Prediction
Entity set	Customers / items
Entity	Customer / item
Entity attributes	Customer demographics
Relationship information	Behavioral data

class kernel machine (discriminative model), we leverage the recently developed *Regularized Bayes* theory [13].

II. PRELIMINARIES

In this section, we review several concepts that appear in our model: *entity-relationship data*, *mixture-of-experts* models, and *co-clustering* methods.

A. Entity-Relationship Data

In enterprise, customer information including demographics and purchase histories are often organized as *Entity-Relationship Data*. As shown in Figure 1, entity-relationship data consists of (1) several *entity sets* that contain "master" information (e.g., customer demographics or item prices), and (2) a *relationship information* that connects those entity sets (e.g., purchase histories such as "customer A bought item B" in retail stores). Although its main target is demographics prediction, our relational mixture-of-experts model can be used in the more general task: *entity attribute prediction in entity-relationship data*. In this paper, to make the discussion general, we sometimes use the terms in Table I interchangeably.

B. Mixture-of-experts model

In classification problems such as customer demographics prediction, a classifier must capture a non-linear dependency between feature vector $\boldsymbol{x}$ and its true label y to achieve accurate prediction. So far, many non-linear classifiers such as non-linear kernel machines [12] and neural networks [7] have been proposed and widely used. However, such non-linear models tend to lack interpretability because they do not show how each dimension of the input feature vector affects the classification result [14], [15].

Mixture-of-experts model [16] is an approach for capturing non-linearity without sacrificing interpretability. Rather than constructing a single prediction model for a whole dataset, a mixture-of-experts model separates input feature space into K regions and constructs a local prediction model $\boldsymbol{w}_k$, called an *expert*, at each region k. Since feature vector $\boldsymbol{x}$ and label y are likely to show a linear dependency at each region k, a simple linear model can work well as an expert, allowing us to investigate how each dimension of the feature vector $\boldsymbol{x}$ affects the classification result y.

In the prediction phase, a mixture-of-experts model classifies input feature $\boldsymbol{x}$ as label y using experts $\mathcal{W} = \{ \boldsymbol{w}_1, \ldots, \boldsymbol{w}_K \}$ in accordance with

$$p(y \mid \boldsymbol{x}, \mathcal{W}) := \mathbb{E}_{p(z|\boldsymbol{x})}[\, p(y \mid \boldsymbol{x}, \boldsymbol{w}_z)\,] = \sum_{k}^{K} p(z = k \mid \boldsymbol{x}) p(y \mid \boldsymbol{x}, \boldsymbol{w}_k), \quad (1)$$

where z is the latent variable of input feature $\boldsymbol{x}$ representing the expert assignment. Probability density function $p(z = k \mid \boldsymbol{x})$ is called a *gating function* (or gating network) that softly assigns the input feature $\boldsymbol{x}$ to expert $\boldsymbol{w}_k$ in accordance with its value. Depending on the form of a gating function, mixture-of-experts models can become various prediction models including a Gaussian mixture of linear classifiers [16], a probabilistic decision tree [17], and a supervised co-clustering model [18].

In this paper, we present R-iSVM, a mixture-of-experts model for entity-relationship data. R-iSVM has a special gating function that assigns entity d_1 (e.g., a customer) to an expert by considering its relationship information $\{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}$ (e.g., his purchase histories) where $r_{d_1 d_2}$ indicates a relationship information (e.g., customer d_1 has bought item d_2) and $\mathcal{I}$ indicates indices of observed relationships. That is, the gating function in R-iSVM has the form of

$$p(z_{d_1} = k \mid \boldsymbol{x}_{d_1}, \{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}), \quad (2)$$

whereas the gating function in a conventional mixture-of-experts model has the form of $p(z_{d_1} = k \mid \boldsymbol{x}_{d_1})$, which use the value of entity attribute $\boldsymbol{x}_{d_1}$ (e.g., basic demographics of customer d_1). We elaborate on this discussion in Sections III-A and III-B2.

C. Co-clustering

Co-clustering (relational clustering) methods conduct clustering on multiple datasets simultaneously by using the relationship information [10], [11], [19]. Since co-clustering is based on the relationship information that represents "behavior," it can find clusters of behaviorally similar entities. For instance, as shown in Figure 2, co-clustering on users, items, and purchase histories detects customers who have similar buying habits and items bought by similar customers. *Stochastic Block Model* (SBM) [19] is a seminal work in probabilistic latent variable modeling for such co-clustering. Kemp *et al.* developed the *Infinite Relational Model* (IRM) that extends SBM into a Bayesian nonparametrics model [10], [11]. IRM uses a Dirichlet process for cluster construction, and it can infer the number of clusters without computationally intensive model selection, which is required in SBM.

III. PROPOSED MODEL

In this section, we present the *Relational Infinite Support Vector Machine* (R-iSVM), a mixture-of-experts model that can leverage behavioral data.

TABLE II
LIST OF SYMBOLS/NOTATIONS.

Symbol/Notation	Description
$[N]$	$\{ 1, \ldots, N \}$
$d_i \in D_i$	Identifier of an entity in i-th entity set (e.g., a customer / an item)
$\boldsymbol{x}_{d_i} \in \mathbb{R}^{M_i}$	Input feature vector of entity d_i, which is M_i-dimensional (e.g., basic demographics)
$y_{d_i} \in Y_i$	Class label of entity d_i (e.g., sensitive demographics)
$\mathcal{I} \subset D_i \times D_j$	Indices of observed relationships
$r_{d_i d_j} \in \mathbb{R}$	Relationship between entity d_i and entity d_j $((d_i, d_j) \in \mathcal{I})$ (e.g., purchase amount)
$K_i \in \mathbb{N}$	# of clusters in i-th entity set
$k_i \in [K_i]$	Identifier of a cluster in i-th entity set
$\boldsymbol{\eta}_{k_i} \in \mathbb{R}^{\lvert Y_i \rvert M_i}$	Expert at cluster k_i (e.g., demographics prediction model)
$z_{d_i} \in [K_i]$	Latent variable of entity d_i (cluster assignment)

