Improving Design Preference Prediction Accuracy using Feature Learning

Alex Burnap*
Design Science
University of Michigan
Ann Arbor, Michigan 48109
Email: aburnap@umich.edu

Yanxin Pan*
Design Science
University of Michigan
Ann Arbor, Michigan 48109
Email: yanxinp@umich.edu

Ye Liu
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan 48109
Email: yeliu@umich.edu

Yi Ren
Mechanical Engineering
Arizona State University
Tempe, AZ 85287
Email: yiren@asu.edu

Honglak Lee
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan 48109
Email: honglak@eecs.umich.edu

Richard Gonzalez
Psychology
University of Michigan
Ann Arbor, Michigan 48109
Email: gonzo@umich.edu

Panos Y. Papalambros
Mechanical Engineering
University of Michigan
Ann Arbor, Michigan 48109
Email: pyp@umich.edu

Quantitative preference models are used to predict customer choices among design alternatives by collecting prior purchase data or survey answers. This paper examines how to improve the prediction accuracy of such models without collecting more data or changing the model. We propose to use features as an intermediary between the original customer-linked design variables and the preference model, transforming the original variables into a feature representation that captures the underlying design preference task more effectively. We apply this idea to automobile purchase decisions using three feature learning methods (principal component analysis, low rank and sparse matrix decomposition, and exponential sparse restricted Boltzmann machine), and show that the use of features offers improvement in prediction accuracy using over 1 million real passenger vehicle purchase data. We then show that the interpretation and visualization of these feature representations may be used to help augment data-driven design decisions.

1 Introduction

Much research has been devoted to develop design preference models that predict customer design choices. A common approach is to: (i) Collect a large database of previous purchases that includes customer data, e.g., age, gender, income, and purchased product design data, e.g., # cylinders, length, curb weight — for an automobile; and (ii) statistically infer a design preference model that links customer and product variables, using conjoint analysis or discrete choice analysis such as logit, mixed logit, and nested logit models [1, 2].

However, a customer may not purchase a vehicle solely due to interactions between these two sets of variables, e.g., a 50-year old male prefers 6-cylinder engines. Instead, a customer may purchase a product for more meaningful design attributes that are functions of the original variables, such as environmental sustainability or sportiness [3, 4]. These 'meaningful' intermediate functions of the original variables, both of the customer and of the design, are hereafter termed *features*. We posit that using customer and product features, instead of the original customer and product variables, may increase the prediction accuracy of the design preference model.

Our goal then is to find features that improve this accuracy. To this end, one common approach is to ask design and marketing domain experts to choose these features intuitively, such as a design's social context [5] and visual design interactions [6]. For example, eco-friendly vehicles may be a function of miles per gallon (MPG) and emissions, whereas environmentally active customers may be a function of age, income, and geographic region. An alternative explored in this paper is to find features "automatically" using feature learning methods studied in computer science and statistics. As shown in Figure 1, feature learning methods create an

*Authors contributed equally to this work.

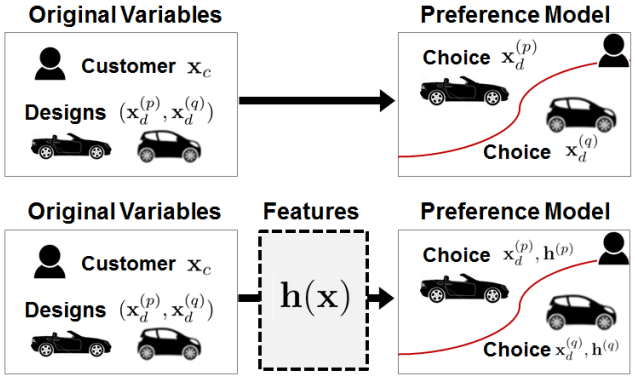

Fig. 1. The concept of feature learning as an intermediate mapping between variables and a preference model. The diagram on top depicts conventional design preference modeling (e.g., conjoint analysis) where an inferred preference model discriminates between alternative design choices for a given customer. The diagram on bottom depicts the use of features as an intermediate modeling task.

intermediate step between the original data and the design preference model by forming a more efficient feature representation of the original data. Certain well-known methods such as principal component analysis may be viewed similarly, but more recent feature learning methods have shown impressive results in 2D image object recognition [7] and 1D waveform prediction [8].

We conduct an experiment on automobile purchasing preferences to assess whether three feature learning methods increase design preference prediction accuracy: (1) principle component analysis, (2) low-rank + sparse matrix decomposition, and (3) exponential family sparse restricted Boltzmann machines [9, 10]. We cast preference prediction as a binary classification task by asking the question, "given customer $\mathbf{x}$, do they purchase vehicle p or vehicle q." Our data set is comprised of 1,161,056 data points generated from 5582 real passenger vehicle purchases in the United States during model year 2006 (MY2006).

The first contribution of this work is an increase of preference prediction accuracy by 2%-7% just using simple "single-layer" feature learning methods, as compared with the original data representation. These results suggest features indeed better represent the customer's underlying design preferences, thus offering deeper insight to inform decisions during the design process. Moreover, this finding is complementary to recent work in crowdsourced data gathering [11,12] and nonlinear preference modeling [13,14]) since they do not affect the preference model or data set itself.

The second contribution of this work is to show how features may be used in the design process. We show that feature interpretation and feature visualization offer designers additional tools for augmenting design decisions. First, we interpret the most influential pairings of vehicle features and customer features to the preference task, and contrast this with the same analysis using the original variable representation. Second, we visualize the theoretically optimal vehicle for a given customer within the learned feature representation, and show how this optimal vehicle, which does not exist, may be used to suggest design improvements upon current models of vehicles that do exist in the market.

Methodological contributions include being the first to use recent feature learning methods on a heterogeneous design and marketing data set. Recent feature learning research has focused on homogeneous data, in which all variables are real-valued numbers such as pixel values for image recognition [7, 15]; in contrast, we explicitly model the heterogeneous distribution of the input variables, for example 'age' being a real-valued variable and 'General Motors' being a categorical variable. Subsequently, we give a number of theoretical extensions: First, we use exponential family generalizations for the sparse restricted Boltzmann machines, enabling explicit modeling of statistical distributions for heterogeneous data. Second, we derive theoretical bounds on the reconstruction error of the low-rank + sparse matrix decomposition feature learning method.

This paper is structured as follows: Section 2 discusses efforts to increase prediction accuracy by the design community, as well as feature learning advances in the machine learning community. Section 3 sets up the preference prediction task as a binary classification problem. Section 4 details three feature learning methods and their extension to suit heterogeneous design and market data. Section 5 details the experimental setup of the preference prediction task, followed by results showing improvement of preference prediction accuracy. Section 6 details how features may be used to inform design decisions through feature interpreation and feature visualization. Section 7 concludes this work.

2 Background and Related Work

Design preference modeling has been investigated in design for market systems, where quantitative engineering and marketing models are linked to improve enterprise-wide decision making [16–18]. In such frameworks, the design preference model is used to aggregate input across multiple stakeholders, with special importance on the eventual customer within the targeted market segment [19].

These design preference models have been shown to be especially useful for the design of passenger vehicles, as demonstrated across a variety of applications such as engine design [20], vehicle packaging [21], brand recognition [22], and vehicle styling [3, 6, 23]. Connecting many of these research efforts is the desire for improved prediction accuracy of the underlying design preference model. With increased prediction accuracy, measured using "held out" portions of the data, greater confidence may be placed in the fidelity of the resulting design conclusions.

Efforts to improve prediction accuracy involve: (i) Developing more complex statistical models to capture the heterogeneous and stochastic nature of customer preferences; examples yuinclude mixed and nested logit models [1, 2], consideration sets [24], and kernel-based methods [13, 14, 25]; and (ii) creating adaptive questionnaires to obtain stated information more efficiently using a variety of active learning methods [26, 27].

This work is different from (i) above in that the set

of features learned is agnostic of the particular preference model used. One can just as easily switch out the l^2 logit design preference model used in this paper for another model, whether it be mixed logit or a kernel machine. This work is also different from (ii) in that we are working with a set of revealed data on actual vehicle purchases, rather than eliciting this data through a survey. Accordingly, this work is among recent efforts towards data-driven approaches in design [28], including design analytics [29] and design informatics [30], in that we are directly using data to augment existing modeling techniques and ultimately suggest actionable design decisions.

2.1 Feature learning

Feature learning methods capture correlations implicit in the original variables by "encoding" the original variables in a new feature representation. This representation keeps the number of data the same while changing the length of each data point from M variables to K features. The idea is to minimize an objective function defining the reconstruction error between the original variables and their new feature representation. If this representation is more meaningful for the discriminative design preference prediction task, we can use the same supervised model (e.g., logit model) as before to achieve higher predictive performance. More details are given in Section 4.

The first feature learning method we examined is principal component analysis (PCA). While not conventionally referred to as a feature learning method, PCA is chosen for its ubiquitous use and its qualitative difference from the other two methods. In particular, PCA makes the strong assumption that the data is Gaussian noise distributed around a linear subspace of the original variables, with the goal of learning the eigenvectors spanning this subspace [31]. The features in our case are the coefficients of the original variables when projected onto this subspace or, equivalently, the inner product with the learned eigenvectors.

The second method is low-rank + sparse matrix decomposition (LSD). This method is chosen as it defines the features implicitly withing the preference model. In particular, LSD decomposes the "part-worth" coefficients contained in the design preference model (e.g., conjoint analysis or discrete choice analysis) into a low-rank matrix plus a sparse matrix. This additive decomposition is motivated by results from the marketing literature suggesting certain purchase consideration are linearly additive [32], and thus well captured by decomposed matrices [33]. An additional motivation for a linear decomposition model is the desire for interpretability [34]. Predictive consumer marketing oftentimes uses these learned coefficients to work hand-in-hand with engineering design to generate competitive products or services [35]. Such advantages are bolstered by separation of factors captured by matrix decomposition, as separation may lead to better capture of heterogeneity among market segments [36]. Readers are referred to [37] for further in-depth discussion.

