Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2012
August 12-15, 2012, Chicago, United States

DETC2012-70624

ON THE USE OF ACTIVE LEARNING IN ENGINEERING DESIGN

Yi Ren*
Optimal Design Laboratory
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan, 48104
Email: yiren@umich.edu

Panos Y. Papalambros
Optimal Design Laboratory
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan, 48104
Email: pyp@umich.edu

ABSTRACT

Active learning refers to the mechanism of querying users to accomplish a classification task in machine learning or a conjoint analysis in econometrics with minimum cost. Classification and conjoint analysis have been introduced to design research to automate design feasibility checking and to construct marketing demand models, respectively. In this paper, we review active learning algorithms from computer and marketing science, and establish the mathematical commonality between the two approaches. We compare empirically the performance of active learning and static D-optimal design on simulated classification and conjoint analysis test problems with labelling noise. Results show that active learning outperforms D-optimal design when query size is large or noise is small.

1 Introduction

Active learning refers to an iterative query process that starts with partial information of the system that needs to be modeled and chooses the next query based on this information. This method potentially helps to reduce the number of queries needed to accurately model the system. In a design context, such a "system" can be a human user and the user's unknown preference or expertise for a design is the system behavior to be modeled.

The active learning discussion in this paper focues on two machine-learning problems of particular interest to the engineering design community. The first problem is related to classification: As demonstrated in [1–3], classification algorithms have been adopted in design automation research for machine understanding of the difference between valid and invalid designs. Within the various classification techniques, Support Vector Machines (SVM) [4] are of special interest due to their nonlinear classification capability and low computation cost in training. Labeling designs is usually costly, especially when the labels are assigned based on simulation results or expert insight. The high labeling cost motivates exploration of active learning, i.e., finding a query algorithm that determines the next sample to label based on current labeled samples so that a good classifier can be derived with a small number of queries.

The second problem is related to choice-based conjoint (CBC) analysis, used to incorporate consumer preferences in design decision making. When revealed data, e.g., previous consumer purchase records, are unavailable, stated data from surveys need to be collected. CBC creates a preference model for a population based on the collected data. Studies have shown that CBC surveys with large numbers of queries (questions) cause fatigue; subjects resort to simple heuristics in responding to queries and survey results become unreliable [5]. The importance of an engaging interaction [6] motivates the development of active learning algorithms in CBC [7–9]. Such algorithms generate the most relevant queries in real time so that a good preference model can be estimated with a small number of queries.

The present study shows that these two problems are mathematically the same. Several active learning algorithms developed in machine learning and marketing science are then discussed.

*Address all correspondence to this author.

Further, we compare the performance of active learning and D-optimal design (a static sampling method) on a set of simulation tests. Empirical results show that active learning offers advantages if we can afford a lager number of queries, while D-optimal design is more favorable when the query size is small and noise is large.

In the discussion that follows, the *design space* denoted as $\mathscr{D}$ is the input space containing training samples. The *feature mapping* denoted as $\mathbf{v}(\cdot) : \mathscr{D} \to \mathscr{F}$ maps a sample $\mathbf{x} \in \mathscr{D}$ to a *feature space* $\mathscr{F}$, where linear classification can be performed [4]. The term "feature" is used in the computer science context and may or may not correspond to an actual physical feature of the design as understood in the engineering literature. The candidate set to be labeled is represented by $\mathscr{X}$ and the corresponding feature set is $\mathscr{V} := \{\mathbf{v}(\mathbf{x}_i)\}_{i=1}^N$ in the classification problem. The *feature difference* is defined as $\mathbf{z} = \mathbf{v}(\mathbf{x}_1) - \mathbf{v}(\mathbf{x}_2)$. With N designs, the total number of pairwise comparisons is $M = N(N-1)/2$, and we define $\mathscr{Z} := \{\mathbf{z}_i\}_{i=1}^M$ as the candidate set of feature differences. The model parameters and their estimators are denoted as $\mathbf{w}$ and $\hat{\mathbf{w}}$, respectively, in both the classification and conjoint analysis problems. In addition, we often use $\mathbf{v}_i$ as shorthand for $\mathbf{v}(\mathbf{x}_i)$.

The rest of the paper is structured as follows: Section 2 formulates both the classification and the conjoint analysis problem. Section 3 reviews active learning algorithms developed for both these problems, and discusses their similarities. Active learning and D-Optimal design algorithms are studied in a variety of test settings in Section 4, followed by conclusions in Section 5.

2 An SVM Formulation of Classification and Conjoint Analysis

We show that classification and conjoint analysis can be formulated as the same mathematical problem using SVM.

2.1 Classification

In classification problems we are given points $\mathbf{x}$s sampled in a space $\mathscr{D}$ and their categorical labels. In binary classification, the labels will be either "1" or "-1". We will focus our discussion on binary classification since this is the most common scenario in engineering applications. The objective of a binary classification is to find a classifier, a function defined on $\mathscr{D}$, that separates the two classes. In this study, we assume the following classifier model:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{v}(\mathbf{x}) + \varepsilon. \tag{1}$$

Here ε is an error term. It represents the randomness in labeling. For example, a design can be labeled as invalid by chance when it is in fact valid. The effect of ε on active learning performance will be evaluated in Section 4. The label "1" is assigned to a design $\mathbf{x}$ when $f(\mathbf{x}) > 0$ and label "-1" when $f(\mathbf{x}) < 0$. For a given data set $\{\mathbf{x}_i\}_{i=1}^n \subset \mathscr{X}$ and associated labels $\{y_i\}_{i=1}^n$, we can derive the estimator $\hat{\mathbf{w}}$ by solving the following convex problem:

$$\begin{aligned} (P1) \quad \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n \xi_i \\ \text{subject to:} \quad & y_i \mathbf{w}^T \mathbf{v}_i \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad \forall i = 1, ..., n, \end{aligned}$$

where C is an algorithmic parameter that weights the importance of training error.