A. Motivation

Our model is based on the intuition that incorporating behavioral data into demographics prediction will improve the predictive performance, as in previous demographics prediction methods [2], [3], [4], [5], [6], [7], [8]. However, in contrast to the previous work, we pursue an interpretable model, and augmenting a feature vector with representation learning on behavioral data or non-linear transformation of a feature vector is unpromising. This poses a research question:

Research Question (Explainable Demographics Prediction)**.** *Can we utilize behavioral data to improve the performance of demographics prediction without losing interpretability, i.e., keeping original feature vectors as they are?*

We found that mixture-of-experts [16], [17] has the similar purpose: to improve the predictive performance without changing the original feature vectors. However, applying the vanilla mixture-of-experts model does not meet our needs since mixture-of-experts models select prediction model on the basis of feature vector $\boldsymbol{x}$. In demographics prediction, feature vector $\boldsymbol{x}$ represents basic demographics, such as age and gender, and selecting a prediction model on the basis of these demographics means *customers who have similar basic demographics will have the same prediction model*. This policy differs from our expectation: *customers who have similar behavior will have the same prediction model*.

To make behaviorally similar customers have the same prediction model, we need to change the form of the gating function from

$$p(z_{d_1} = k \mid \boldsymbol{x}_{d_1}),$$

to

$$p(z_{d_1} = k \mid \boldsymbol{x}_{d_1}, \{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}), \quad (3)$$

where $\mathcal{I}$ indices indices of all behaviors and $r_{d_1 d_2}$ indicates a behavior such as customer d_1 has bought item d_2. In this paper, we use probabilistic co-clustering models [10], [11], [19] to define a gating function, though various gating functions that have the form of Equation 3 can be chosen. This is because co-clustering models have achieved successes in behavioral data modeling and we empirically confirmed that it performs well as a gating function on real-world datasets. Designing a new gating function can be an interesting future research direction in relational mixture-of-experts models.

B. Overview

Before explaining R-iSVM in more detail, we first provide the overview: how it trains a mixture-of-experts model from existing demographics and behavioral data, and uses the trained model to predict demographics. For ease of explanation, we hereinafter focus on a simple situation:

- Entity set D_1 represents customers, and each customer is identified by $d_1 \in D_1$.
- Entity set D_2 represents items, and each item is identified by $d_2 \in D_2$.
- Relationship $r_{d_1 d_2}$ represents a purchase history[1]. When customer d_1 has bought item d_2, $(d_1, d_2) \in \mathcal{I}$ and $r_{d_1 d_2} = 1$.

1) Training: Given a set of customer demographics

$$\{ (\boldsymbol{x}_{d_1}, y_{d_1}) \}_{d_1 \in D_1}$$

and their purchase histories $\{ r_{d_1 d_2} : (d_1, d_2) \in \mathcal{I} \}$ as a training set, R-iSVM trains K_1 experts and computes the probability of the expert assignment $p(z_{d_1} \mid \{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \})$ for all customers. Refer to the caption of Figure 2 for the detailed training flow.

2) Prediction: Given customer d_1's basic demographics $\boldsymbol{x}_{d_1}$ and his purchase histories $\{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}$, R-iSVM predicts his sensitive demographics y^* by

$$y^* = \arg\max_{y \in Y_1} F(y, \boldsymbol{x}_{d_1})$$

where

$$F(y, \boldsymbol{x}_{d_1}) := \sum_{k_1}^{K_1} p(z_{d_1} = k_1 \mid \{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}) \mathbb{E}_{q(\boldsymbol{\eta})}[F(y, \boldsymbol{x}_{d_1}; \boldsymbol{\eta}_{k_1})], \quad (4)$$

and $F(y, \boldsymbol{x}_{d_1}; \boldsymbol{\eta}_{k_1})$ is the discriminant function in a multi-class kernel machine [12], which computes the score of classifying feature vector $\boldsymbol{x}_{d_1}$ to label y by prediction model $\boldsymbol{\eta}_{k_1}$. We use *Maximum Entropy Discrimination* (MED) [20] to treat the multi-class kernel machine in a probabilistic way and MED infers posterior distribution of the prediction model $q(\boldsymbol{\eta})$. Since the prediction model is obtained as a distribution, MED computes the expectation of discriminant function over the posterior distribution, as $\mathbb{E}_{q(\boldsymbol{\eta})}[F(y, \boldsymbol{x}_{d_1}; \boldsymbol{\eta}_{k_1})]$.

What is noteworthy in Equation 4 is the gating function

$$p(z_{d_1} = k_1 \mid \{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}),$$

which selects experts for customer d_1 in accordance with his purchase histories $\{ r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}$. As mentioned before, this behavior is different from the ordinary

[1]Note that R-iSVM can support more than two entity sets by extending a relationship matrix (e.g., customers $\times$ items) to a tensor (e.g., customers $\times$ items $\times$ stores).

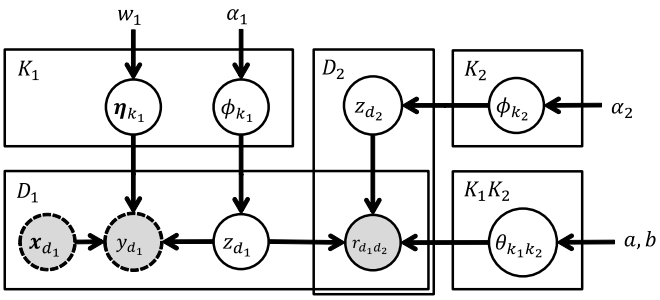


Fig. 3. Graphical representation of R-iSVM for two entity sets and one relationship matrix. Dirichlet processes for users/item clusters are formulated by truncated SBPs with truncation level K_1 and K_2, respectively.

mixture-of-experts models that use the customer demographics $\boldsymbol{x}_{d_1}$ for expert-selection. If we model customer demographics in a probabilistic manner [11], we can define a gating function

$$p(z_{d_1} = k_1 \mid \boldsymbol{x}_{d_1}, \{\, r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \,\}),$$

which uses both customer demographics and his purchase histories in expert-selection.