The third method is the exponential family sparse restricted Boltzmann machine (RBM) [9, 38]. This method is chosen as it explicitly represents the features, in contrast with the LSD. The method is a special case of a Boltzmann machine, an undirected graph model in which the energy associated within an energy state space defines the probability of finding the system in that state [9]. In the RBM, each state is determined by both visible and hidden nodes, where each node corresponds to a random variable. The visible nodes are the original variables, while the hidden nodes are the feature representation. The "restricted" portion of the RBM refers to the restriction on visible-visible connections and hidden-hidden connections as shown in Figure 4

All three feature learning methods are considered simple in that they are single-layer models. The aforementioned results in 2D image object recognition and 1D waveform speech recognition have been achieved using hierarchical models, built by stacking multiple single-layer models. We chose single-layer feature learning methods here as an initial effort and to explore parameter settings more easily; as earlier noted, there is limited work on feature learning methods for heterogeneous data (e.g., categorical variables) and most advances are currently only on homogeneous data (e.g., real-valued 2D image pixels).

3 Preference Prediction as Binary Classification

We cast the task of predicting a customer's design preferences as a binary classification problem: Given customer j, represented by a vector of heterogeneous customer variables $\mathbf{x}_c^{(j)}$, as well as two passenger vehicle designs p and q, each represented by a vector of heterogeneous vehicle design variables $\mathbf{x}_d^{(p)}$ and $\mathbf{x}_d^{(q)}$, which passenger vehicle will the customer purchase? We use a real data set of customers and their passenger vehicle purchase decisions as detailed below [39].

3.1 Customer and vehicle purchase data from 2006

The data used in this work combines the Maritz vehicle purchase survey from 2006 [39], the Chrome vehicle variable database [40], and the 2006 estimated U.S. state income and living cost data from the U.S. Census Bureau [41] to create a data set with both customer and passenger vehicle variables. These combined data result in a matrix of purchase records, with each row corresponding to a separate customer and purchased vehicle pair, and each column corresponding to a variable describing the customer (e.g., age, gender, income) or the purchased vehicle (e.g., # cylinders, length, curbweight).

From this original data set, we focus only on the customer group who bought passenger vehicles of size classes between mini-compact and large vehicles, thus excluding data for station wagons, trucks, minivans, and utility vehicles. In addition, purchase data for customers who did not consider other vehicles before their purchases were removed, as well data for customers who purchased vehicles for another party.

The resulting database contained 209 unique passenger vehicle models bought by 5582 unique customers. The full list of customer variables and passenger vehicle variables can

Table 1. Customer variables $\mathbf{x}_c$ and their variable types

Customer Variable	Type	Customer Variable	Type
Age	Real	U.S. State Cost of Living	Real
Number of House Members	Real	Gender	Binary
Number of Small Children	Real	Income Bracket	Categorical
Number of Med. Children	Real	House Region	Categorical
Number of Large Children	Real	Education Level	Categorical
Number of Children	Real	U.S. State	Categorical
U.S. State Average Income	Real		

Table 2. Design variables $\mathbf{x}_d$ and their variable types

Design Variable	Type	Design Variable	Type
Invoice	Real	AWD/4WD	Binary
MSRP	Real	Automatic Transmission	Binary
Curbweight	Real	Turbocharger	Binary
Horsepower	Real	Supercharger	Binary
MPG (Combined)	Real	Hybrid	Binary
Length	Real	Luxury	Binary
Width	Real	Vehicle Class	Categorical
Height	Real	Manufacturer	Categorical
Wheelbase	Real	Passenger Capacity	Categorical
Final Drive	Real	Engine Size	Categorical
Diesel	Binary		

be found in Tables 1 and 2. The variables in these tables are grouped into three unit types: Real, binary, and categorical, based on the nature of the variables.

3.2 Choice set training, validation, and testing split

We converted the data set of 5582 passenger vehicle purchases into a binary choice set by generating all pairwise comparisons between the purchased vehicle and the other 208 vehicles in the data set for all 5582 customers. This resulted in $N = 1,161,056$ data points, where each datum indexed by n consisted of a triplet (j,p,q) of a customer indexed by j and two passenger vehicles indexed by p and q, as well as a corresponding indicator variable $y^{(n)} \in \{0,1\}$ describing which of the two vehicles was purchased.

This full data were then randomly shuffled, and split into training, validation, and testing sets. As previous studies have shown the impact on prediction performance given different generations of choice sets [42], we created 10 random shufflings and subsequent data splits of our data set, and run the design preference prediction experimental procedure of Section 5 on each one independently. This work is therefore complementary to studies on developing appropriate choice set generation schemes such as [43]. Full details into the data processing procedure are given in Section 5.

3.3 Bilinear design preference utility

We adopt the conventions of utility theory for the measure of customer preference over a given product [44]. Formally, each data point consists of a pairwise comparison between vehicles p and q for customer j , with corresponding customer variables $\mathbf{x}_c^{(j)}$ for $j \in \{1,\ldots,5582\}$ and original variables of the two vehicle designs, $\mathbf{x}_d^{(p)}$ and $\mathbf{x}_d^{(q)}$ for $p,q \in \{1,\ldots,209\}$. We assume a bilinear utility model for customer j and vehicle p:

$$U_{jp} = \left[\text{vec}\left(\mathbf{x}_c^{(j)} \otimes \mathbf{x}_d^{(p)}\right)^T, \left(\mathbf{x}_d^{(p)}\right)^T\right] \boldsymbol{\omega}, \tag{1}$$

where $\otimes$ is an outer product for vectors, $\text{vec}(\cdot)$ is vectorization of a matrix, $[\cdot,\cdot]$ is concatenation of vectors, and $\boldsymbol{\omega}$ is the part-worth vector.

3.4 Design preference model

The preference model refers to the assumed relationship between the bilinear utility model described in Section 3.3 and a label indicating which of the two vehicles the customer actually purchased. While the choice of preference model is not the focus of this paper, we pilot-tested popularly used models including l^1 and l^2 logit model, naïve Bayes, l^1 and l^2 linear as well as kernelized support vector machine, and random forests.

Based on these pilot results, we chose the l^2 logit model due to its widespread use in the design and marketing communities [37, 45]; in particular, we used the primal form of the logit model. Equation (2) captures how the logit model describes the probabilistic relationship between customer j's preference for either vehicle p or vehicle q as a function of their associated utilities given by Equation (1). Note that ε are Gumbel-distributed random variables accounting for noise over the underlying utility of the customer j's preference for either vehicle p or vehicle q.

$$P^{(n)} = P_{(j,p,q)} = P\left(U_{jp} + \varepsilon_{jp} > U_{jq} + \varepsilon_{jq}\right) = \frac{e^{U_{jp}}}{e^{U_{jp}} + e^{U_{jq}}} \tag{2}$$

Parameter Estimation

We estimate the parameters of the logit model in Eq. (2) using conventional convex loss function minimization using

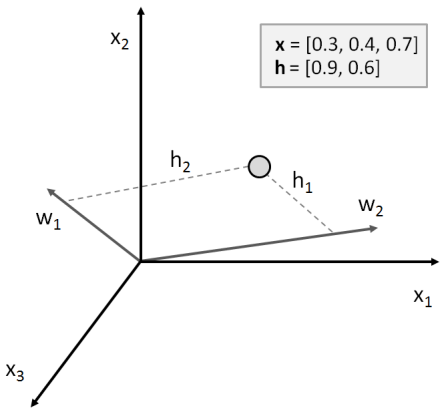

Fig. 2. The concept of principle component analysis shown using an example with a data point represented by three original variables $\mathbf{x}$ projected to a two dimensional subspace spanned by $\mathbf{w}$ to obtain features $\mathbf{h}$.

the log-loss regularized with the l^2 norm.

$$\min_{\omega,\alpha} \frac{1}{N} \sum_{n=1}^{N} (y^{(n)} \log P^{(n)} + (1-y^{(n)}) \log(1-P^{(n)})) + \alpha \|\omega\|^2 \tag{3}$$

where $y^{(n)} = y_{(jpq)}$ is 1 if customer j chose vehicle p to purchase, and 0 if vehicle q was purchased; and α is the l^2 regularization hyperparameter. The optimization algorithm used to minimize this regularized loss function was stochastic gradient descent, with details of hyperparameter settings given in Section 5.

4 Feature Learning

We present three qualitatively different feature learning methods as introduced in Section 2: (1) principal component analysis, (2) low-rank + sparse matrix decomposition, and (3) exponential family sparse restricted Boltzmann machine. Furthermore, we discuss their extensions to better suit the market data described in Section 3, as well as derivation of theoretical guarantees.

4.1 Principal Component Analysis

Principal component analysis (PCA) maps the original data representation $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T \in \mathbb{R}^{M\times 1}$ to a new feature representation $\mathbf{h} = [h_1, h_2, \ldots, h_K]^T \in \mathbb{R}^{K\times 1}$, $K \leq M$, with an orthogonal transformation $\mathbf{W} \in \mathbb{R}^{M\times K}$. Assume that the original data representation $\mathbf{x}$ has zero empirical mean (otherwise we simply subtract the empirical mean from $\mathbf{x}$). The mapping is given by:

$$\mathbf{h} = \mathbf{x}^T \mathbf{W} \tag{4}$$

The PCA representation has the following properties: (1) h_1 has the largest variance, and the variance of h_i is not smaller than the variance of h_j for all $j < i$; (2) the columns of $\mathbf{W}$ are orthogonal unit vectors; and (3) $\mathbf{h}$ and $\mathbf{W}$ minimize the reconstruction error ε:

$$\varepsilon = ||\mathbf{x} - \mathbf{h}||^2 \tag{5}$$

When the q columns of $\mathbf{W}$ consist of the first q eigenvectors of $\mathbf{x}^T\mathbf{x}$, the above properties are all satisfied, and the PCA representation can be calculated by Equation (4).