2.2 Conjoint analysis

We focus on choice-based conjoint analysis with pairwise comparisons, where the user is shown two designs in each iteration and asked to select the better design according to her preference. The user preference (or utility) can again be modeled by Equation (1), where ε in this case represents the randomness in comparison, i.e., the user may make a choice inconsistent with his preference. Without loss of generality, we assign a label "1" to the pair when the first design is preferred over the second, and label "-1" in the opposite case. For a given set of pairs $\{\mathbf{z}_i\}_{i=1}^m \subset \mathscr{Z}$ and their comparison labels $\{y_i\}_{i=1}^m$, we can derive the estimator $\hat{\mathbf{w}}$ of the preference model by solving the following convex problem:

$$\begin{aligned} (P2) \quad \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n \xi_i \\ \text{subject to:} \quad & y_i \mathbf{w}^T \left(\mathbf{v}(\mathbf{x}_i) - \mathbf{v}(\mathbf{x}_j)\right) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad \forall i = 1, ..., n. \end{aligned}$$

By comparing $(P1)$ and $(P2)$, we see that classification and conjoint analysis problems can be similarly formulated, with the only difference being that a query contains a design feature vector $\mathbf{v}$ in classification while it contains a feature difference $\mathbf{z}$ in conjoint analysis. In this paper, we explicitly introduce a nonlinear feature mapping:

$$v_k(\mathbf{x}) = \exp(-\lambda ||\mathbf{x} - \mathbf{x}_k||^2), \tag{2}$$

where λ is the Gaussian spread. The mapping in Equation (2) is used for both classification and conjoint analysis tests. In the case of classification, we set $\mathbf{x}_k$ for $k = 1, ..., N$ to be all the candidates, so that the candidate set for classification, $\mathscr{V}$, contains N design feature vectors $\mathbf{v}_i$, $i = 1, ..., N$ and each $\mathbf{v}_i$ has N features (dimensions). Following the same setting, in conjoint analysis we have the difference candidate set $\mathscr{Z}$ with M elements

$\mathbf{z}_i,\ i = 1,...,M$, where the kth element in the ith candidate is

$$z_{ik} = \exp(-\lambda||\mathbf{x}_{i1} - \mathbf{x}_k||^2) - \exp(-\lambda||\mathbf{x}_{i2} - \mathbf{x}_k||^2).$$

We will show in Section 4 that this explicit feature mapping can work well on the tested nonlinear models.

2.3 Comparison between machine learning and traditional conjoint analysis formulations

Conjoint analysis is traditionally performed using logistic regression. We will show here that we can use Equation (2) instead. To simplify the discussion, we do not consider a hierarchical model of $\mathbf{w}$. Readers may refer to [10] for a detailed comparison between conjoint analysis using machine learning and a hierarchical Bayes model.

When using a multivariate logit model, the negative log-likelihood for comparison data $\{y_i\}_{i=1}^m$ and $\{\mathbf{z}_i\}_{i=1}^m$ can be written in the form of a loss function:

$$L_{\text{logit}} = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{z}_i)\right), \tag{3}$$

An estimation of $\mathbf{w}$ can be derived by minimizing L_{ML}. For comparison purposes, we can rewrite Equation (2) to minimize the following L_{SVM}:

$$L_{\text{SVM}} = \sum_i \max\{0, 1 - y_i \mathbf{w}^T \mathbf{z}_i\} + C\mathbf{w}^T \mathbf{w}, \tag{4}$$

We show below that minimizing Equation (4) has the same goal as minimizing Equation (3), thus justifying the usage of Equation (2). Both problem formulations try to minimize training error penalties. The single penalty terms for violation to one observation are shown in Equations (5) and (6). To simplify, we only compare the case where $y = 1$ and denote $u = -\mathbf{w}^T \mathbf{z}$. Figure 1 shows the penalty curves as functions of u: A large u means a large disagreement between the observation and the model. According to the figure, the penalties from L_{logit} and L_{SVM} are similar, therefore minimizing L_{SVM} without the term $C\mathbf{w}^T \mathbf{w}$ achieves the same goal as minimizing L_{logit}.

$$\text{Penalty}_{\text{logit}}(u) = \log(1 + \exp(\theta u)), \tag{5}$$

$$\text{Penalty}_{\text{SVM}}(u) = \max\{0, 1 + u\}. \tag{6}$$

From a machine learning perspective, the term $C\mathbf{w}^T \mathbf{w}$ represents the model complexity. Indeed, $\mathbf{w} = \mathbf{0}$ is the simplest model and the appearance of non-zero elements in $\mathbf{w}$ adds complexity (non-linearity) to the model. It has been shown that minimizing model complexity and training error (penalties) together, as in Equation (2) or Equation (4), provides a better model than minimizing the training error alone [4, 11].

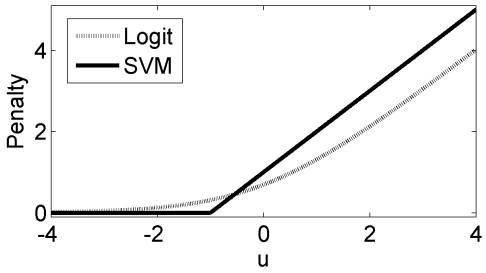

FIGURE 1. $\text{Penalty}_{\text{logit}}$ and $\text{Penalty}_{\text{SVM}}$ on $u \in [-4, 4]$.

2.4 Summary

To summarize, this section introduced the classification and conjoint analysis problems and formulated them using the same machine learning problem (soft-margin SVM). We discussed how the kernel trick, commonly used to deal with the nonlinearity in the model, can fail for pairwise comparison data in conjoint analysis.