C. Generative Process

Figure 3 shows a graphical model of R-iSVM. The generative process of R-iSVM can be represented as follows:

$$\begin{aligned}
v_{k_1} | \alpha_1 &\sim \text{Beta}(1, \alpha_1) && (5)\\
\phi_{k_1} &= v_{k_1} \prod_{k_1'=1}^{k_1-1} (1 - v_{k_1'}) && (6)\\
v_{k_2} | \alpha_2 &\sim \text{Beta}(1, \alpha_2) && (7)\\
\phi_{k_2} &= v_{k_2} \prod_{k_2'=1}^{k_2-1} (1 - v_{k_2'}) && (8)\\
\theta_{k_1 k_2} &\sim \text{Beta}(a, b), && (9)\\
z_{d_1} \mid \boldsymbol{\phi}_1 &\sim \text{Multinomial}(\boldsymbol{\phi}_1), && (10)\\
z_{d_2} \mid \boldsymbol{\phi}_2 &\sim \text{Multinomial}(\boldsymbol{\phi}_2), && (11)\\
r_{d_1 d_2} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta} &\sim \text{Bernoulli}(\theta_{z_{d_1} z_{d_2}}), && (12)\\
\boldsymbol{\eta}_{k_1} \mid \boldsymbol{w}_1 &\sim \mathcal{N}(\boldsymbol{w}_1, I). && (13)
\end{aligned}$$

First, for each entity set (customers or items), cluster mixing parameter $\boldsymbol{\phi}$ is drawn from a Dirichlet Process (DP) with concentration parameter α (Equation 5 to 8).[2] Next, for each combination of customer cluster k_1 and item cluster k_2, its cluster relationship strength $\theta_{k_1 k_2}$ is drawn (Equation 9). Then, for all customers and items, cluster assignments z are drawn (Equation 10 to 11). After that, for all combinations of customers and items, purchase information $r_{d_1 d_2}$ is drawn in accordance with cluster assignments z and cluster relationship strength $\boldsymbol{\theta}$ (Equation 12). Finally, for each customer cluster k_1, the demographics prediction model $\boldsymbol{\eta}_{k_1}$ is constructed by a multi-class kernel machine [12] with MED [20] with a Gaussian prior (Equation 13). MED makes the training of a multi-class kernel machine into a posterior distribution inference of $\boldsymbol{\eta}_k$. We elaborate on this discussion in Section IV.

As in *Infinite Relational Model* (IRM) [10], Equations 5 to 12 model a generative process of customer-item purchase histories. In R-iSVM, this process defines a gating function $p(z_{d_1} = k_1 \mid \{\, r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \,\})$, which computes the weight of k_1-th expert (prediction model) for the customer d_1 from his behavioral data $\{\, r_{d_1 d_2} : (i, d_2) \in \mathcal{I} \wedge i = d_1 \,\}$.

[2]To develop an efficient variational inference algorithm, we use stick-breaking process (SBP) [21] to formulate Dirichlet Process instead of Chines Restaurant Process (CRP).

IV. Inference

In this section, we introduce the inference algorithm of our proposed model based on variational EM-algorithm and quadratic programming.

A. Bayesian Inference with Discriminative Model

By using the result by Zeller [22], ordinary Bayesian inference, which finds the posterior distribution of the parameters, can be formalized as the optimization problem:

$$\begin{aligned}
\min_{q \in \mathcal{Q}} \; & \text{KL}(q(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \,||\, p(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2)) \\
& - \mathbb{E}_q[\log p(\{\, r_{d_1 d_2} : (d_1, d_2) \in \mathcal{I} \,\} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2)]
\end{aligned} \tag{14}$$

where $\mathcal{Q}$ is the space of the valid probability distributions. Since it is in the optimization form, it can incorporate other optimization problems as constraints to the posterior distribution. This technique is called a *Regularized Bayes* and is recently actively studied [13].

To incorporate a multi-class kernel machine [12] into a co-clustering model, we use Maximum Entropy Discrimination (MED) [20], which formulates the training of discriminative model in a probabilistic manner. With MED, the training of a multi-class kernel machine can be formalized as follows:

$$\begin{aligned}
\min_{q \in \mathcal{Q}, \boldsymbol{\xi}} \; & \text{KL}(q(\boldsymbol{\eta}_1) \,||\, p(\boldsymbol{\eta}_1)) + C_1 \sum_{d_1}^{D_1} \xi_{d_1} \\
\text{s.t.} \; & \forall d_1 \in D_1, \forall y_1 \in Y_1 : \\
& l_{d_1}^{\Delta}(y_1) - \mathbb{E}_q[\boldsymbol{\eta}_1]^{\top} f_{d_1}^{\Delta}(y_1) \leq \xi_{d_1}, \xi_{d_1} \geq 0,
\end{aligned} \tag{15}$$

where $\boldsymbol{\eta}_1$ is the hyperplane of a multi-class support vector machine that is treated as a random variable, C_1 is a cost parameter, $\xi_{d_1} \in \boldsymbol{\xi}_1$ is the slack variable of entity d_1, $l_{d_1}^{\Delta}(y_1) := I(y_1 \neq y_{d_1})$ is a label-loss function that returns 1 if y_1 is not equal to the true class label y_{d_1}, and $f_{d_1}^{\Delta} : Y_1 \to \mathbb{R}^{|Y_1| M_1}$ is a feature mapping function that returns input feature vector of d_1 regarding it as class y_1. Solving this optimization problem results in different algorithms depends on the choice of prior distribution $p(\boldsymbol{\eta}_1)$: Gaussian prior results in l_2-SVM and Laplace prior results in l_1-SVM [23]. In this paper, we use Gaussian prior as shown in Equation 13.