4.2 Low-Rank + Sparse Matrix Decomposition

The utility model U_{rp} given in Equation (1) can be rewritten into matrix form, in which Ω is a matrix reshaped from the "part-worth" coefficients vector ω:

$$U_{rp} = [\left(\mathbf{x}_c^{(j)}\right)^T, 1] \Omega \mathbf{x}_d^p \tag{6}$$

The decomposition of the original part-worth coefficients into a low-rank matrix and a sparse matrix may better represent customer purchase decisions than the large coefficient matrix of all pairwise interactions given in Equation (1) and as detailed in Section 2. Accordingly, we decompose Ω into a low-rank matrix $\mathbf{L}$ of rank r superimposed with a sparse matrix $\mathbf{S}$, i.e. $\Omega = \mathbf{L} + \mathbf{S}$. This problem may be solved in the general case exactly with the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{S}} l(\mathbf{L},\mathbf{S};\mathbf{X}_c,\mathbf{X}_d,\mathbf{y}) \\ &s.t. \quad rank(\mathbf{L}) \leq r \\ &\quad\quad \mathbf{S} \in \mathcal{C} \end{aligned} \tag{7}$$

where $\mathbf{X}_u$ and $\mathbf{X}_c$ are the full set of customer and vehicle data, $\mathbf{y}$ is the vector of whether customer j chose vehicle p or vehicle q, $l(\cdot)$ is the log-loss without the l^2 norm,

$$\begin{aligned} &l(\mathbf{L},\mathbf{S};\mathbf{X}_c,\mathbf{X}_d,\mathbf{y}) \\ &= \frac{1}{N} \sum_{n=1}^{N} (y^{(n)} \log P^{(n)} + (1-y^{(n)}) \log(1-P^{(n)})) \end{aligned} \tag{8}$$

and $\mathcal{C}$ is a convex set corresponding to the sparse matrix $\mathbf{S}$. As this problem is intractable (NP-hard), we instead learn this decomposition of matrices using an approximation obtained via regularized loss function minimization:

$$\min_{\mathbf{L},\mathbf{S}} l(\mathbf{L},\mathbf{S};\mathbf{X}_c,\mathbf{X}_d,\mathbf{y}) + \lambda_1 ||\mathbf{L}||_* + \lambda_2 ||\mathbf{S}||_1 \tag{9}$$

where $||\cdot||_*$ is the nuclear norm to promote low-rank structure, and $||\cdot||_1$ is the l_1-norm.

In particular, while a number of low-rank regularizations may be used to solve Eq. (9), e.g., trace norm and log-determinant norm [46]. We choose the nuclear norm

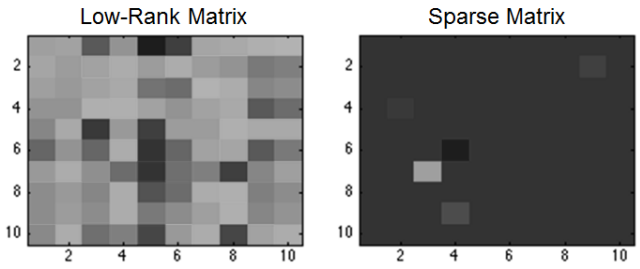

Fig. 3. The concept of low-rank + sparse matrix decomposition using an example "part-worth coefficients" matrix of size 10 x 10 decomposed into two 10 x 10 matrices with low rank or sparse structure. Lighter colors represent larger values of elements in each decomposed matrix.

as it may be applied to any general matrix, while the trace norm and log-determinant regularization are limited to positive semidefinite matrices. Moreover, the nuclear norm is often considered optimal as $||\mathbf{L}||_*$ is the convex envelop of $Rank(\mathbf{L})$, implying that $||\mathbf{L}||_*$ is the largest convex function smaller than $Rank(\mathbf{L})$ [46].

Definition 1. *For matrix* $\mathbf{L}$*,the nuclear norm is defined as,*

$$||\mathbf{L}||_* := \sum_{i=1}^{\min(dim(L))} s_i(\mathbf{L})$$

where $s_i(\mathbf{L})$ *is a singular value of* $\mathbf{L}$*.*

4.2.1 Parameter Estimation

The non-differentiability of the convex low-rank + sparse approximation given in Eq. (9) necessitates optimizations techniques such as augmented Lagrangian [47], semidefinite programming [48], and proximal methods [49]. Due to theoretical guarantees on convergence, we choose to train our model using proximal methods which are defined as follows.

Definition 2. *Let* $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be a closed proper convex function. The proximal operator of* f *is defined as*

$$\mathbf{prox}_f(\mathbf{v}) = \arg\min_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{1}{2}||\mathbf{v}-\mathbf{x}||_2^2 \right)$$

With these preliminaries, we now detail the proximal gradient algorithm used to solve Eq. (9) using low-rank and l^1 proximal operators. Denote $f(\cdot) = ||\cdot||_*$, and its proximal operator as $prox_f$. Similarly denote the proximal operator for the l^1 regularization term by $prox_S$, $i = 1,\ldots n$. Details of calculating $prox_f$ and $prox_S$ may be found in Appendix A.

With this notation, the proximal optimization algorithm to solve Equation (9) is given by Algorithm 1. Moreover, this algorithm is guaranteed to converge with constant step size as given by the following lemma [49].

Algorithm 1 Low-Rank + Sparse Matrix Decomposition

Input: Data $\mathbf{X}_c$, $\mathbf{X}_d$, $\mathbf{y}$
Initialize $\mathbf{L}^0 = \mathbf{0}, \mathbf{S}^0 = \mathbf{0}$
repeat
  $\mathbf{L}^{t+1} = prox_f(\mathbf{L}^t - \eta_t \nabla_{\mathbf{L}^t} l(\mathbf{L},\mathbf{S};\mathbf{X}_c,\mathbf{X}_d,\mathbf{y}))$
  $\mathbf{S}^{t+1} = prox_S(\mathbf{S}^t - \eta_t \nabla_{\mathbf{S}^t} l(\mathbf{L},\mathbf{S};\mathbf{X}_c,\mathbf{X}_d,\mathbf{y}))$
until $\mathbf{L}^t$, $\mathbf{S}_i^t$ are converged

Lemma 1. ***Convergence Property***
When ∇l *is Lipschitz continuous with constant* ρ*, this method can be shown to converge with rate* $O(\frac{1}{k})$ *when a fixed step size* $\eta_t = \eta \in (0, 1/\rho]$ *is used. If* ρ *is not known, the step sizes* η_t *can be found by a line search; that is, their values are chosen in each step.*

4.2.2 Error Bound on Low-Rank + Sparse Estimation

We additionally prove a variational bound that guarantees this parameter estimation method converges to a unique solution with bounded error as given by the following theorem.

Theorem 1. *Error Bound on Low-Rank+Sparse Estimation*

$$|\triangle l| \leq \min\left(dim(L_0)\right) ||\mathbf{L}^* - \mathbf{L}_0||_2$$

where $\mathbf{L}^*$ *is the optima of problem (9) and* $\mathbf{L}_0$ *is the matrix minimizing the loss function* $l(\cdot)$*.*

The proof of this theorem is given in Appendix A.

4.3 Restricted Boltzmann machine

The restricted Boltzmann machine (RBM) is an energy-based model in which an energy state is defined by a layer of M visible nodes corresponding to the original variables $\mathbf{x}$ and a layer of K features denoted as $\mathbf{h}$. The energy for a given pair of original variables and features determines the probability associated with finding the system in that state; like nature, systems tend to states that minimize their energy and thus maximize their probability. Accordingly, maximizing the likelihood of the observed data $\mathbf{x}^{(1)} \ldots \mathbf{x}^{(N)} \in \mathbb{R}^M$ and its corresponding feature representation $\mathbf{h}^{(1)} \ldots \mathbf{h}^{(N)} \in \mathbb{R}^K$ is a matter of finding the set of parameters that minimize the energy for all observed data.

While traditionally this likelihood consists of binary variables and binary features, as described in Table 1 and Table 2, our passenger vehicle purchase data set consists of M_G Gaussian variables, M_B binary variables, and M_C categorical variables. We accordingly define three corresponding energy functions E_G, E_B, and E_C, in which each energy function connects the original variables and features via a weight matrix $\mathbf{W}$, as well as biases for each original variable and feature, $\mathbf{a}$ and $\mathbf{b}$ respectively.

Real-valued random variables (e.g., vehicle curb weight) are modeled using the Gaussian density. The energy function

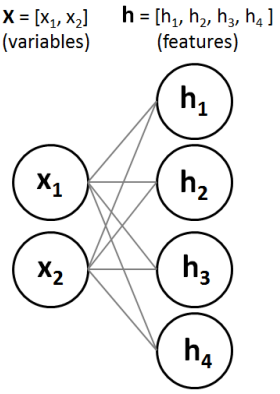

Fig. 4. The concept of the exponential family sparse restricted Boltzmann machine. The original data are represented by nodes in the visible layer by $[x_1, x_2]$, while the feature representation of the same data is represented by nodes in the hidden layer $[h_1, h_2, h_3, h_4]$. Undirected edges are restricted to being only between the original layer and the hidden layer, thus enforcing conditional independence between nodes in the same layer.

for Gaussian inputs and binary hidden nodes is:

$$E_G(\mathbf{x},\mathbf{h};\theta) = -\sum_{m=1}^{M_G}\sum_{k=1}^{K} h_k w_{km} x_m + \frac{1}{2}\sum_{m=1}^{M_G}(x_m - b_m)^2 - \sum_{k=1}^{K} a_k h_k \tag{10}$$

where the variance term is clamped to unity under the assumption that the input data are standardized.

Binary random variables (e.g., gender) are modeled using the Bernoulli density. The energy function for Bernoulli nodes in both the input layer and hidden layer is:

$$E_B(\mathbf{x},\mathbf{h};\theta) = -\sum_{m=1}^{M_B}\sum_{k=1}^{K} h_k w_{km} x_m - \sum_{m=1}^{M_B} x_m b_m - \sum_{k=1}^{K} a_k h_k \tag{11}$$

Categorical random variables (e.g., vehicle manufacturer) are modeled using the categorical density. The energy function for categorical inputs with Z_m classes for m-th categorical input variable (e.g., Toyota, General Motors, etc.) is given by:

$$E_C(\mathbf{x},\mathbf{h};\theta) = -\sum_{m=1}^{K_m}\sum_{k=1}^{K}\sum_{z=1}^{Z_m} h_k w_{kmz}\delta_{mz}x_{mz} - \sum_{m=1}^{M_C}\sum_{z=1}^{Z_m}\delta_{mz}x_{mz}b_{mz} - \sum_{k=1}^{K} a_k h_k \tag{12}$$

where $\delta_{mz} = 1$ if $x_{mz} = 1$ and 0 otherwise.