The problems and solutions presented in this section can be applied to each iteration during an active learning interaction, using the current data and labels to get the current estimator $\hat{\mathbf{w}}$. In the following section, we discuss querying criteria that will improve this estimation.

3 Annotated Review of Active Learning

Active learning was first introduced in relevance feedback where the goal is to train a machine to understand human concepts using interaction. For example, Tong et al. and Chang et al. [12, 13] trained the search engine to generate more relevant images based on input keywords, and Mandel et al. [14] created a music retrieval system.

In choice-based conjoint analysis, the concept of adaptive query was first introduced in Toubia et al. [7] to replace the non-adaptive D-optimal design. The authors then extended this work to address the zero-volume version space issue which happens when users are prone to fault choices [8]. Later, the concept of adaptive choice-based conjoint was further enhanced with the introduction of statistical learning. Abernethy et al. [15] summarized the two criteria for choosing the next query, which we will review later in this section. It was shown that this adaptive query strategy has robust performance under different user fault choice rates [15].

It should be noted that active learning in machine learning and adaptive query in marketing science are developed independently with little mutual recognition in these two communities. Here we use "active learning" to refer to both lines of research.

3.1 Active learning in classification

Exploitation Recall that in classification, we estimate the classifier parameter $\mathbf{w}$ by solving Equation (2). Tong et al [12] explain the geometrical meaning of Equation (2) as follows: First we define the version space as the feasible space for $\mathbf{w}$, as shown in Equation (7).

$$\mathscr{V} = \{\mathbf{w} \mid \mathbf{w}^T\mathbf{w} = 1,\ y_i(\mathbf{w}^T\mathbf{z}) > 0,\ i = 1...n\}. \tag{7}$$

Essentially, the version space is part of a hypersphere in $\mathscr{F}$ with radius 1, constrained by hyperplanes $y_i(\mathbf{w}^T\mathbf{v}_i) \geq 0$. It can then be shown that the solution of Equation (2), $\hat{\mathbf{w}}$, is the center of the largest hypersphere bounded in $\mathscr{W}$. This is visualized in Figure 2. If the entire candidate set is labeled, the actual version space can be determined. Therefore, an informative query is made when its corresponding hyperplane intersects with $\mathscr{W}$, so that the area of $\mathscr{W}$ can be reduced once the query is labeled. Tong et al. proved that a query that cuts $\mathscr{W}$ into two equal-sized halves will minimize its maximum expected size [16]. In other words, halving $\mathscr{W}$ is a conservative (better than the worst) active learning strategy. According to the geometrical meaning of $\hat{\mathbf{w}}$, it is reasonable to consider $\hat{\mathbf{w}}$ as an approximation of the center of $\mathscr{W}$. Thus bisection can be approximated by querying such a $\mathbf{v}$ that goes through $\hat{\mathbf{w}}$. In practice, the next query $\mathbf{v}$ minimizes the angle $|\hat{\mathbf{w}}^T\mathbf{v}|$. A strategy that approximates $\mathbf{w}$ as the center of $\mathscr{W}$ is called the "simple" algorithm in Tong et al. [16]. This algorithm exploits existing knowledge by querying the next sample on the current decision boundary, i.e., $\hat{\mathbf{w}}^T\mathbf{v} = 0$. The algorithm is intuitive since such a sample cannot be labeled using existing knowledge.

Besides the simple algorithm, Tong et al. [16] also provided another two algorithms rooted in the same motivation: (1) The "maxmin algorithm" approximates the area of two halves of $\mathscr{W}$, denoted as m_+ for $y = 1$ and m_- for $y = -1$, resulting from any query and then chooses the query that maximizes the minimum of the two, i.e., $\mathbf{v} = \arg\max_{\mathbf{v}}\ \min\{m_+(\mathbf{v}), m_-(\mathbf{v})\}$. (2) The "ratio" algorithm takes a similar route as "maximin". It chooses a query that maximizes the minimum of the area ratio and its inverse, i.e.,

$$\mathbf{v} = \arg\max_{\mathbf{v}}\ \min\{\frac{m_+(\mathbf{v})}{m_-(\mathbf{z})}, \frac{m_-(\mathbf{v})}{m_+(\mathbf{v})}\}. \tag{8}$$

Tong et al. [16] show that all three algorithms outperform random queries. While "simple" has the least computational cost, "maximin" and "ratio" have more stable performance across a variety of training data sets.

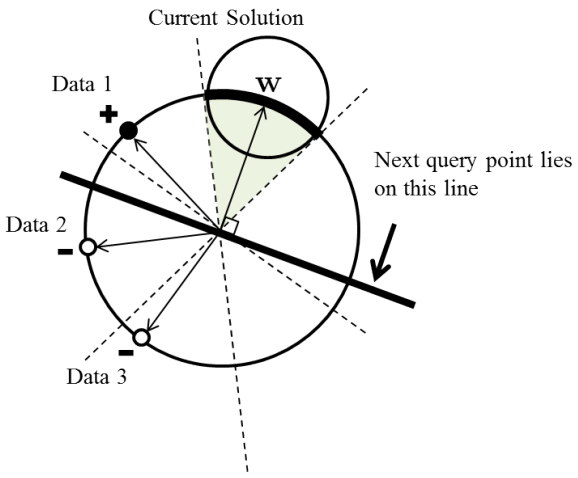

FIGURE 2. Geometrical representation of the version space. In this 2D case, the version space is the highlighted arc of the circle. Each normal vector of a constraining hyperplane represents a sample point $\mathbf{v}_i$ and the label y_i determines which side of the hyperplane is feasible for $\mathbf{w}$. The solution $\hat{\mathbf{w}}$ of a classification problem is the center of the largest hypersphere within the gray cone $y_i(\mathbf{w}^T\mathbf{v}) > 0,\ i = 1...n$. The bisection of the version space $\mathscr{W}$ can be approximated by cutting through the current solution $\hat{\mathbf{w}}$, i.e., $\hat{\mathbf{w}}^T\mathbf{v} = 0$.