Algorithm 1: Inference Algorithm

while *not converged* **do**
- **begin (QP-Step)** Solve SVM's dual QP:
  - Solve QP Equation 24 to obtain Lagrangians.
  - **for** $k_1 \in [K_1]$ **do**
    - Update $q(\eta_{k_1})$ by Equation 26.
- **begin (VB E-Step)** Update latent variables:
  - **for** $d_1 \in [D_1]$ **do**
    - **for** $k_1 \in [K_1]$ **do**
      - Update $q(z_{d_1} = k_1)$ by Equation 23.
  - **for** $d_2 \in [D_2]$ **do**
    - **for** $k_2 \in [K_2]$ **do**
      - Update $q(z_{d_2} = k_2)$.
- Reorder clusters in descending order of size [24].
- **begin (VB M-Step)** Update parameters:
  - **for** $k_1 \in [K_1]$ **do**
    - Update $q(v_{k_1})$ by Equation 20.
  - **for** $k_2 \in [K_2]$ **do**
    - Update $q(v_{k_2})$.
  - **for** $(k_1, k_2) \in [K_1] \times [K_2]$ **do**
    - Update $q(\theta_{k_1 k_2})$ by Equation 22.

1) Problem Definition: By unifying the Problem 14 and 15, we get the objective of R-iSVM as follows:

$$\begin{aligned}
&\min_{q \in \mathcal{Q}, \boldsymbol{\xi}} \ \mathrm{KL}(q(\boldsymbol{z}_1, \boldsymbol{\eta}_1) \,||\, p(\boldsymbol{z}_1, \boldsymbol{\eta}_1)) + C_1 \sum_{d_1}^{D_1} \xi_{d_1} \\
&+ C_2 \Big\{ \mathrm{KL}(q(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \,||\, p(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2)) \\
&\quad - \mathbb{E}_q[\log p(\{\, r_{d_1 d_2} : (d_1, d_2) \in \mathcal{I} \,\} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2)] \Big\} \\
&\text{s.t.} \quad \forall d_1 \in D_1, \forall y_1 \in Y_1 : \\
&\quad l^{\Delta}_{d_1}(y_1) - \mathbb{E}_q[\boldsymbol{\eta}_{z_{d_1}}]^{\top} f^{\Delta}_{d_1}(y_1) \leq \xi_{d_1}, \xi_{d_1} \geq 0
\end{aligned} \tag{16}$$

where C_2 is a hyperparameter that controls the effect of the discriminative model on the Bayesian inference.

Problem 16 shows two coupled tasks in R-iSVM: (1) the third term corresponds to the co-clustering of customers and items, and (2) the other terms and constraints correspond to the training of the demographics prediction model at each customer cluster. Since the latent variables $\boldsymbol{z}_1$ appear in both objectives, both tasks affect each other to improve the quality of their tasks: co-clustering results are adjusted to improve the performance of the customer demographics prediction.

B. Variational Inference

Since directly solving the Problem 16 is intractable, we impose the *mean-field approximation* on the posterior distribution q as in ordinary variational inference:

$$\begin{aligned}
q(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{\theta}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = &\prod_{d_1}^{D_1} q(z_{d_1}) \prod_{d_2}^{D_2} q(z_{d_2}) \\
&\prod_{k_1}^{K_1} \prod_{k_2}^{K_2} q(\theta_{k_1 k_2}) \prod_{k_1}^{K_1} q(\phi_{k_1}) \prod_{k_2}^{K_2} q(\phi_{k_2}),
\end{aligned} \tag{17}$$

and

$$q(\boldsymbol{z}_1, \boldsymbol{\eta}_1) = \prod_{d_1}^{D_1} q(z_{d_1}) \prod_{k_1}^{K_1} q(\boldsymbol{\eta}_{k_1}). \tag{18}$$

Then, we use the Lagrangian method and coordinate ascent method to minimize the objective of Problem 16. Alternatively maximizing the dual form of Problem 16 by random variables and slack variables results in an iterative inference algorithm shown in Algorithm 1.

In the remainder of this section, we elaborate on the derivation of the each step.

1) Model Parameters: For random variables other than $\boldsymbol{\eta}_1$ and $\boldsymbol{z}_1$, we obtain ordinary variational posteriors of IRM [11], [25] as follows: [3] The variational posterior of the parameter of the stick-breaking process is

$$q(v_{k_1}) = \mathrm{Beta}(\hat{\alpha}_{k_1}, \hat{\beta}_{k_1}), \tag{19}$$

and its update is

$$\begin{aligned}
\hat{\alpha}_{k_1} &= 1 + \sum_{d_1}^{D_1} q(z_{d_1} = k_1) \\
\hat{\beta}_{k_1} &= \alpha_1 + \sum_{d_1}^{D_1} \sum_{k_1' = k_1 + 1}^{K_1} q(z_{d_1} = k_1').
\end{aligned} \tag{20}$$

The variational posterior of the relationship strength is

$$q(\theta_{k_1 k_2}) = \mathrm{Beta}(\hat{a}_{k_1 k_2}, \hat{b}_{k_1 k_2}) \tag{21}$$

and its update is

$$\begin{aligned}
\hat{a}_{k_1 k_2} &= a + \sum_{(d_1, d_2) \in \mathcal{I}} q(z_{d_1} = k_1) q(z_{d_2} = k_2) r_{d_1 d_2} \\
\hat{b}_{k_1 k_2} &= b + \sum_{(d_1, d_2) \in \mathcal{I}} q(z_{d_1} = k_1) q(z_{d_2} = k_2)(1 - r_{d_1 d_2}).
\end{aligned} \tag{22}$$