Given these energy functions for the heterogeneous original variables, the probability of a state with energy $E(\mathbf{x},\mathbf{h};\theta) = E_G(\mathbf{x},\mathbf{h};\theta) + E_B(\mathbf{x},\mathbf{h};\theta) + E_C(\mathbf{x},\mathbf{h};\theta)$, in which $\theta = \{\mathbf{W},\mathbf{a},\mathbf{b}\}$ are the energy function weights and bias parameters, is defined by the Boltzmann distribution.

$$P(\mathbf{x},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h};\theta)}}{\sum_{\mathbf{x}}\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h};\theta)}} \tag{13}$$

The "restriction" on the RBM is to disallow visible-visible and hidden-hidden node connections. This restriction results in conditional independence of each individual hidden unit h given the vector of inputs $\mathbf{x}$, and each visible unit x given the vector of hidden units $\mathbf{h}$.

$$P(\mathbf{h}|\mathbf{x}) = \prod_{n=1}^{N} P(h_n|\mathbf{x}) \tag{14}$$

$$P(\mathbf{x}|\mathbf{h}) = \prod_{k=1}^{K} P(x_k|\mathbf{h}) \tag{15}$$

The conditional density for a single binary hidden unit given the combined K_G Gaussian, K_B binary, and K_C categorical input variables is then:

$$\sigma(a_n + \sum_{k=1}^{K_G} w_{nk}x_k + \sum_{k=1}^{K_B} w_{nk}x_k + \sum_{k=1}^{K_C}\sum_{d=1}^{D_k} w_{nk}\delta_{kd}x_k) \tag{16}$$

where $\sigma(s) = \frac{1}{1+\exp(-s)}$ is a sigmoid function.

For an input data point $\mathbf{x}^{(n)}$, its corresponding feature representation $\mathbf{h}^{(n)}$ is given by sampling the "activations" of the hidden nodes.

$$[P(h_1 = 1|\mathbf{x},\theta), \ldots, P(h_N = 1|\mathbf{x},\theta)] \tag{17}$$

Parameter Estimation

To train the model, we optimize the weight and bias parameters $\theta = \{\mathbf{W},\mathbf{b},\mathbf{a}\}$ by minimizing the negative log-likelihood of the data $\{\mathbf{x}^{(1)}\ldots\mathbf{x}^{(N)}\}$ using gradient descent. The gradient of the log-likelihood is:

$$\begin{aligned}\frac{\partial}{\partial\theta}\sum_{n=1}^{N}\log P\left(\mathbf{x}^{(n)}\right) &= \frac{\partial}{\partial\theta}\sum_{n=1}^{N}\log\sum_{\mathbf{h}} P\left(\mathbf{x}^{(n)},\mathbf{h}\right)\\ &= \frac{\partial}{\partial\theta}\sum_{n=1}^{N}\log\sum_{\mathbf{h}}\frac{e^{-E(\mathbf{x}^{(n)},\mathbf{h})}}{\sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x}^{(n)},\mathbf{h})}}\\ &= \sum_{n=1}^{N}\mathbb{E}_{\mathbf{h}|\mathbf{x}^{(n)}}\left[\frac{\partial}{\partial\theta}E\left(\mathbf{x}^{(n)},\mathbf{h}\right)\right]\\ &\quad - \mathbb{E}_{\mathbf{h},\mathbf{x}}\left[\frac{\partial}{\partial\theta}E\left(\mathbf{x},\mathbf{h}\right)\right]\end{aligned} \tag{18}$$

The gradient is the difference of two expectations, the first of which is easy to compute since it is "clamped" at the input datum $\mathbf{x}$, but the second of which requires the joint density over the entire $\mathbf{x}$ space for the model.

In practice, this second expectation is approximated using the contrastive divergence algorithm by Gibbs, sampling the hidden nodes given the visible nodes, then the visible

nodes given the hidden nodes, and iterating a sufficient number of steps for the approximation [50]. During training, we induce sparsity of the hidden layer by setting a target activation β_k, fixed to 0.1, for each hidden unit h_k [38]. The overall objective to be minimized is then the negative log-likelihood from Equation (18) and a penalty on the deviation of the hidden layer from the target activation. Since the hidden layer is made up of sigmoid densities, the overall objective function is:

$$\sum_{n=1}^{N} \log \sum_{\mathbf{h}} P\left(\mathbf{x}^{(n)}, \mathbf{h}\right) + \lambda_3 \sum_{k=1}^{K} \left(\beta_k^{(n)} \log h_k + \left(1-\beta_k^{(n)}\right) \log\left(1-h_k\right)\right), \tag{19}$$

where λ_3 is the hyperparameter trading off the sparsity penalty with the log-likelihood.

5 Experiment

The goal in this experiment was to assess how preference prediction accuracy changes when using the same preference model on three different representations of the same data set. The preference model used, as discussed in Section 3.4, was the l^2 logit, while the three representations were the original variables, low-rank + sparse features, and RBM features. The same experimental procedure was run on each of these three representations, where the first representation acts as a baseline for prediction accuracy, and the next two representations demonstrate the relative gain in preference prediction accuracy when using features.

In addition, we performed an analysis of how the hyperparameters affected design preference prediction accuracy for the hyperparameters used in the PCA, LSD, and RBM feature learning methods. For PCA, the hyperparameter was the dimensionality K of the subspace spanned by the eigenvectors of the PCA method. For LSD, the hyperparameters were the rank penalty λ_1, which affects the rank of the low-rank matrix L, and the sparsity penalty λ_2, which influences the number of non-zero elements in the sparse matrix S, both found in Equation (9). For RBM, the hyperparameters were the sparsity penalty λ_3, which controls the number of features activated for a given input datum, and the overcompleteness factor γ, which defines by what factor the dimensionality of the feature space is larger than the dimensionality of the original variable space, both of which are found in Equation (19).

The detailed experiment flow is summarized below and illustrated in Figure 5:

1. The raw choice data set of pairs of customers and purchased designs, described in Section 3.1, was randomly split 10 times into 70% training, 10% validation, and 20% test sets. This was done in the beginning to ensure no customers in the training sets ever existed in the validation or test sets.
2. Choice sets were generated for each training, validation, and test sets for all 10 randomly shuffled splits as described in Section 3.2. This process created a training data set of 832,000 data points, a validation data set of 104,000 data points, and a testing data set of 225,056 data points, for each of the 10 shuffled splits.
3. Feature learning was conducted on the training sets of customer variables and vehicle variables for a vector of 5 different values of K for PCA features, a grid of 25 different pairs of low-rank penalty λ_1 and sparsity penalty λ_2 for the LSD features, and a grid of 56 different pairs of sparsity λ_3 and overcompleteness γ hyperparameters for RBM features. For PCA features, these hyperparameters were $K \in \{30, 50, 70, 100, 150\}$. For LSD features, these hyperparameters were $\lambda_1 \in \{0.005, 0.01, 0.05, 0.1, 0.5\}$ and $\lambda_2 \in \{0.005, 0.01, 0.05, 0.1, 0.5\}$. For RBM, these hyperparameters were $\lambda_3 \in \{4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0\}$ and $\gamma \in \{0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 2.5, 3.0\}$. These hyperparameter settings were selected by pilot testing large ranges of parameter settings to find relevant regions for upper and lower hyperparameter bounds, with numbers of hyperparameters selected based on computational constraints.
4. Each of the validation and testing data sets were encoded using the feature learning methods learned for each of the 5 PCA hyperparameters K, 25 (λ_1, λ_2) LSD hyperparameter pairs, and 56 (λ_3, γ) RBM hyperparameter pairs.
5. The encoded feature data was combined with the original variable data. Each datum consists of the features concatenated with the original variables, then input into the bilinear utility model. Specifically, for some customer features $\mathbf{h}_u$ and customer variables $\mathbf{x}_u$, we used $\mathbf{h}_{u'}^T := [\mathbf{x}_u^T, \mathbf{h}_u^T]$ to define the new representation of the customer; likewise, for some vehicle features $\mathbf{h}_c$ and vehicle variables $\mathbf{x}_c$, we used $\mathbf{h}_{c'}^T := [\mathbf{x}_c^T, \mathbf{h}_c^T]$ to define the new representation of the customer. Combined with Equation (1), a single data point used for training is the difference in utilities between vehicle p and vehicle q for a given customer r.

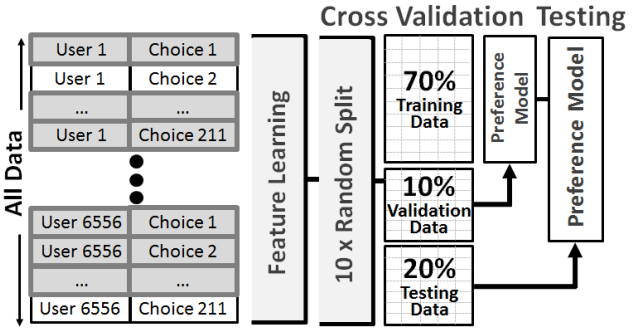

Fig. 5. Data processing, training, validation, and testing flow.

$$\left[\mathbf{h}_{u'}^{(r)} \otimes \left(\mathbf{h}_{c'}^{(p)} - \mathbf{h}_{c'}^{(q)}\right), \mathbf{h}_{c'}^{(p)} - \mathbf{h}_{c'}^{(q)}\right] \tag{20}$$

Note that the dimensionality of each datum could range above 100,000 dimensions for the largest values of γ.