Exploration In case the candidate set $\mathscr{V}$ is infinite, it is likely that the exploitation criterion alone is not enough to determine the best next query. In fact, infinite design feature vectors may satisfy $\hat{\mathbf{w}}^T\mathbf{v} = 0$. To this end, exploration criteria are proposed so that the next query can be created by considering both exploitation and exploration. Chang et al. [13] investigated two exploration heuristics, namely, the error-reduction and angle-diversity strategies.

The error-reduction heuristic (see also Roy et al. [17]) attempts to reduce the expected error on future test examples. Denote the set of queried samples and associated labels as L. The distribution of output $\hat{P}_L(y|\mathbf{v})$ for a given $\mathbf{v}$ can be estimated following the trained $\hat{\mathbf{w}}$. If the unknown true distribution is $P(y|\mathbf{v})$, then the expected error of the classifier can be written as

$$E[\text{error}] = \int_{\mathbf{v}} f_{\text{loss}}(P(y|\mathbf{v}), \hat{P}_L(y|\mathbf{v}))P(\mathbf{z})d\mathbf{v}, \tag{9}$$

where the loss function measures the difference between the estimation and its true distribution. In both Chang et al [13] and Roy et al. [17], a log-loss function is used which has the form

$$f_{\text{loss}}(P(y|\mathbf{v}), \hat{P}_L(y|\mathbf{v})) = \sum_{y\in -1,1} P(y|\mathbf{v})\log(\hat{P}_L(y|\mathbf{v})). \tag{10}$$

In practice, the unknown $P(y|\mathbf{v}$ is replaced by $\hat{P}_L(y|\mathbf{v})$ from Equa-

tions (9) and (10). The error-reduction heuristic selects a query $\mathbf{v}^*$ such that $E[\text{error}_{\hat{P}_{L\cup\mathbf{v}^*}}]$ is smaller than $E[\text{error}_{\hat{P}_{L\cup\mathbf{v}}}]$ for any other query $\mathbf{v}$.

As an alternative strategy, the idea of angle-diversity is to select queries close to the decision boundary and also maintain their diversity, which is measured by the angle between queries. For two feature vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the angle between the two can be written as

$$|\cos(\angle(\mathbf{v}_1,\mathbf{v}_2))| = \frac{\mathbf{v}_1^T\mathbf{v}_2}{||\mathbf{v}_1||||\mathbf{v}_2||}. \tag{11}$$

Introducing a weighting parameter r, the angle-diversity algorithm chooses a query that minimizes the merit function

$$f_{\text{merit}}(\mathbf{v}) = |\hat{\mathbf{w}}^T\mathbf{v}| + r\left(\max_{\mathbf{v}^*\in\mathbf{V}} \frac{\mathbf{v}^T\mathbf{v}^*}{||\mathbf{v}||||\mathbf{v}^*||}\right). \tag{12}$$

In Equation (12), the first term requires the next query $\mathbf{v}^*$ to be close to the decision boundary and the second term maximizes the smallest angle between the next query and all existing queries. The angle-diversity heuristic is reported to have the best empirical performance according to [13].

3.2 Active learning in conjoint analysis

Exploitation The same exploitation strategy was independently developed in conjoint analysis from a different background. To start, we first introduce the variance of $\hat{\mathbf{w}}$ derived by McFadden [18], which is asymptotically normal with its mean equal to $\mathbf{w}$ and its covariance matrix equal to the inverse of the information matrix Ω. With responses from n pairwise comparison queries, Ω is given by:

$$\Omega = \mathbf{Z}^T\text{diag}(e_1,...,e_n)\mathbf{Z}, \tag{13}$$

where the ith row of $\mathbf{Z}$ represents the feature difference $\mathbf{z}_i$ from the ith query, and e_i has the form

$$e_i = \frac{\exp(\mathbf{w}^T\mathbf{z}_i)}{(1+\exp(\mathbf{w}^T\mathbf{z}_i))^2}. \tag{14}$$

Therefore reducing the variance of $\hat{\mathbf{w}}$ is equivalent to minimizing the volume of Ω^{-1} or maximize the determinant of Ω. To this end, a common practice is to consider $\hat{\mathbf{w}} = \mathbf{0}$ a priori (assuming that the user will have equal chance to choose any design from a pair) so that $\text{diag}(e_1,...,e_n)$ reduces to an identity matrix. This treatment reduces the problem to maximizing the determinant of $\mathbf{Z}^T\mathbf{Z}$. This criterion produces D-optimal design. For details, see [19–22]. Note that the term "design" in D-optimal design refers to the static choice of queries from $\mathscr{Z}$ and is different from the meaning of design as $\mathbf{x}$.

Researchers have proposed using a non-zero estimator of $\mathbf{w}$ in D-optimal design, where the estimator can be derived from a set of pre-test subjects or from prior belief ([19, 20, 23, 24]). With prior knowledge $\hat{\mathbf{w}}_0$, one can minimize the covariance volume by choosing a pair of designs that have balanced (equivalent) utilities. To elaborate, notice that e_i achieves its maximum when the utilities of the pair are balanced, i.e., $e_i \leq 1$ with the equality achieves if and only if $\mathbf{w}^T\mathbf{z}_i = 0$. This criterion is called "utility balance" [15]. In practice, when the a priori knowledge $\hat{\mathbf{w}}_0$ is close to $\mathbf{w}$, subjects will find the pair of designs with balanced utility hard to differentiate. The queries generated this way are called aggregate customization designs. The active learning strategy in conjoint analysis employs the same concept. Instead of using a fixed prior knowledge $\hat{\mathbf{w}}_0$, active learning finds the next closest query to balance the utility with a continuously updating estimator $\hat{\mathbf{w}}$.