2) Latent Variables: Optimizing the dual form of Problem 16 by $q(z_{d_1} = k_1)$, we obtain the variational posterior of the latent variable as follows:

$$\begin{aligned}
\log q(z_{d_1} = k_1) \propto &\\
&\mathbb{E}_q[\log p(z_{d_1} = k_1 | v_{k_1})] \\
&+ \rho \sum_{d_2 \in \mathcal{I}_{d_1}} \sum_{k_2}^{K_2} q(z_{d_2} = k_2) \mathbb{E}_q[\log p(r_{d_1 d_2} | \theta_{k_1 k_2})] \\
&+ (1 - \rho) \left\{ \sum_{y_1}^{Y_1} \omega_{y_1 d_1} \mathbb{E}_q[\boldsymbol{\eta}_{k_1}]^{\top} f^{\Delta}_{d_1}(y_1) \right\},
\end{aligned} \tag{23}$$

[3] We omit variational posteriors for second entity sets (items), since they have the same form with the first entity sets (customers).

where $\mathcal{I}_{d_1} = \{ d_2 : (i, d_2) \in \mathcal{I} \wedge i = d_1 \}$, $\rho = C_2/(1 + C_2)$, and $\omega_{y_1 d_1}$ is a Lagrangian variable.

In Equation 23, the first and second terms are the same as with ordinary IRM [11], [25]. What is noteworthy here is the third term, which is the expected demographics prediction score of customer d_1, indicating the *goodness* of cluster k_1 for the customer d_1. Through incorporating such discriminative prediction scores into the posterior inference, R-iSVM adjusts clustering results to reduce the training error.

C. Quadratic Programming

Optimizing the dual form of Problem 16 by the slack variables $\boldsymbol{\xi}$, we obtain a quadratic programming

$$\begin{aligned} \max_{\boldsymbol{\omega}} \quad & -\frac{1}{2}\sum_{k_1}^{K_1} \hat{\mu}_{k_1}^{\top}\hat{\mu}_{k_1} + \sum_{y_1}^{Y_1}\sum_{d_1}^{D_1} \omega_{y_1 d_1} l_{d_1}^{\Delta}(y_1) \\ \text{s.t.} \quad & \forall d_1 : 0 \leq \sum_{y_1}^{Y_1} \omega_{y_1 d_1} \leq C_1, \end{aligned} \tag{24}$$

where $l_{d_1}^{\Delta}(y_1) := I(y_1 \neq y_{d_1})$ is a label-loss function that returns 1 if y_1 is not equal the true class label y_{d_1}.

Optimizing the dual form of Problem 16 by the hyperplane of a multi-class kernel machine $\boldsymbol{\eta}_{k_1}$, we obtain the update equation of its variational posterior distribution

$$q(\boldsymbol{\eta}_{k_1}) = \mathcal{N}(\hat{\mu}_{k_1}, I) \tag{25}$$

as

$$\hat{\mu}_{k_1} = w_1 + \sum_{d_1}^{D_1} q(z_{d_1} = k_1) \sum_{y_1}^{Y_1} \omega_{y_1 d_1} f_{d_1}^{\Delta}(y_1). \tag{26}$$

As shown in Algorithm 1, we solve the quadratic programming Problem 24 inside our variational-EM algorithm to adjust the clustering results. By using a relaxation technique that decomposes Problem 24 into K_1 sub problems [23], the quadratic programming can be solved by ordinary SVM solvers, such as a linear-time one-slack cutting plane solver [26].

D. Computational Complexity

Algorithm 1 is scalable because its computational complexity is linear in the number of observed relationships $|\mathcal{I}|$, which is often much smaller than the number of possible relationships $|D_1||D_2|$ in real-world data. As shown in Algorithm 1, variational inference of co-clustering runs in $O(K_1 K_2 |\mathcal{I}|)$, given K_1 customer clusters, K_2 item clusters, and $\mathcal{I}$ observed relationships. For quadratic programming of a multi-class kernel machines, we have K_1 experts for customers, and each expert can be learned by a linear-time one-slack cutting plane algorithm [26], the computational complexity of which is $O(M_1|D_1|)$, where M_1 is the dimension of explanatory variable in the first entity set's attribute prediction. Thus, the final computational complexity of our algorithm is $O(K_1 K_2 |\mathcal{I}| + K_1 M_1 |D_1|)$, which is scalable.

Furthermore, for loops of variational E-Steps and M-Steps in Algorithm 1 are fully-parallelizable, which plays an important role in practice. We use OpenMP to parallelize the for loops in our R-iSVM implementation.

E. Convergence

Convergence of Algorithm 1 is guaranteed. As mentioned in Section IV-B, Algorithm 1 is a coordinate ascent maximization of the dual form of Problem 16, which alternately maximizes the objective by each variable. Coordinate ascent is guaranteed to converge to a local optimum, because each step (such as QP-, E-, and M-steps in Algorithm 1) maximizes the objective with respect to a variable and does not decrease the objective value [27].

V. Experiments

In this section, we evaluate the proposed R-iSVM on real datasets, and demonstrate its effectiveness. Through the experiments, we attempted to confirm that R-iSVM satisfies our four design goals:

- **(Generality)** R-iSVM does not require application specific modeling. To confirm this characteristic, we evaluate three real-world datasets in two domains: real retailers and online movie review site.
- **(Interpretability)** R-iSVM does not produce hard-to-interpret features. Furthermore, it provides additional information that helps people interpret prediction results. To confirm this characteristic, we prepare visualizations that help interpretation of the model.
- **(Accuracy)** R-iSVM improves entity attribute predictive performance by adopting relationship data in a mixture-of-experts manner. Further, *it can co-exist with conventional feature engineering or representation learning methods*, and helps to improve their predictive performances.
- **(Scalability)** The computational complexity of R-iSVM training is linear in the number of observed relationships in entity-relationship data. Moreover, it can be fully parallelized.