Design Preference Model	Feature Representation	Prediction Accuracy (std. dev.) (ρ-value) $N = 10,000$	Prediction Accuracy (std. dev.) (ρ-value) $N = 1,161,056$
Logit Model	Original Variables (No Features)	69.98% (1.82%) (N/A)	75.29% (0.98%) (N/A)
Logit Model	Principle Component Analysis	61.69% (1.24%) (1.081e-7)	62.03% (0.89%) (8.22e-10)
Logit Model	Low-Rank + Sparse Matrix Decomposition	**76.59% (0.89%) (3.276e-8)**	**77.58% (0.81%) (4.286e-8)**
Logit Model	Exponential Family Sparse RBM	74.99% (0.64%) (2.3e-5)	75.15% (0.81%) (0.136)

Table 3. Averaged preference prediction accuracy on held-out test data using the logit model with the original variables or the three feature representations. Average and standard deviation were calculated from 10 random training and testing splits common to each method, while test parameters for each method were selected via cross validation on the training set.

6. For each of these training sets, 6 logit models were trained in parallel over minibatches of the training data, corresponding to 6 different settings of the l^2 regularization parameter $\alpha = 0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0$. These logit models were optimized using stochastic gradient descent, with learning rates inversely related to the number of training examples seen [51].
7. Each logit model was then scored according to its respective held-out validation data set. The hyperparameter settings ($\alpha_{BASELINE}$) for the original variables, (K_{PCA}, α_{PCA}) for PCA feature learning, (λ_1, λ_2) for LSD feature learning, and ($\lambda_3, \gamma, \alpha_{RBM}$) for RBM feature learning with the best validation accuracy were saved. For each of these four sets of best hyperparameters, Step 3 was repeated to obtain the set of corresponding features on each of the 10 random shuffled training plus validation sets.
8. Logit models corresponding to the baseline, PCA features, LSD features, and RBM features were retrained for each of the 10 randomly shuffled and combined training and validation. The prediction accuracy for each of these 10 logit models was assessed on the corresponding "held out" test sets in order to give average and standard deviations of the design preference predictive accuracy for the baseline, PCA features, LSD features, and RBM features.

5.1 Results

Table 3 shows the averaged test set prediction accuracy of the logit model using the original variables, PCA features, LSD features, and RBM features. Prediction accuracy averaged over 10 random training and held-out testing data splits are given, both for the partial data $N = 10,000$ and the full data $N = 1,161,056$ cases. Furthermore, we include the standard deviation of the prediction accuracies and a 2-sided t-test relative to the baseline accuracy for each feature representation.

The logit model trained with LSD features achieved the highest predictive accuracy on both the partial and full data sets, at 76.59% and 77.58%, respectively. This gives evidence that using features can improve design preference prediction accuracy as the logit model using the original variables achieved an averaged accuracy of 69.98% and 75.29%, respectively. The improvement in design preference prediction accuracy is greatest for the partial data case, as evidenced by both the LSD and RBM, yet the improvement with the full data case shows that the LSD feature learning method is still able to improve prediction accuracy within the capacity of the logit model. The RBM results for the full data case do not show significant improvement in prediction accuracy. Finally, we note a relative loss in design preference prediction accuracy when using PCA as a feature learning method, both for the partial and full data sets, suggesting the heavy assumptions built into PCA are overly restrictive.

The parameter settings for the LSD feature learning method give additional insight to the preference prediction task. In particular, the optimal settings of λ_1 and λ_2 obtained through cross validation on the 10 random training sets was ranged from $r = 29$ to $r = 31$. This significantly reduced rank of the part-worth coefficient matrix given in Eq. (1) suggests that the vast majority of interactions between customer variables and design variables given in Table 1 and Table 2 do not significantly contribute to overall design preferences. This insight allows us to introspect into important feature pairings on a per-customer basis to inform design decisions.

We have shown that even "simple" single-layer feature learning can significantly increase predictive accuracy for design preference modeling. This finding signifies that features more effectively capture the design preferences than the original variables, as features form functions of the original variables more representative of the customer's underlying preference task. This offers designers opportunity for new insights if these features can be successfully interpreted and translated to actionable design decisions; however, given the relatively recent advances in feature learning methods, interpretation and visualization of features remains an open challenge–see Section 6 for further discussion.

Further increases to prediction accuracy might be achieved by stacking multiple feature learning layers, often referred to as "deep learning". Such techniques have recently shown impressive results by breaking previous records in image recognition by large margins [7]. Another possible direction for increasing prediction accuracy may be in developing novel architectures that explicitly capture the conditional statistical structure between customers and designs. These efforts may be further aided through better understanding of the limitations of using feature learning methods for design and marketing research. For example, the large number of parameters associated with feature learning methods results in greater computational cost when performing model selection; in addition to the cross-validation techniques used in this paper, model selection metrics such as BIC and AIC may give further insight along these lines.

6 Using Features for Design

Using features can support the design process in at least two directions: (1) Features interpretation can offer deeper insights into customer preferences than the original variables, and (2) feature visualization can lead to a market segmentation with better clustering than with the original variables. These two directions are still open challenges given the relative nascence of feature learning methods. Further investigation is necessary to realize the above design opportunities and to justify the computational cost and implementation challenges associated with feature learning methods.

The interpretation and visualization methods may be used with conventional linear discrete choice modeling (e.g., logit models). However, deeper insights are possible through interpreting and visualizing features, assuming that features capture more effectively the underlying design preference prediction task of the customer as shown through improved prediction accuracy on held-out data. Since we are capturing "functions" of the original data, we are more likely to interpret and visualize feature pairings such as "eco-friendly" vehicle and "environmentally conscious" customer; such pairing may ultimately lead to actionable design decisions.

6.1 Feature Interpretation of Design Preferences

Similar to PCA, LSD provides an approach to interpret the learned features by looking at the linear combinations of original variables. The major difference between features learned using PCA versus LSD is their different linear combinations; in particular, features learned by LSD are more representative as they contains information from both the data distribution and the preference task, while PCA features only contain information from the data distribution.

As introduced in section 4.2, the weight matrix Ω is decomposed into a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$, i.e. $\Omega = \mathbf{L} + \mathbf{S}$. The nonzero elements in the sparse matrix $\mathbf{S}$ may be interpreted as the weight of the product of its corresponding original design variables and customer variables. As for the low-rank matrix $\mathbf{L}$, features can be extracted by linearly combining the original variable according to the singular value decomposition (SVD) for $\mathbf{L}$. The singular value decomposition is a factorization of the $(m+1)\times n$ matrix $\mathbf{L}$ in the form $\mathbf{L} = \mathbf{U}\Sigma\mathbf{V}$, where $\mathbf{U}$ is a $(m+1)\times(m+1)$ unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers $\sigma_1, \sigma_2, \ldots, \sigma_{\min(m+1,n)}$ on the diagonal, and $\mathbf{V}$ is a $(n)\times(n)$ unitary matrix. Rewriting Equation (6):

$$\begin{aligned} U_{rp} &= \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{L}\mathbf{x}_d^p + \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{S}\mathbf{x}_d^p \\ &= \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{U}\Sigma\mathbf{V}\mathbf{x}_d^p + \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{S}\mathbf{x}_d^p \\ &= \sum_{i=1}^{min(m+1,n)} \sigma_i \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{u}_i\mathbf{v}_i\mathbf{x}_d^p + \left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{S}\mathbf{x}_d^p \end{aligned} \tag{21}$$

where $\mathbf{u}_i$ is the i-th column of matrix U, and $\mathbf{v}_i$ is the i-th row of matrix V. The i-th user feature $\left[\left(\mathbf{x}_c^{(j)}\right)^T, 1\right]\mathbf{u}_i$ is a linear combination of original user variables; the i-th design feature $\mathbf{v}_i\mathbf{x}_d^p$ is a linear combination of original design variables; and σ_i represents the importance of this pair of features for the customer's design preferences.

Interpreting these features in the vehicle preference case study, we found that the most influential feature pairing (i.e., largest σ_i) corresponds to preference trends at the population level: Low price but luxury vehicles are preferred, and Japanese vehicles receive the highest preference while GM vehicles receive the lowest preference. The second most influential feature pairing represents a rich customer group, with preferred vehicle groups being both expensive and luxurious. The third most influential feature pairing represents an elder user group, with their preferred vehicles as large but with low net horsepower.

6.2 Features Visualization of Design Preferences

We now visualize features to understand what insights for design decision making may be had through visual market segmentation. We begin by looking at the utility model U_{rp} given in Equation (1) and note that the inner product between Ω and the variables $\mathbf{x}_u^{(r)}$ representing customer r may be interpreted as customer r's optimal vehicle, denoted $\mathbf{x}_{opt}^{(r)}$:

$$\mathbf{x}_{opt}^{(r)} = \left(\mathbf{x}_u^{(r)}\right)^T \Omega_{out} + \mathbf{1}^T\Omega_{main} \tag{22}$$

where Ω_{out} is the matrix reshaped from the coefficients of Ω corresponding to the outer product given in Equation (1), Ω_{main} is the matrix reshaped from the remaining coefficients, and $\mathbf{1}$ is a vector consisting of 1's with the same dimension as $\mathbf{x}_u^{(r)}$.