Exploration The exploration criterion for active learning in conjoint analysis is to reduce the variance in the estimator through an iterative process. Abernethy et al. [15] showed that at some iteration s, a good approximation of the estimation covariance Ω is the regularized information matrix

$$\mathbf{H}_s = C_s\mathbf{I} + \mathbf{Z}_s^T\mathbf{Z}_s, \tag{15}$$

where C_s is a weighting parameter and rows of $\mathbf{Z}_s$ are current queried feature differences. Considering $\mathbf{H}_s$ as an ellipsoid in the space of $\mathbf{z}$, a new query $\mathbf{z}_{s+1}$ will be a good query if it morphs $\mathbf{H}_s$ towards a sphere, and increases its determinant. Taking utility balance into consideration, such a $\mathbf{z}_{s+1}$ will be found in the subspace of $\text{span}(\mathbf{H}_s)$ that is perpendicular to the current estimator $\hat{\mathbf{w}}_s$. This subspace can be derived as $\text{span}(\mathbf{Q}_s) = \text{span}(\mathbf{P}_s\mathbf{H}_s)$, where $\mathbf{P}_s = \mathbf{I} - \hat{\mathbf{w}}_s\hat{\mathbf{w}}_s^T/\hat{\mathbf{w}}_s^T\hat{\mathbf{w}}_s$ is a projection matrix. It is suggested that $\mathbf{z}_{s+1}$ be the eigenvector associated with the second minimum eigenvalue of $\mathbf{Q}_s$. Note that $\mathbf{Q}_s$ has at least one zero eigenvalue since it represents a subspace of $\mathscr{W}$. In fact, $\hat{\mathbf{w}}_s/||\hat{\mathbf{w}}_s||_2$ is the eigenvector associated with that zero eigenvalue. Obviously $\hat{\mathbf{w}}_s/||\hat{\mathbf{w}}_s||_2$ is inappropriate to be $\mathbf{z}_{s+1}$ since we require utility balance. Therefore the eigenvector we are looking for should have the second minimum eigenvalue (can be zero) and is thus perpendicular to $\hat{\mathbf{w}}_s$.

We show below that when the projection matrix is not changed by the new query, i.e., $\hat{\mathbf{w}}$ stays the same so that $\mathbf{P}_s = \mathbf{P}_{s+1}$, the above strategy of choosing $\mathbf{z}_{s+1}$ increases $\mathbf{Q}_s$ by 1: At the $s+1$th iteration we have $\mathbf{Q}_{s+1} = \mathbf{Q}_s + \mathbf{P}_s\mathbf{z}_{s+1}\mathbf{z}_{s+1}^T$. Since $\mathbf{z}_{s+1}$ is the eigenvector of $\mathbf{Q}_s$ associated with the second smallest

eigenvalue $\lambda_{\mathbf{Q}}$ and $||\mathbf{z}_{s+1}||_2 = 1$, we have

$$\begin{aligned}\mathbf{Q}_{s+1}\mathbf{z}_{s+1} &= \lambda_{\mathbf{Q}}\mathbf{z}_{s+1} + \mathbf{P}_s\mathbf{z}_{s+1} \\ &= (\lambda_{\mathbf{z}} + 1)\mathbf{z}_{s+1}. \end{aligned} \quad (16)$$

Since the projection matrix usually changes with the accumulating observations, this strategy of choosing $\mathbf{z}_{s+1}$ is a heuristic that tries to increase the determinant of an approximated information matrix. This step is called "minimizing maximum uncertainty" in Abernethy et al..

To summarize, active learning in conjoint analysis has two steps at some iteration s: 1) Compute the estimator $\hat{\mathbf{w}})_s$; 2) Let $\bar{\mathbf{z}}$ be the eigenvector associated with the second smallest eigenvalue of $\mathbf{Q}_s$. Use $\mathbf{z}_0$ in the next query.

Nonetheless, candidates from $\mathscr{Z}$ may not coincide with the eigenvectors of $\mathbf{Q}_s$ in practice. Therefore we choose $\mathbf{z}_{s+1} \in \mathscr{Z}$ that minimizes the following merit function:

$$f_{\text{merit}}(\mathbf{z}) = \frac{|\mathbf{w}^T\mathbf{z}|}{||\mathbf{z}||_2} - r\frac{|\bar{\mathbf{z}}^T\mathbf{z}|}{||\mathbf{z}||_2}, \quad (17)$$

where r is an algorithmic parameter that weights the importance of exploration. With pre-process on $\mathscr{Z}$ to set $||\mathbf{z}|| = 1$, this can be rewritten as

$$f_{\text{merit}}(\mathbf{z}) = |\mathbf{w}^T\mathbf{z}| - r|\bar{\mathbf{z}}^T\mathbf{z}|. \quad (18)$$

The active learning criterion for conjoint analysis is therefore similar to that for classification in Equation (12), although it is derived from a different principle.

3.3 Balancing exploitation and exploration

The strategy we take in the above discussion is to combine exploitation (reducing version space or utility balance) and exploration (angle-diversity or minimization of maximum uncertainty) together as a unified criterion. However, how these two considerations should be weighted is yet to be understood. The unfortunate reality, as demonstrated in Baram et al. [25] is that no fixed active learning strategy will outperform others on any unknown models. To this end, Baram et al. proposed a heuristic master algorithm that chooses from a set of active learning algorithms based on their performance. This approach is analogy to a multi-armed bandit algorithm which determines which arm to try based on current trial success rates. Osugi et al. took a similar approach that flips a biased coin in each iteration to determine whether an exploitation or exploration query shall be made. The change in the estimation is used to measure how successful a query is and the biased coin is updated based on this measurement [26]. A slightly different approach proposed by Basudhar et al. [1] is to query multiple candidates at once where the candidates are driven by either exploitation or exploration motivations.