A. Setup

1) Datasets: We used three real-world datasets in our experiments:

(MovieLens) The first is the MovieLens 1M dataset [28], which comes from an online movie review website[4]. The dataset contains $1,000,209$ ratings ($\mathcal{I}$) for $6,040$ users (D_1) on $3,900$ movies (D_2). Each user has three demographics: **gender** (male/female), **age** (categorized into 7 ranges), and **occupation** (21 types). Since these attributes are categorical, we convert them into dummy variables, producing 31 dimensional vector for each user ($|x_{d_1}| = 31$). We also convert five-star ratings into a binary value if the rating is higher than the user's average rating, and vice versa.

[4]We chose MovieLens 1M because larger MovieLens datasets such as MovieLens 10M do not have user demographics.

TABLE III
DATASETS USED IN EXPERIMENTS.

Dataset	Relationship type	Entity attributes	$\lvert D_1 \rvert$	$\lvert D_2 \rvert$	$\lvert \mathcal{I} \rvert$	Density
MovieLens 1M	user × movie (5-star review)	gender, age, occupation	6,040	3,900	1,000,209	4.24%
Ta-Feng	user × item (purchase history)	age, resident	16,578	17,860	227,827	0.07%
BeiRen	user × item (purchase history)	gender, age, marital, income, education	57,693	61,097	6,396,551	0.18%

TABLE IV
(ACCURACY) PREDICTIVE PERFORMANCE OF DEMOGRAPHICS PREDICTION. EACH VALUE REPRESENTS AVERAGE F1-MICRO SCORE OF FIVE TRIALS WITH STANDARD DEVIATION. FOR ALL TASKS, R-ISVM (WITH OR WITHOUT DEMOGRAPHICS AUGMENTATION) ACHIEVED THE HIGHEST PERFORMANCE.

Dataset	**MovieLens 1M**			**Ta-Feng**		**BeiRen**				
Method	gender	age	occupation	age	residence	gender	age	education	marital	income
MC	62.7 ± 1.5	14.9 ± 3.4	9.5 ± 0.9	10.7 ± 0.7	14.0 ± 2.0	63.7 ± 0.7	56.2 ± 1.0	18.8 ± 4.2	26.9 ± 5.6	24.9 ± 8.0
DA+MC	69.4 ± 1.3	22.6 ± 0.7	11.1 ± 0.7	12.5 ± 2.9	14.9 ± 3.7	63.7 ± 0.4	56.2 ± 0.6	20.4 ± 2.3	26.0 ± 1.9	26.4 ± 1.1
R-iSVM*	69.1 ± 3.2	22.8 ± 1.9	14.2 ± 2.4	12.9 ± 2.3	20.1 ± 3.5	$\mathbf{66.1 \pm 3.2}$	$\mathbf{61.5 \pm 2.8}$	27.2 ± 4.4	$\mathbf{41.1 \pm 4.8}$	36.7 ± 4.6
DA+R-iSVM*	$\mathbf{73.7 \pm 1.2}$	$\mathbf{30.3 \pm 1.8}$	$\mathbf{15.3 \pm 0.8}$	$\mathbf{15.3 \pm 1.6}$	$\mathbf{23.9 \pm 1.9}$	64.4 ± 3.2	58.4 ± 2.0	$\mathbf{27.9 \pm 3.0}$	41.0 ± 1.3	36.8 ± 2.4

(Ta-Feng) The second is the Ta-Feng dataset[5], which comes from a real supermarket in China. The dataset contains transactions collected by a supermarket from November 2000 to February 2001. We picked out records of November 2000, which contains $16,578$ users (D_1), $17,860$ items (D_2), and $6,396,551$ purchase histories ($\mathcal{I}$). Each user has two demographic attributes: **age** (categorized into 10 ranges) and **residence** (8 areas). We convert these categorical attributes into dummy variables as in MovieLens dataset.

(BeiRen) The third is the BeiRen dataset[6], which comes from a real retailer in China. The dataset contains $57,693$ users (D_1), $61,097$ items (D_2), and $6,396,551$ purchase histories ($\mathcal{I}$). Each user has five demographic attributes: **gender** (male/female), **age** (categorized into four ranges), **marital status** (single/married), **income** (categorized into four ranges), and **education level** (six levels). We convert these categorical attributes into dummy variables as in MovieLens dataset.

2) Compared Methods: We compared four methods in terms of entity attribute prediction:

- **(MC)** A multi-class kernel machine [12]. For the implementation, we used a Python implementation of multi-class support vector machine in PyStruct package [29].
- **(DA+MC)** Demographics augmentation by Tucker decomposition of behavioral data [5]. Latent feature vectors are retrieved from Tucker decomposition of behavioral data and used to augment original basic demographics. Then, we use the augmented features to train a prediction model with **MC**.
- **(R-iSVM)** Our proposed mixture-of-experts model with basic demographics.
- **(DA+R-iSVM)** Our proposed model with user demographics augmentation as in **DA+MC**. Before running R-iSVM, user demographics are augmented as in **DA+MC**, and then R-iSVM constructs a mixture-of-experts model.

3) Tasks and Parameters: For each combination of a method and a prediction target demographics, we repeated nested five-fold cross validation five times, and computed the average and standard deviation of F1-micro scores.

For hyperparameter selection, we used two parameter grids in grid-search:

- $C_1 = \{0.1, 1, 10\}$ for SVM's hyperparameter (in MC, DA+MC, R-iSVM, and DA+R-iSVM).
- $C_2 = \{1, 64\}$ for R-iSVM's (in R-iSVM and DA+R-iSVM). Note that for other hyperparemeters in R-iSVM that come from a Bayesian part (namely, α, a, and b), we took empirical Bayes approach to learn hyperparameter values from data, and no grid-search was needed.