We rewrite the utility model U_{rp} given in Equation (1) in terms of the optimal vehicle $\mathbf{x}_{opt}^{(r)}$:

$$U_{rp} = \left(\mathbf{x}_{opt}^{(r)}\right)^T \mathbf{x}_d^p \tag{23}$$

According to the geometric meaning of inner product, the smaller the angle between $\mathbf{x}_d^p$ and $\mathbf{x}_{opt}^{(r)}$ is, the larger will be

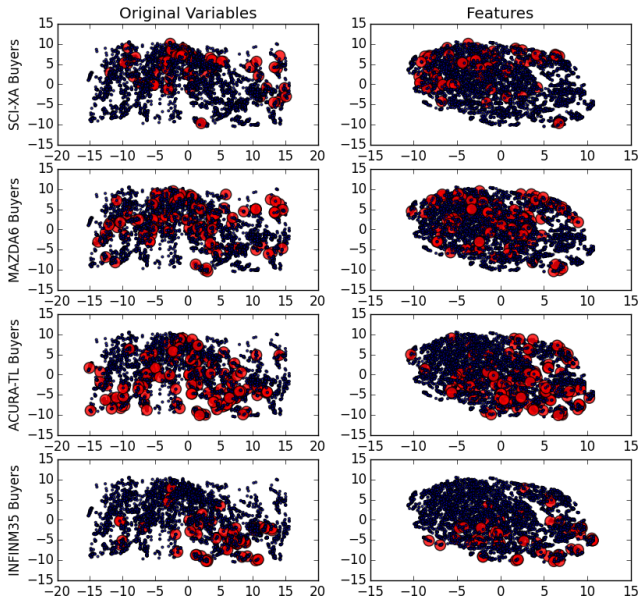

Fig. 6. Optimal vehicle distribution visualization. Every point represents the optimal vehicle for one consumer. In the left column, the optimal vehicle is inferred using the utility model with original variables. In the right column, LSD features are used to infer the optimal vehicle. In the first row, the optimal vehicles from SCI-XA buyers are marked in big red points. Similarly, the optimal vehicles from MAZDA6, ACURA-TL and INFINM35 buyers are marked in big red points respectively.

the utility U_{rp}. In this way, we have an interpretable method of improving upon the actual purchased vehicle design in the form of an 'optimal' vehicle vector. This optimal vehicle vector could be useful for a manufacturer developing a next-generation design from a current design, particularly as the manufacturer would target a specific market segment.

We now provide a visual demonstration of using an optimal vehicle derived from feature learning to suggest a design improvement direction. First, we calculate the optimal vehicle using Equation (22) for every customer in the data set. Then, we visualize these optimal vehicle points by reducing their dimension using t-distributed stochastic neighbor embedding (t-SNE), an advanced nonlinear dimension reduction technique that embeds similar objects into nearby points [52]. Finally, optimal vehicles from targeted market segments are marked in red.

Figure 6 shows the optimal vehicles for the SCI-XA, MAZDA6, ACURA-TL and INFINM35 buyer groups using red points respectively. We observe that the optimal vehicle moves from the left-top corner to the right-bottom corner as the purchased vehicles become more luxurious using the LSD features, while the optimal vehicles in the original variable representation show overlap, especially for MAZDA6 and ACURA-TL buyers. The figure visualizes what has been shown quantitatively through increased preference prediction accuracy; namely, that optimal vehicles represented using LSD features as opposed to the original variables result in a larger separation of various market segments' optimal vehicles.

7 Conclusion

Feature learning is a promising method to improve design preference prediction accuracy without changing the design preference model or the data set. This improvement is obtained by transforming the original variables to a feature space acting as an intermediate step as shown in Figure 1. Thus, feature learning complements advances in both data gathering and design preference modeling.

We presented three feature learning methods, Principal component analysis, low-rank plus sparse matrix decomposition, and sparse exponential family restricted Boltzmann machines, and we applied them to a design preference data set consisting of customer and passenger vehicle variables with heterogeneous unit types, e.g., gender, age, # cylinders.

We conducted and experiment to measure design preference prediction accuracy involving 1,161,056 data points generated from a real purchase dataset of 5582 customers. The experiment showed that feature learning methods improve preference prediction accuracy by 2-7% for a small and full dataset, respectively. This finding is significant, as it shows that features offer a better representation of the customer's underlying design preferences than the original variables. Moreover, the finding shows that feature learning methods may be successfully applied to design and marketing data sets made up of variables with heterogeneous data types; this is a new result as feature learning methods have primarily been applied on homogeneous data sets made up of variables of the same distribution.

Feature interpretation and visualization offer a promise for using features to support the design process. Specifically, interpreting features can give designers deeper insights of the more influential pairings of vehicle features and customer features, while visualization of the feature space can offer deeper insights when performing market segmentation. These new findings suggest opportunities to develop feature learning algorithms that are not only more representative of the customer preference task as measured by prediction accuracy but also easier to interpret and visualize by a domain expert. Methods allowing easier interpretation of features would be valuable when translating the results of more sophisticated feature learning and preference prediction models into actionable design decisions.

Acknowledgments

An earlier conference version of this work appeared at the 2014 International Design Engineering Technical Conference. This work has been supported by the National Science Foundation under Grant No. CMMI-1266184. This support is gratefully acknowledged.The authors would like to thank Bart Frischknecht and Kevin Bolon for their assistance in coordinating data sets, Clayton Scott for useful suggestions, and Maritz Research Inc. for generously making the use of their data possible.

References

[1] Berkovec, J., and Rust, J., 1985. "A nested logit model of automobile holdings for one vehicle house-

holds". *Transportation Research Part B: Methodological,* **19**(4), pp. 275–285. 1, 2

[2] McFadden, D., and Train, K., 2000. "Mixed MNL models for discrete response". *Journal of Applied Econometrics,* **15**(5), pp. 447–470. 1, 2

[3] Reid, T. N., Frischknecht, B. D., and Papalambros, P. Y., 2012. "Perceptual attributes in product design: Fuel economy and silhouette-based perceived environmental friendliness tradeoffs in automotive vehicle design". *Journal of Mechanical Design,* **134**, p. 041006. 1, 2

[4] Norman, D. A., 2007. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books, Mar. 00000. 1

[5] He, L., Wang, M., Chen, W., and Conzelmann, G., 2014. "Incorporating social impact on new product adoption in choice modeling: A case study in green vehicles". *Transportation Research Part D: Transport and Environment,* **32**, pp. 421–434. 1

[6] Sylcott, B., Michalek, J. J., and Cagan, J., 2013. "Towards understanding the role of interaction effects in visual conjoint analysis". In ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V03AT03A012–V03AT03A012. 1, 2

[7] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in Neural Information Processing Systems 25*. pp. 1097–1105. 1, 1, 5.1

[8] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al., 2012. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". *Signal Processing Magazine, IEEE,* **29**(6), pp. 82–97. 1

[9] Smolensky, P., 1986. "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1". MIT Press, Cambridge, MA, USA, ch. Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281. 1, 2.1

[10] Salakhutdinov, R., Mnih, A., and Hinton, G., 2007. "Restricted Boltzmann machines for collaborative filtering". *Proceedings of the 24th International Conference on Machine Learning*, pp. 791–798. 1

[11] Burnap, A., Ren, Y., Gerth, R., Papazoglou, G., Gonzalez, R., and Papalambros, P. Y., 2015. "When crowdsourcing fails: A study of expertise on crowdsourced design evaluation". *Journal of Mechanical Design,* **137**(3), p. 031101. 1

[12] Panchal, J., 2015. "Using crowds in engineering designâtowards a holistic framework". In Proceedings of the 2015 International Conference on Engineering Design, Design Society, pp. 1–10. 1

[13] Chapelle, O., and Harchaoui, Z., 2004. "A machine learning approach to conjoint analysis". *Advances in Neural Information Processing Systems*, pp. 257–264. 1, 2

[14] Evgeniou, T., Pontil, M., and Toubia, O., 2007. "A convex optimization approach to modeling consumer heterogeneity in conjoint estimation". *Marketing Science,* **26**(6), pp. 805–818. 1, 2

[15] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y., 2011. "Unsupervised learning of hierarchical representations with convolutional deep belief networks". *Communications of the Association for Computing Machinery,* **54**(10), pp. 95–103. 1

[16] Wassenaar, H. J., and Chen, W., 2003. "An approach to decision-based design with discrete choice analysis for demand modeling". *Journal of Mechanical Design,* **125**, p. 490. 2

[17] Lewis, K. E., Chen, W., and Schmidt, L. C., 2006. *Decision making in engineering design*. American Society of Mechanical Engineers. 2

[18] Michalek, J., Feinberg, F., and Papalambros, P., 2005. "Linking marketing and engineering product design decisions via analytical target cascading". *Journal of Product Innovation Management,* **22**(1), pp. 42–62. 2

[19] Chen, W., Hoyle, C., and Wassenaar, H. J., 2013. *Decision-Based Design*. Springer London, London. 2

[20] Wassenaar, H., Chen, W., Cheng, J., and Sudjianto, A., 2005. "Enhancing discrete choice demand modeling for decision-based design". *Journal of Mechanical Design,* **127**(4), pp. 514–523. 2

[21] Kumar, D., Hoyle, C., Chen, W., Wang, N., Gomez-Levi, G., and Koppelman, F., 2007. "Incorporating customer preferences and market trends in vehicle packaging design". *International Journal of Production Design*. 2

[22] Burnap, A., Hartley, J., Pan, Y., Gonzalez, R., and Papalambros, P. Y., 2015. "Balancing design freedom and brand recognition in the evolution of automotive brand styling". *Design Science*. 2

[23] Orsborn, S., Cagan, J., and Boatwright, P., 2009. "Quantifying aesthetic form preference in a utility function". *Journal of Mechanical Design,* **131**(6), p. 061001. 2

[24] Morrow, W. R., Long, M., and MacDonald, E. F., 2014. "Market-system design optimization with consider-then-choose models". *Journal of Mechanical Design,* **136**(3), p. 031003. 2

[25] Ren, Y., Burnap, A., Papalambros, P., et al., 2013. "Quantification of perceptual design attributes using a crowd". In DS 75-6: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 6: Design Information and Knowledge, Seoul, Korea, 19-22.08. 2013. 2

[26] Toubia, O., Simester, D. I., Hauser, J. R., and Dahan, E., 2003. "Fast polyhedral adaptive conjoint estimation". *Marketing Science,* **22**(3), pp. 273–303. 2

[27] Abernethy, J., Evgeniou, T., Toubia, O., and Vert, J.-P., 2008. "Eliciting consumer preferences using robust adaptive choice questionnaires". *IEEE Transactions on Knowledge and Data Engineering,* **20**(2), pp. 145–155. 2

[28] Tuarob, S., and Tucker, C. S., 2015. "Automated dis-

covery of lead users and latent product features by mining large scale social media networks". *Journal of Mechanical Design,* **137**(7), p. 071402. 2

[29] Van Horn, D., and Lewis, K., 2015. "The use of analytics in the design of sociotechnical products". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* **29**(01), pp. 65–81. 2