The purpose of the simulated interaction tests below is to compare the performance of active learning and D-optimal design, and so we ignore tuning the active learning strategy with the weighting parameter r fixed to 1.

4 Simulation Tests and Results

In this section we compare the performance of active learning and D-optimal design in classification and conjoint analysis using simulated interaction tests.

4.1 Algorithm setting

In both active learning and D-optimal design settings, we pre-define the candidate sets $\mathscr{V}$ and $\mathscr{Z}$ according to the discrete design space provided from the test settings. Gaussian bases with a spread $1/p$ where p is the dimensionality of $\mathscr{V}$ (or $\mathscr{Z}$) are used throughout all calculations for generating $\mathscr{V}$ and $\mathscr{Z}$. Both candidate sets are normalized so that each row has L_2 norm of 1.

Active learning setting The following algorithm will be used in both classification and conjoint analysis tests. To avoid redundancy, the algorithm steps are only listed in the classification context.

1. At iteration 0, an initial query $\mathbf{V}_0$ containing two design feature vectors is generated and labelled as $\mathbf{y}_0$; $\hat{\mathbf{w}}_1$ is trained according to Equation (2). In case $\mathbf{y}_0$ contains all "1"s or all "-1"s, a farthest-first query is generated and we jump to step 3. The farthest-first query contains the design feature vector that maximizes the minimum angle from existing ones.
2. At iteration s,
   (a) Obtain $\mathbf{H}_s = C_s\mathbf{I} + \mathbf{V}_s^T\mathbf{V}_s$, where C_s is set to be $1/s$ so that the impact of regularization decreases when more observations are available.
   (b) Denote the plane perpendicular to $\hat{\mathbf{w}}_s$ as $\mathbf{P}_s = \mathbf{I} - \hat{\mathbf{w}}_s\hat{\mathbf{w}}_s^T/\hat{\mathbf{w}}_s)^T\hat{\mathbf{w}}_s$. Project $\mathbf{H}_s$ to span$(\mathbf{P}_s)$ to have: $\mathbf{Q}_s = \mathbf{P}_s\mathbf{H}_s$.
   (c) Find the eigenvector of the second smallest eigenvalue of $\mathbf{Q}_s$, and denote it as $\bar{\mathbf{v}}$.
   (d) Find $\mathbf{v}_{s+1}$ that minimizes $-|\bar{\mathbf{v}}^T\mathbf{v}_{s+1}| + |\hat{\mathbf{w}}_s^T\mathbf{v}_{s+1}|$ (Equation (18) with $r = 1$).
3. Make a query with $\mathbf{v}_{s+1}$ to get y_{s+1}; train with the new data and go back to step 2 until the maximum iteration number is reached.

D-optimal design setting Recall that D-optimal design requires to find $\mathbf{Z}$ such that the determinant of the information

matrix $\mathbf{Z}^T\mathbf{Z}$ is maximized. Finding such optimal designs is not a convex problem and heuristics have been proposed to achieve near-optimal designs [27]. In this study, we produce near-optimal designs simply by enumerating a large number of random $\mathbf{Z}$s given the query sizes, and pick the ones that have largest determinant. Near-optimal $\mathbf{Z}$s are produced separately for different query sizes, i.e., a $\mathbf{Z}$ with 10 queries is not a subset of that with 20 or 50 queries.

4.2 Classification tests

Test settings We compare performance of the proposed algorithm against D-optimal designs on 2D and 5D tests. 2D tests are set to have 10 levels on each dimension and 5D tests have 3. The candidate sets are defined below:

$$\begin{aligned} &\text{2D test candidate set:} \\ &\mathscr{X}_{\text{2D}} = \{[x_1,x_2]|x_{1,2} \in \{1,2,...,10\}\}, \\ &\text{5D test candidate set:} \\ &\mathscr{X}_{\text{5D}} = \left\{[x_1,...,x_5]|x_{1,...,5} \in \{1,2,3\}\right\}. \end{aligned} \tag{19}$$

The test function we use is a weighted summation of Gaussian bases where some of the basis weights are much higher than the others. More specifically, let $\mathbf{x}_i$, $i = 1,...,N$, the complete set of vertices from a discrete space, be the candidate set. Let w_i be the pre-determined weight associated with the ith vertex. We call the vertices with large weights as "class center"s, and denote the set of class centers as $\mathscr{I}$. In both 2D and 5D tests we set one and three vertices as class centers to represent classification tasks of different difficulties. The test function $f_{\text{cls}}(\mathbf{x})$ is defined as follows:

$$\begin{aligned} f_{\text{cls}}(\mathbf{x}) &= \sum_{i=1}^{N} w_i \exp(-||\mathbf{x}-\mathbf{x}_i||), \\ w_i &= \begin{cases} c, & i \in \mathscr{I} \\ -1, & \text{otherwise} \end{cases}, \end{aligned} \tag{20}$$

where c is the class center weight for each test, as listed in Table 1, where test settings are summarized. This setting produces two classes in the given space where the class with label "1" is defined as $\{\mathbf{x}|f_{\text{cls}}(\mathbf{x}) > 0\}$ and that with label "-1" the opposite. We also allow subject choice error through the error term in Equation (1). The low error setting uses a variance $\sigma_\varepsilon = \sqrt{2}/4$ and the high setting uses $\sigma_\varepsilon = 1$. Due to the randomness of choice errors, we conduct 100 independent runs for each test setting. The test performance is measured by the generalization error of