B. Predictive Performance

Table IV shows average F1-micro scores of compared methods in each task, together with their standard deviations. By comparing the predictive performance between MC and R-iSVM, we can observe that R-iSVM successfully increased predictive performances without producing hard-to-interpret features. By comparing DA+MC (demographics augmentation approach) and DA+R-iSVM, we can observe that our proposed model, R-iSVM, can co-exist with state-of-the-art demographics augmentation and can further improve the performance.

C. Interpretability

Next, we show the interpretability of R-iSVM in details. Figure 4 shows a visual representation of two experts in an R-iSVM model that predicts "occupation." In the figure, a row represents a class y, a column represents a feature dimension of x, and the corresponding cell represents the positive (red) or negative (blue) impact of the feature dimension on the class. From the figure, we can find out how each feature dimension affects the classification results. For example, by checking red cells in expert 12, we can find several intuitive rules in user cluster 12, such as ***if*** *gender="Female"* ***and*** *age="50–55"* ***then*** *"homemaker"*. Further, as we mentioned before, R-iSVM offers additional information to experts thanks to the co-clustering nature of its model. Figure 5 shows the cluster relationship information between experts and movie clusters in the R-iSVM model, namely, the values of $q(\boldsymbol{\theta})$. We omit weak

[5]http://recsyswiki.com/wiki/Grocery_shopping_datasets
[6]http://www.bigdatalab.ac.cn/benchmark/bm/bd?code=SNE

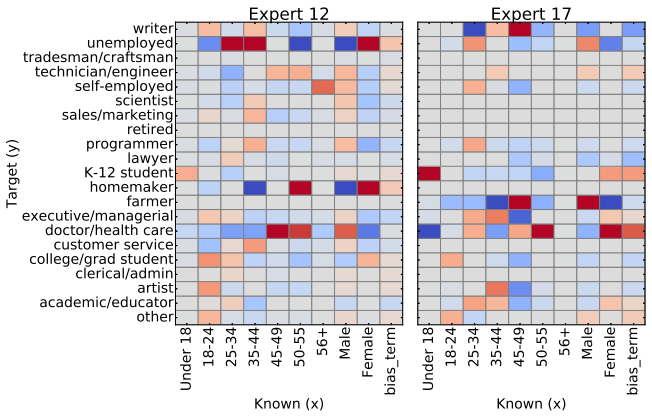


Fig. 4. (Interpretability) Visual representation of experts in R-iSVM on MovieLens 1M dataset. We can check effects of original features on classification results at a glance. For models with large number of input feature and classes, we can limit the visualization to prominent parts.

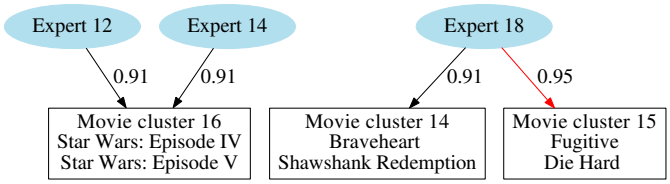


Fig. 5. (Interpretability) Visual representation of relevance between experts (user demographics prediction model) and movie clusters on MovieLens 1M dataset. For each expert, R-iSVM identifies relevant movie clusters, which helps to clarify experts' behavior in collaboration with visualization in Figure 4.

relationships from the figure and thus it only shows *heavy movie watchers* and their *movie preferences*. From Figure 5, we can learn that users in clusters 12, 14, and 18 are heavy movie watchers. By incorporating this knowledge into the rules obtained from the visual representation of the experts in Figure 4, we can further improve the interpretation. These results confirm that R-iSVM is interpretable.

D. Scalability

Finally, we demonstrate the scalability of R-iSVM. We subsampled BeiRen dataset to construct seven different-sized datasets. Figure 6 shows the runtime of R-iSVM on a computer that has two Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz with 256GB DDR memory. As in Figure 6, the runtime of R-iSVM increases linearly to the size of observed relationships, which demonstrates that R-iSVM is scalable.

VI. Related Work

Customer demographics prediction has been conducted with a wide variety of problem settings and methods [2], [3], [4], [5], [6], [7], [8]. We elaborate on two lines of work that we overviewed in the Introduction: *application-specific feature engineering* and *representation learning*. For the feature engineering approach, Culotta *et al.* proposed constructing a user's feature vector from users he follows on a social-network [6].

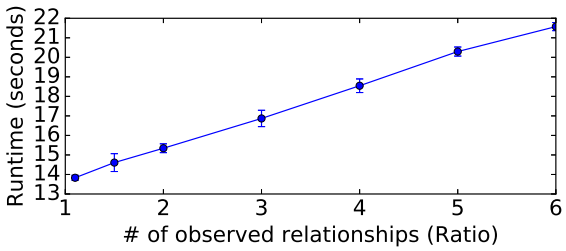


Fig. 6. (Scalability) R-iSVM's runtime on BeiRen dataset varying the number of observed relationships $|\mathcal{I}|$. Runtime increases linearly to the size of observed relationships $|\mathcal{I}|$, confirming computational complexity discussed in Section IV-D.

For representation learning, singular value decomposition (SVD) on a relationship matrix (users' behavioral data), such as web browsing history [2], [3] and location check-in [7], has achieved successes. To capture multiple types of behaviors simultaneously, Zhong *et al.* has extended relationship matrix to tensor (i.e., location check-ins and online-review), and used Tucker decomposition to obtain the latent features of users [5]. Wang *et al.* proposed using a user's purchase history in retail stores in the prediction [7]. They formulated the problem as multi-task learning that predicts multiple attributes of a user simultaneously and used a neural embedding approach to obtain a highly discriminative representation for each user. In contrast to these approaches, our design goal is to achieve high predictive performance without producing any additional hard-to-interpret features.