[30] Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., and Leifer, L. J., 2005. "Engineering design thinking, teaching, and learning". *Journal of Engineering Education,* **94**(1), pp. 103–120. 2

[31] Friedman, J., Hastie, T., and Tibshirani, R., 2001. *The elements of statistical learning*, Vol. 1. Springer series in statistics Springer, Berlin. 2.1

[32] Gonzalez, R., and Wu, G., 1999. "On the shape of the probability weighting function". *Cognitive psychology,* **38**(1), pp. 129–166. 2.1

[33] Evgeniou, T., Boussios, C., and Zacharia, G., 2005. "Generalized robust conjoint estimation". *Marketing Science,* **24**(3), Aug., pp. 415–429. 2.1

[34] Hauser, J. R., and Rao, V. R., 2004. "Conjoint analysis, related modeling, and applications". *Advances in Marketing Research: Progress and Prospects*, pp. 141–68. 2.1

[35] Papalambros, P. Y., and Wilde, D. J., 2000. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, July. 2.1

[36] Lenk, P. J., DeSarbo, W. S., Green, P. E., and Young, M. R., 1996. "Hierarchical bayes conjoint analysis: recovery of partworth heterogeneity from reduced experimental designs". *Marketing Science,* **15**(2), pp. 173–191. 2.1

[37] Netzer, O., Toubia, O., Bradlow, E. T., Dahan, E., Evgeniou, T., Feinberg, F. M., Feit, E. M., Hui, S. K., Johnson, J., Liechty, J. C., Orlin, J. B., and Rao, V. R., 2008. "Beyond conjoint analysis: Advances in preference measurement". *Marketing Letters,* **19**(3-4), July, pp. 337–354. 2.1, 3.4

[38] Lee, H., Ekanadham, C., and Ng, A. Y., 2008. "Sparse deep belief net model for visual area V2". *Advances in Neural Information Processing Systems 20*, pp. 873–880. 2.1, 4.3

[39] Maritz Research Inc., 2007. Maritz Research 2006 new vehicle customer satisfactions survey. Information online at: `http://www.maritz.com`. 3, 3.1

[40] Chrome Systems Inc., 2008. Chrome New Vehicle Database. Information inline at: `http://www.chrome.com`. 3.1

[41] United States Census Bureau, 2006. 2006 U.S. Census estimates. Information online at: `http://www.census.gov`. 3.1

[42] Shocker, A. D., Ben-Akiva, M., Boccara, B., and Nedungadi, P., 1991. "Consideration set influences on consumer decision-making and choice: Issues, models, and suggestions". *Marketing Letters,* **2**(3), pp. 181–197. 3.2

[43] Wang, M., and Chen, W., 2015. "A data-driven network analysis approach to predicting customer choice sets for choice modeling in engineering design". *Journal of Mechanical Design,* **137**(7), p. 071410. 3.2

[44] Von Neumann, J., and Morgenstern, O., 2007. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press. 3.3

[45] Fuge, M., 2015. "A scalpel not a sword: On the role of statistical tests in design cognition". In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 1–11. 3.4

[46] Fazel, M., 2002. "Matrix rank minimization with applications". PhD thesis. 4.2

[47] Tomioka, R., Suzuki, T., Sugiyama, M., and Kashima, H., 2010. "A fast augmented lagrangian algorithm for learning low-rank matrices". In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 1087–1094. 4.2.1

[48] Liu, G., and Yan, S., 2014. "Scalable low-rank representation". In *Low-Rank and Sparse Modeling for Visual Analysis*, Y. Fu, ed. Springer International Publishing, pp. 39–60. 4.2.1

[49] Parikh, N., and Boyd, S., 2013. "Proximal algorithms". *Foundations and Trends in Optimization,* **1**(3). 4.2.1, 4.2.1, A

[50] Hinton, G. E., 2002. "Training products of experts by minimizing contrastive divergence". *Neural computation,* **14**(8), pp. 1771–1800. 4.3

[51] Bottou, L., 2010. "Large-scale machine learning with stochastic gradient descent". In *Proceedings of COMPSTAT'2010*. Springer, pp. 177–186. 6

[52] van der Maaten, L., 2008. "Visualizing data using t-sne". *Journal of Machine Learning Research,* **9**, pp. 2579–2605. 6.2

A APPENDIX: Proof of Low-Rank Matrix Estimation Guarantee

Though the low-rank matrix is estimated jointly with the decomposed matrices as well as the loss function, an accurate estimation of the low-rank matrix can still be achieved as guaranteed by the bound as in this section. We subsequently provide a variational bound of the divergence of the estimated likelihood from the true likelihood.

To simplify the notation in our proof, we redefine the following notation:

$$(m,n) = dim(\Omega)$$

Before our proof, however, we state the following relevant prepositions.

Proposition 1. *The proximal operator for the nuclear norm $f = \lambda_1 ||\cdot||_*$ is the singular value shrinkage operator $\mathcal{D}_{\lambda_1}$. Consider the singular value decomposition (SVD) of a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ with rank r.*

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$$
$$\mathcal{D}_{\lambda_1}(\mathbf{X}) = \mathbf{U}\mathcal{S}_{\lambda_\mathbf{1}}(\Sigma)\mathbf{V}^T \tag{24}$$

where the soft-thresholding operator $\mathcal{S}_{\lambda_1}(\Sigma) = diag(\{max(s_i - \lambda_1, 0)\}_{i=1,\ldots,\min(m,n)})$. Moreover, $\mathcal{S}_{\lambda_2}(\cdot)$ is also the proximal operator for the l^1 norm.

The matrix decomposition structure of our model builds on the separable sum property [49]:

Proposition 2. *Separable Sum Property*
If f is separable across two variables x and y, i.e., $f(\mathbf{x},\mathbf{y}) = f_1(\mathbf{x}) + f_2(\mathbf{y})$, then,

$$prox_f(\mathbf{x},\mathbf{y}) = (prox_{f_1}(\mathbf{x}), prox_{f_2}(\mathbf{y})) \tag{25}$$

Our proof proceeds as follows. Let us denote the optima of problem (9) as $\mathbf{L}^*$, the gradient of the loss function $l(\cdot)$ w.r.t $\mathbf{L}^*$ as $\nabla_{\mathbf{L}^*} l$, and the matrix minimizing the loss function $l(\cdot)$ as $\mathbf{L}_0$.

We next prove the following theorems: Theorem 2 provides a tight bound on $\nabla_{\mathbf{L}^*} l$. Corollary 1 bounds the estimation error for the learned matrix $\mathbf{L}^*$. Theorem 3 follows by bounding the divergence of likelihood from the true data distribution where $l(\cdot)$ is a likelihood function.

First, we make the weak assumption that the optimization problem given in Equation (9) is strictly convex, since a necessary and sufficient condition is that the saddle points for $l(\cdot)$ and the regularization terms are not overlapping.

Theorem 2. *Loss Function Gradient Bound.*

$$||\nabla_{\mathbf{L}^*} l||_2 \leq \min(m,n)$$

Proof. Under the strictly convex assumption, the stationary point (i.e., the optima $\mathbf{L}^*$ for the optimization problem (9)) is unique. By Lemma 1, iterations of the proximal gradient optimization method $\mathbf{L}_k$ converge to this optima $\mathbf{L}^*$. According to the fixed point equation for $\mathbf{L}$ (Algorithm 1), we have,

$$\mathbf{L}^* = prox_f(\mathbf{L}^* - \eta\nabla_{\mathbf{L}^*} l) \tag{26}$$

Denote $\mathbf{L}^* - \eta\nabla_{\mathbf{L}^*} l$ as $\mathbf{M}$, representing the argument of the proximal operator at the optimal low-rank estimation. The singular value decomposition (SVD) for $\mathbf{L}^*$, $\mathbf{M}$, and $prox_f(\mathbf{M})$ yields,

$$\mathbf{L}^* = \mathbf{U}\Sigma\mathbf{V}^T \tag{27}$$
$$\mathbf{M} = \mathbf{U}_M\Sigma^M\mathbf{V}_M^T \tag{28}$$
$$prox_f(\mathbf{M}) = \mathbf{U}_{prox}\Sigma^{prox}\mathbf{V}_{prox}^T \tag{29}$$

where $\mathbf{U}$, $\mathbf{U}_{prox} \in \mathbb{R}^{m\times r}$; $\mathbf{V}^T$, $\mathbf{V}_{prox}^T \in \mathbb{R}^{r\times n}$; $\Sigma, \Sigma^{prox} \in \mathbb{R}^{r\times r}$ with $\Sigma = diag(\{s_i\}_{i=1,\ldots,r})$, $\Sigma^{prox} = diag(\{s_i^{prox}\}_{i=1,\ldots,r})$. $\mathbf{U}_M \in \mathbb{R}^{m\times m}$, $\mathbf{V}_M^T \in \mathbb{R}^{n\times n}$ and Σ^M is a $m \times n$ rectangular diagonal matrix.

Without loss of generality, assume that $s_1 > s_2 > \cdots > s_r > 0$, i.e., these singular values are distinct and positive, thus ensuring column orderings are unique. Thus, we may assert that $\mathbf{U} = \mathbf{U}_{prox}$, $\mathbf{V} = \mathbf{V}_{prox}$ and $\Sigma = \Sigma^{prox}$ due to the uniqueness of SVD for distinct singular values in $\mathbf{L}^* = prox_f(\mathbf{M})$.