TABLE 1. Preference identification test settings

test ID	#dim	#level	class centers	c
1	2	10	[5,5]	60
2	2	10	[1,1], [5,5], [10,10]	60
3	5	3	[2,2,2,2,2]	200
4	5	3	[1,1,1,1,1], [2,2,2,2,2], [3,3,3,3,3]	80

TABLE 2. Classification test results: mean e

		#query = 10		#query = 20		#query = 50	
		Low	High	Low	High	Low	High
test 1	active	10.0%	10.0%	**0.0%**	**1.8%**	**0.0%**	**1.8%**
	D-opt.	**7.6%**	**7.3%**	5.0%	4.4%	1.3%	2.2%
test 2	active	**14.0%**	**13.8%**	**7.0%**	**7.0%**	**1.6%**	**4.2%**
	D-opt.	17.9%	19.6%	7.8%	9.4%	3.8%	6.0%
test 3	active	**46.1%**	**50.5%**	**46.1%**	**49.8%**	**22.2%**	27.9%
	D-opt.	49.2%	50.0%	50.8%	51.6%	23.1%	**25.4%**
test 4	active	**28.8%**	**28.4%**	29.2%	**25.1%**	9.1%	8.9%
	D-opt.	29.3%	29.1%	**24.9%**	26.3%	**5.9%**	**7.1%**

TABLE 3. Conjoint analysis test results: mean e_p

		#query = 10		#query = 20		#query = 50	
		Low	High	Low	High	Low	High
fn. 1	active	**9.2%**	29.3%	**0.4%**	**19.7%**	**0.0%**	**1.8%**
	D-opt.	11.4%	**20.0%**	16.0%	22.3%	10.1%	25.5%
fn. 2	active	16.9%	31.7%	**14.4%**	29.3%	**1.7%**	**12.8%**
	D-opt.	**14.3%**	**23.4%**	17.1%	**24.2%**	14.2%	32.0%
fn. 3	active	34.8%	34.8%	**21.9%**	31.0%	**8.3%**	18.9%
	D-opt.	**20.7%**	**20.7%**	25.0%	**25.3%**	15.7%	**15.7%**
fn. 4	active	**22.3%**	32.2%	26.5%	30.3%	**13.4%**	**18.5%**
	D-opt.	24.1%	**29.5%**	**19.0%**	**29.5%**	32.7%	38.7%

classification e, defined as

$$e = \frac{\text{number of candidate points that are labeled wrong}}{\text{total number of candidates}}. \tag{21}$$

Test results Reported in Table 2 are means of generalization errors under all test settings. The better performance in each test setting is highlighted. The results require some digestion: We see that when designs are of low dimension (2D cases),

the advantage of active learning emerges as the query size increases. However, for difficult classification problems, e.g. Test 3 and 4, the active learning algorithm has similar performance as D-optimal design in test settings with all three different query sizes. The intuitive explanation is that the efficiency of active learning is built upon adequate knowledge of the underlying model. In fact, the performance of Test 3 is so poor under 10 and 20 queries that it is not better than flipping a coin. Instead, although Test 4 has a more complex model (disconnected regions of class "1"), the performance under the same query sizes is better than that in Test 3. This is because in Test 3 with so limited queries it is hard for active learning to find two classes to start with. Therefore the algorithm performs similarly as D-optimal. These observations imply that an algorithm that starts with a small set of D-optimal designs and switches to active learning later could work better than pure active learning.

4.3 Conjoint analysis tests

Test settings Here we compare the performance of the active learning algorithm and D-optimal design in the presence of user choice error. Four 2D test functions are used to simulate linear and nonlinear user preferences. The functions and the candidate set are defined in Equations (23) to (26).

Similar to classification tests, two scenarios with high and low user choice error are tested. To ensure that user choice error (σ_ε) will have the same effect on all test functions, the discretized values of each function are normalized to have zero mean and standard error. Again due to the randomness of choice errors, we conduct 100 independent runs for each test setting. The test performance is measured by the generalization error of pairwise comparison e_p, defined as

$$e_p = \frac{\text{number of pairs that have wrong comparison labels}}{\text{total number of candidate pairs}}. \quad (22)$$

Test results Table 3 presents the performance comparison between active learning and D-optimal design. We highlight the best performer between the two algorithms under two levels of choice error. D-optimal design has better performance than the adaptive algorithms when the query size is small; while the latter starts to take a lead as the number of queries increases. Examining more closely, one will find that while the performance of D-optimal design is not correlated with the query size, that of active learning is almost always improved with extra queries added. A rationale behind these observations is that when only limited queries are available, D-optimal design ensures the diversity of queries and also prevents misled queries to be generated by erroneous choices. On the other hand, when we can afford a larger number of queries, active learning uses the existing knowledge and refines the estimation more effectively. A D-optimal design with more queries may not work better than one with fewer queries. This is an important finding that not only justifies the usage of D-optimal design in short questionnaires but also indicates that when more queries are allowed, one should switch to an active learning algorithm, which is consistent with findings from classification tests.

test function 1 (linear):

$$f(x_1,x_2) = x_1 + x_2, \quad (23)$$

test function 2 (polynomial):

$$f(x_1,x_2) = x_1^2 + x_1x_2 - x_2^3, \quad (24)$$

test function 3 (Branin):

$$f(x_1,x_2) = (x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10, \quad (25)$$

test function 4 (Six-hump Camelback):

$$f(x_1,x_2) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xx_2 + (-4 + 4x_2^2)x_2^2, \quad (26)$$

Design space for all tests:

$$\mathscr{X} = \{[x_1,x_2] | x_{1,2} \in \{-1,-0.5,0,0.5,1\}\}. \quad (27)$$

5 Conclusion

Classification has been used to automate design feasibility checks so that an optimization algorithm "learns" its constraints on-line, therefore facilitating complex system design. Conjoint analysis bridges engineering with marketing to create demand models. We showed that both the binary classification problem and the conjoint analysis problem can be formulated under the same machine-learning framework.