VII. Conclusions

To solve customer demographics prediction problems, we developed a *Relational Infinite Support Vector Machine* (R-iSVM), a novel mixture-of-experts model that can leverage behavioral data. R-iSVM has four properties:

1) **(General)** R-iSVM does not require application-specific modeling, and is not even limited to demographics prediction. *It is widely applicable for general entity-relationship data*.
2) **(Interpretable)** *R-iSVM does not produce hard-to-interpret features.* Furthermore, it provides additional information that helps people interpret prediction results.
3) **(Accurate)** R-iSVM improves entity attribute predictive performance by adopting relationship data in a mixture-of-experts manner. Further, *it can co-exist with conventional feature engineering or representation learning methods*, and helps to improve predictive performance of these methods.
4) **(Scalable)** The computational complexity of R-iSVM training is linear in the number of observed relationships in entity-relationship data. Moreover, it can be fully-parallelized.

We have evaluated R-iSVM on various real world datasets including retail store data and online movie-review data. The

experimental results demonstrated its generality, interpretability, accuracy, and scalability.

ACKNOWLEDGMENT

We thank Dr. Hao Zhang and Dr. Koji Ichikawa for critically proofreading the paper.

REFERENCES

[1] X. W. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li, "We Know What You Want to Buy: A Demographic-based System for Product Recommendation on Microblogs," in *KDD*, 2014, pp. 1935–1944.

[2] D. Murray and K. Durrell, "Inferring demographic attributes of anonymous internet users," *Web Usage Analysis and User Profiling*, vol. 1836, pp. 7–20, 2000.

[3] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen, "Demographic prediction based on user's browsing behavior," in *WWW*, 2007, pp. 151–160.

[4] Y. Dong, Y. Yang, J. Tang, Y. Yang, and N. V. Chawla, "Inferring User Demographics and Social Strategies in Mobile Social Networks," in *KDD*, 2014, pp. 15–24.

[5] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie, "You Are Where You Go: Inferring Demographic Attributes from Location Check-ins," in *WSDM*, 2015, pp. 295–304.

[6] A. Culotta, N. K. Ravi, and J. Cutler, "Predicting the Demographics of Twitter Users from Website Traffic Data," in *AAAI*, 2015, pp. 72–78.

[7] P. Wang, J. Guo, Y. Lan, J. Xu, and X. Cheng, "Your Cart tells You: Inferring Demographic Attributes from Purchase Data," in *WSDM*, 2016.

[8] J. Zhang, K. Du, R. Cheng, Z. Wei, C. Qin, H. You, and S. Hu, "Reliable Gender Prediction Based on Users' Video Viewing Behavior," in *ICDM*, 2016, pp. 649–658.

[9] M. R. Anderson and M. Cafarella, "Input Selection for Fast Feature Engineering," in *ICDE*, 2016, pp. 577–588.

[10] C. Kemp, J. J. B. Tenenbaum, T. L. T. Griffiths, T. Yamada, and N. Ueda, "Learning Systems of Concepts with an Infinite Relational Model," in *AAAI*, vol. 21, 2006, pp. 381–388.

[11] Z. Xu, "Statistical Relational Learning with Nonparametric Bayesian Models," Ph.D. dissertation, Ludwig-Maximilians-University of Munich, 2007.

[12] K. Crammer and Y. Singer, "On The Algorithmic Implementation of Multiclass Kernel-based Vector Machines," *JMLR*, vol. 2, pp. 265–292, 2001.

[13] J. Zhu, N. Chen, and E. P. Xing, "Bayesian Inference with Posterior Regularization and Applications to Infinite Latent SVMs," *JMLR*, vol. 15, pp. 1799–1847, 2014.

[14] R. D. Cook, "Detection of Influential Observation in Linear Regression," *Technometrics*, vol. 19, no. 1, pp. 15–18, 1977.

[15] Z. C. Lipton, "The Mythos of Model Interpretability," in *ICML Workshop on Human Interpretability in Machine Learning*, no. Whi, 2016.

[16] R. A. Jacobs and M. I. Jordan, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[17] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[18] M. Deodhar and J. Ghosh, "SCOAL: A Framework for Simultaneous Co-Clustering and Learning from Complex Data," *TKDD*, vol. 4, no. 3, pp. 1–31, 2010.

[19] Y. J. Wang and G. Y. Wong, "Stochastic Blockmodels for Directed Graphs," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 8–19, 1987.

[20] T. Jaakkola, M. Meila, and T. Jebara, "Maximum Entropy Discrimination," in *NIPS*, vol. AITR-1668, 1999, pp. 470–476.

[21] J. Sethuraman, "A Constructive Deefinition of Dirichlet Priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[22] A. Zellner, "Optimal Information Processing and Bayes's Theorem," *The American Statistician*, vol. 42, no. 4, pp. 278–280, 1988.

[23] J. Zhu, N. Chen, and E. P. Xing, "Infinite SVM: a Dirichlet Process Mixture of Large-margin Kernel Machines," in *ICML*, 2011, pp. 617–624.

[24] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed Variational Dirichlet Process Mixture Models." in *IJCAI*, 2007.

[25] K. Ishiguro, I. Sato, and N. Ueda, "Collapsed Variational Bayes Inference of Infinite Relational Model," *arXiv*, vol. 1, p. arXiv:1409.4757 [cs.LG], 2014.

[26] T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-Plane Training of Structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[27] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," 2016.

[28] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1—-19:19, 2015.

[29] A. Müller and S. Behnke, "PyStruct - Learning Structured Prediction in Python," *JMLR*, vol. 15, pp. 2055–2060, 2013.