According to Proposition 1,

$$prox_f(\mathbf{M}) = \mathbf{U}_M \max\left(\Sigma^M - \eta\mathbf{I}, \mathbf{0}\right)\mathbf{V}_M^T \tag{30}$$

Note that the dimensionality of $prox_f(\mathbf{M})$ is less than that of the value of M. To bridge the gap between them, we define diagonal sub-matrices Σ_+^M and Σ_-^M. (In other words, we partition Σ^M into two sub-matrices Σ_+^M and Σ_-^M.) For all singular values s_i^M of $\mathbf{M}$, $i = 1,2,\ldots,\min(m,n)$, if $s_i^M - \eta \geq 0$, then s_i^M is a diagonal element of the sub-matrix Σ_+^M, otherwise, s_i^M is a diagonal element of the sub-matrix Σ_-^M. Hence, $max(\Sigma_+^M - \eta\mathbf{I}, \mathbf{0}) = \Sigma_+^M - \eta\mathbf{I}$ and $max(\Sigma_-^M - \eta\mathbf{I}, \mathbf{0}) = \mathbf{0}$.

$$prox_f(\mathbf{M}) = \mathbf{U}_M^+\left(\Sigma_+^M - \eta\mathbf{I}\right)\left(\mathbf{V}_M^+\right)^T$$

where $\mathbf{U}_M^+$ ($\mathbf{V}_M^+$) are left-singular (right-singular) vectors corresponding to Σ_+^M, $\mathbf{U}_M^-$ and $\mathbf{V}_M^-$ are also defined respectively. Again, due to the uniqueness of SVD, we have $\mathbf{U}_M^+ = \mathbf{U}$ and $\mathbf{V}_M^+ = \mathbf{V}$

We now rewrite the SVD formula for $prox(\mathbf{M})$ and $\mathbf{M}$ as,

$$prox(\mathbf{M}) = \mathbf{U}\left(\Sigma_+^M - \eta\mathbf{I}\right)\mathbf{V}^T \tag{31}$$

$$\begin{aligned}\mathbf{M} &= \mathbf{U}_M \Sigma^M \mathbf{V}_M^T \\ &= \mathbf{U}_M^+ \Sigma_+^M (\mathbf{V}_M^+)^T + \mathbf{U}_M^- \Sigma_-^M (\mathbf{V}_M^-)^T \\ &= \mathbf{U} \Sigma_+^M \mathbf{V}^T + \mathbf{U}_M^- \Sigma_-^M (\mathbf{V}_M^-)^T \end{aligned} \tag{32}$$

By definition of $\mathbf{M}$,

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T - \eta \nabla_{\mathbf{L}^*} l \tag{33}$$

Equation (26) and (31) indicates that,

$$\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{U}(\Sigma_+^M - \eta\mathbf{I})\mathbf{V}^T \tag{34}$$

$$\Sigma = \Sigma_+^M - \eta\mathbf{I} \tag{35}$$

By Equation (32), (33), and (35), we have

$$\begin{aligned} -\nabla_{\mathbf{L}^*} l &= \mathbf{U}\left(\Sigma_+^M - \Sigma\right)\mathbf{V}^T + \mathbf{U}_M^- \Sigma_-^M (\mathbf{V}_M^-)^T \\ &= \mathbf{U}\mathbf{V}^T + \frac{1}{\eta}\mathbf{U}_M^- \Sigma_-^M (\mathbf{V}_M^-)^T \end{aligned} \tag{36}$$

Note that every diagonal element s_{-i}^M in Σ_-^M satisfies $0_{-i}^M \leq \eta$. Hence,

$$\begin{aligned} &||\nabla_{\mathbf{L}^*} l||_2 \leq ||\mathbf{U}\mathbf{V}^T||_2 + \frac{1}{\eta}||\mathbf{U}_M^- \Sigma_-^M (\mathbf{V}_M^-)^T||_2 \\ &\leq \lambda_1 \sum_i^r ||U_{:i} V_{:i}^T||_2 + \sum_{j=1}^{\min(m,n)-r} \frac{s_{-i}^M}{\eta} ||[\mathbf{U}_M^-]_{:j} ([\mathbf{V}_M^-]_{:j})^T||_2 \end{aligned} \tag{37}$$

$$\leq \min(m,n) \tag{38}$$

where $\mathbf{U}_{:i}$ or $\mathbf{V}_{:i}$ is the i-th column in matrix $\mathbf{U}$ or $\mathbf{V}$, and $[\mathbf{U}_M^-]_{:j}$ or $[\mathbf{V}_M^-]_{:j}$ is the j-th column in matrix $\mathbf{U}_M^-$ or $\mathbf{V}_M^-$.

Summarizing the proof of Theorem 2, the gradient of the loss function at the estimated low-rank matrix is bounded by a unit ball within the original problem space that has radius of the low-rank regularization parameter . The relaxation of the bound partially comes from the second term in inequality (37). This implies that the bound is tighter if the rank of L^* is increased.

Based on the gradient bound given in Theorem 2, we now bound the estimation error of the learned low-rank matrix $\mathbf{L}^*$. Although the value of the bound is not explicit in this proof, in some cases we are able to explicitly calculate its value.

Corollary 1. *Learned Low-Rank Matrix Estimation Error. The error $||\mathbf{L}^* - \mathbf{L}_0||_2$ is bounded by the diameter of minimum-sized ball that include the following set*

$$\{\mathbf{L} : ||\nabla_{\mathbf{L}} l||_2 \leq \min(m,n)\}$$

Proof. The proof directly follows from Theorem 2 and the fact that $\nabla_{\mathbf{L}_0} l = \mathbf{0}$.

Since the loss function $l(\cdot)$ is convex, the Euclidean norm of its gradient $\nabla_{\mathbf{L}} l$ is non-decreasing as the Euclidean distance $||\mathbf{L} - \mathbf{L}_0||_2$ is increasing.

When the loss function is sharp around its minima, then $\{\mathbf{L} : ||\nabla_{\mathbf{L}} l||_2 \leq \min(m,n)\}$ is a small region which implies that $\mathbf{L}^*$ is a good estimation of $\mathbf{L}_0$.

We next bound the likelihood divergence when the loss function $l(\cdot)$ is a likelihood function. To do this, we use Theorem 2 and Corollary 1 to construct a variational bound.

Theorem 3. *Variational Bound on Estimated Likelihood*

$$|\triangle l| \leq \min(m,n)\, ||\mathbf{L}^* - \mathbf{L}_0||_2$$

Proof. By the Lagrangian mean value theorem, there exists $\mathbf{L}_1 \in \{\mathbf{L} : \mathbf{L}_{ij} \in \left[\mathbf{L}_{ij}^*, \mathbf{L}_0^{ij}\right]\}$ such that,

$$\begin{aligned} l(\mathbf{L}^*; \mathbf{X}, \mathbf{S}) - l(\mathbf{L}_0; \mathbf{X}, \mathbf{S}) &= \langle \nabla_{\mathbf{L}_1} l, (\mathbf{L}^* - \mathbf{L}_0) \rangle \\ &\leq ||\nabla_{\mathbf{L}_1} l||_2 ||\mathbf{L}^* - \mathbf{L}_0|| \end{aligned} \tag{39}$$

where $\langle \mathbf{A}, \mathbf{B} \rangle$ denotes inner product of $vec(\mathbf{A})$ and $vec(\mathbf{B})$, in which $vec(\cdot)$ is the matrix vectorization operator.

Because of the convexity of $l(\cdot)$, $||\nabla_{\mathbf{L}_1} l||_2 \leq ||\nabla_{\mathbf{L}^*} l||_2$. By Theorem 2,

$$|\triangle l| = l(\mathbf{L}^*; \mathbf{X}, \mathbf{S}) - l(\mathbf{L}_0; \mathbf{X}, \mathbf{S}) \tag{40}$$

$$\leq \min(m,n)\, ||\mathbf{L}^* - \mathbf{L}_0||_2 \tag{41}$$

Summarizing the proof of Theorem 3, the variational bound of the estimated likelihood depends on both the bound of gradient of the likelihood function $l(\cdot)$ given in Theorem 2 and the property of the likelihood function in the neighborhood of its optima $\mathbf{L}_0$ as described by Corollary 1.

Notation	Description	Notation	Description
$\mathbf{x}_c$	Customer	$\|\|\cdot\|\|$	l^2 Norm
$\mathbf{x}_d$	Design	$\|\|\cdot\|\|_1$	l^1 Norm
$\mathbf{h}$	Features	$\|\|\cdot\|\|_*$	Nuclear Norm
N	Number of Data Points	$\mathbf{L}$	Low-Rank Matrix
M	Dimension of Original Variables	$\mathbf{S}$	Sparse Matrix
K	Dimension of Features	λ_1	Nuclear Norm Regularization Parameter
n, m, k	Indices for N, M, K	λ_2	Sparsity Regularization Parameter (LSD)
p, q	Indices for Arbitrary Design Pair	λ_3	Sparsity Regularization Parameter (RBM)
j	Index for Arbitrary Customer	$\mathbf{u}, \mathbf{v}$	Arbitrary Vectors (used for Proof)
$y_{(jpq)}$	Indicator Variable of Purchased Design	$prox_f(\cdot)$	Proximal Operator for f
$\mathbf{X}_c$	All Customers	$\mathbf{U}, \Sigma, \mathbf{V}$	Matrices of SVD Decomposition
$\mathbf{X}_d$	All Designs	s_i	Singular Value
$\mathbf{y}$	All Purchase Design Indicators	t	Step Index for Proximal Gradient
α	l^2 Norm Regularization Parameter	η	Step Size for Proximal Gradient
ω, Ω	Part-Worth Coefficients of Preference Model: Vector, Matrix	E_G, E_B, E_C	Energy Function for Gaussian, Binary, and Categorical Variables
ε_{jp}	Gumbel Random Variable	M_G, M_B, M_C	Dimension of Gaussian, Binary, and Categorical Original Variables
T	Transpose	Z_m	Dimension of Categorical Variable
$[\cdot, \cdot]$	Vector Concatenation	w_{mk}	Weight Parameter for RBM
$\otimes$	Outer Product	a_k, b_m	Bias Parameter for RBM Units
$\mathcal{S}_s(\cdot)$	Soft-Threshold Operator	δ	Delta Function
$\mathcal{D}(\cdot)$	Singular Value Threshold Shrinkage Operator	σ	Sigmoid Function
$P(\cdot)$	Probability	r	Rank of Low-Rank Matrix
$\mathbb{E}[\cdot]$	Expectation	γ	Feature / Original Variable Size Ratio
θ	Parameters (Generic)	β	Sparsity Activation Target for RBM

Table 4. Notation and description of symbols used in this study.