The major challenge in using these techniques for practical engineering design problems is the high cost of labeling, i.e., the cost of checking design feasibility by software or by expert knowledge in the case of classification, or the cost of collecting survey data in the case of conjoint analysis. To this end, an iterative query process that starts with a small set of training data and efficiently refines the estimation based on current knowledge will be effective. The active learning algorithm presented in this paper supports such a query process.

We reviewed a variety of active learning techniques from machine learning and marketing science. The two commonly-used criteria are exploitation, make the most of the learned knowledge, and exploration, avoid being biased by the limited knowledge learned. The two criteria are in conflict, and a variety of heuristics to balance them was discussed. The simulated interaction experiments showed that active learning becomes better than D-optimal design when the number of queries increases,

while D-optimal design is preferred when erroneous labels exist and only a limited number of queries can be made.

REFERENCES

[1] Basudhar, A., and Missoum, S., 2010. "An improved adaptive sampling scheme for the construction of explicit boundaries". *Structural and Multidisciplinary Optimization,* ***42***(4), pp. 517–529.

[2] Malak Jr, R., and Paredis, C., 2010. "Using support vector machines to formalize the valid input domain of predictive models in systems design problems". *Journal of Mechanical Design,* ***132***, p. 101001.

[3] Alexander, M., Allison, J., Papalambros, P., and Gorsich, D., 2010. "Constraint management of reduced representation variables in decomposition-based design optimization". *Proceedings of the 2010 ASME International Design Engineering Technical Conferences*.

[4] Vapnik, V., 1998. *Statistical learning theory*, Vol. 2. Wiley New York.

[5] Lloyd, A., 2003. "Threats to the estimation of benefit: are preference elicitation methods accurate?". *Health Economics,* ***12***(5), pp. 393–402.

[6] Netzer, O., Toubia, O., Bradlow, E., Dahan, E., Evgeniou, T., Feinberg, F., Feit, E., Hui, S., Johnson, J., and Liechty, J., 2008. "Beyond conjoint analysis: Advances in preference measurement". *Marketing Letters,* ***19***(3), pp. 337–354.

[7] Toubia, O., Hauser, J., and Simester, D., 2004. "Polyhedral methods for adaptive choice-based conjoint analysis". *Journal of Marketing Research,* ***41***(1), pp. 116–131.

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

[9] Toubia, O., Evgeniou, T., and Hauser, J., 2007. "Optimization-based and machine-learning methods for conjoint analysis: Estimation and question design". *Conjoint Measurement*, pp. 231–258.

[10] 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.

[11] Cristianini, N., and Shawe-Taylor, J., 2000. *An introduction to support vector machines: and other kernel-based learning methods*. Cambridge University Press New York, NY, USA.

[12] Tong, S., and Chang, E., 2001. "Support vector machine active learning for image retrieval". In Proceedings of the 9th ACM International Conference on Multimedia, ACM, pp. 107–118.

[13] Chang, E., Tong, S., Goh, K., and Chang, C., 2005. "Support vector machine concept-dependent active learning for image retrieval". *IEEE Transactions on Multimedia,* ***2***.

[14] Mandel, M., Poliner, G., and Ellis, D., 2006. "Support vector machine active learning for music retrieval". *Multimedia Systems,* ***12***(1), pp. 3–13.

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

[16] Tong, S., and Koller, D., 2002. "Support vector machine active learning with applications to text classification". *The Journal of Machine Learning Research,* ***2***, pp. 45–66.

[17] Roy, N., and McCallum, A., 2001. "Toward optimal active learning through sampling estimation of error reduction". In Proceedings of the 18th International Conference on Machine Learning, Citeseer, pp. 441–448.

[18] McFadden, D., 1973. "Conditional logit analysis of qualitative choice behavior". *Frontiers in Econometrics*, pp. 105–142.

[19] Arora, N., and Huber, J., 2001. "Improving parameter estimates and model prediction by aggregate customization in choice experiments". *Journal of Consumer Research,* ***28***(2), pp. 273–283.

[20] Huber, J., and Zwerina, K., 1996. "The importance of utility balance in efficient choice designs". *Journal of Marketing Research,* ***33***(3), pp. 307–317.

[21] Kuhfeld, W., Tobias, R., and Garratt, M., 1994. "Efficient experimental design with marketing research applications". *Journal of Marketing Research,* ***31***(4), pp. 545–557.

[22] Kuhfeld, W., 2005. "Marketing research methods in sas". *Experimental Design, Choice, Conjoint, and Graphical Techniques. Cary, NC, SAS-Institute TS-722*.

[23] Kanninen, B., 2002. "Optimal design for multinomial choice experiments". *Journal of Marketing Research,* ***39***(2), pp. 214–227.

[24] Sándor, Z., and Wedel, M., 2001. "Designing conjoint choice experiments using managers prior beliefs". *Journal of Marketing Research,* ***38***(4), pp. 430–444.

[25] Baram, Y., El-Yaniv, R., and Luz, K., 2004. "Online choice of active learning algorithms". *The Journal of Machine Learning Research,* ***5***, pp. 255–291.

[26] Osugi, T., Kun, D., and Scott, S., 2005. "Balancing exploration and exploitation: A new algorithm for active machine learning". In Proceedings of the 5th IEEE International Conference on Data Mining, IEEE Computer Society, pp. 330–337.

[27] Harman, R., and Trnovska, M., 2009. "Approximate d-optimal designs of experiments on the convex hull of a finite set of information matrices". *Math. Slovaca,* ***59***, pp. 693–704.