The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Artificial Intelligence for Smart Grids and Smart Buildings:
Technical Report WS-16-04

Cost-Effective Feature Selection and Ordering for Personalized Energy Estimates

Kirstin Early
Machine Learning Department
Carnegie Mellon University
kearly@cs.cmu.edu

Stephen Fienberg
Department of Statistics
Carnegie Mellon University
fienberg@stat.cmu.edu

Jennifer Mankoff
Human-Computer Interaction Institute
Carnegie Mellon University
jmankoff@cs.cmu.edu

Abstract

Selecting homes with energy-efficient infrastructure is important for renters, because infrastructure influences energy consumption more than in-home behavior. Personalized energy estimates can guide prospective tenants toward energy-efficient homes, but this information is not readily available. Utility estimates are not typically offered to house-hunters, and existing technologies like carbon calculators require users to answer (prohibitively) many questions that may require considerable research to answer. For the task of providing personalized utility estimates to prospective tenants, we present a cost-based model for feature selection at training time, where all features are available and costs assigned to each feature reflect the difficulty of acquisition. At test time, we have immediate access to some features but others are difficult to acquire (costly). In this limited-information setting, we strategically order questions we ask each user, tailored to previous information provided, to give the most accurate predictions while minimizing the cost to users. During the critical first 10 questions that our approach selects, prediction accuracy improves equally to fixed order approaches, but prediction certainty is higher.

Introduction

Home infrastructure impacts energy usage far more than occupant behavior (Dietz et al. 2009), yet most efforts to reduce energy consumption focus on behavior because infrastructure upgrades are often costly. However, this ignores the impact of choosing among potential homes. In particular, 30% of the U.S. population rent, and renters move on average every two years (U.S. Census Bureau 2013). They can potentially choose improved infrastructure more frequently than we can expect homeowners to upgrade. A tangible measure of infrastructure quality is utility cost, often a hidden factor in tenant cost. In addition to guiding apartment-seekers toward energy-efficient housing, utility estimates can also give prospective tenants cost-of-living estimates to help them make informed decisions about where to live. However, utility costs for apartment units are not readily available. Prospective tenants can ask landlords (or current occupants) for typical utility charges, but these estimates may not be appropriate, as occupant behavior can change energy usage by as much as 100% (Seryak and Kissock 2003). Additionally, many prospective tenants (especially first-time renters) might not think to ask these questions.

Since energy consumption depends on home infrastructure (*e.g.*, square footage) and occupant behavior (*e.g.*, preferred temperature), we can learn the relationship between these features and energy consumption through established datasets, like the Residential Energy Consumption Survey (RECS) (U.S. Energy Information Administration 2009). Some information can be extracted automatically from online rental advertisements, while other information must be provided by prospective tenants at various costs (for example, the question of how many windows a home has requires more effort to answer than how many people will live there).

Our approach proceeds in two stages: at training time, we begin with the extractable (*i.e.*, "free") features in a regression model to predict energy usage and use forward selection to add a subset of the costly features. At test time, when we want to make a prediction for a new tenant-apartment example, we initially have only extractable features and must ask users for unknown values. Our dynamic question-ordering algorithm (DQO) chooses the best question to ask next by considering which feature, if its value were known, would most reduce uncertainty, measured by the width of the prediction interval, with a penalty term on that feature's cost.

We validated our question-ordering approach on a test subset of RECS by calculating prediction error, uncertainty, and cost as questions are asked. DQO achieves similar accuracy as several fixed-order baselines after asking only 10 questions (26% of the cost of all features), but the fixed orderings do not approach DQO's certainty until 20 questions.

Related work

Many previous studies consider modeling residential energy usage, either to reduce consumption (*e.g.*, (Van Raaij and Verhallen 1983)) or to inform consumers about their energy usage (*e.g.*, carbon calculators (Pandey, Agrawal, and Pandey 2011)). However, these methods require lots of information that prospective tenants cannot easily provide. We first review past work in modeling residential energy consumption, with emphasis on the information that is assumed to be available. Next, we cover previous work in building

predictive models that reduce the cost of future predictions and in gathering information of various costs to make confident, accurate predictions on a new instance at test time.

Residential energy modeling

Prior work modeling residential energy consumption falls into two categories: top-down and bottom-up analyses (Swan and Ugursal 2009). Top-down approaches model aggregate residential energy consumption on broad characteristics, such as macroeconomic indicators, historical climate data, and estimates of appliance ownership (*e.g.*, (Haas and Schipper 1998; Labandeira, Labeaga, and Rodríguez 2005)). In contrast, bottom-up methods model consumption at a more granular level, using features of individual households. Household features may include the macroeconomic indicators from the top-down approach, as well as occupant-specific features (*e.g.*, (Douthitt 1989; Kaza 2010)). However, past work has not explored the feasibility of estimating energy consumption with the limited information available in rental advertisements, nor how to incorporate additional, costly information from occupants into the estimation.

Cost-effective feature selection and ordering

When we refer to "cost" for this problem, we mean the costs of obtaining values for individual features, as opposed to the costs of types of errors (Elkan 2001) or the costs of acquiring labels for instances (Cohn, Ghahramani, and Jordan 1996).

In supervised learning, we have a training set of samples (each with D features) and labels and want to learn a function to map features to labels. Feature selection, choosing $d < D$ relevant features, improves prediction performance by reducing overfitting and feature acquisition costs for test instances (Guyon and Elisseeff 2003). Many feature selection algorithms do not incorporate feature-specific costs, but such cost measures can represent the true cost of feature sets.

After learning a predictor on a subset of d cost-effective features, making a prediction on a test instance requires gathering feature values, which can be costly, especially if it requires cooperation from users who might stop before completion. In this case, we want to strategically order questions we ask (based on previous answers) to get the most useful information first, while providing predictions on partial information. This way, people receive meaningful predictions without spending much time or effort answering questions.

The test-time feature ordering problem resembles active learning, which assumes *labels* are expensive. Active learning algorithms strategically select which unlabeled points to query to maximize the model's performance (using both labeled and unlabeled data) while minimizing the cost of data collection (Cohn, Ghahramani, and Jordan 1996). Our setting is similar in that we want to make accurate predictions while keeping data collection costs low; however, rather than choosing an *example to be labeled*, we want to choose a *single feature to be entered* (by asking the user a question).

Training-time feature selection Several algorithms incorporate unique feature costs into supervised learning by penalizing the benefit of adding a feature by the cost of that feature. For example, Davis et al. (2006) develop a cost-sensitive decision tree algorithm by modifying the ID3 splitting criterion, maximizing information gain (Quinlan 1986), to include a penalty on feature costs. They apply this method to forensic classification, where events cannot be reproduced and so all features need to be acquired before classification; cost refers to computational overhead for features. Saeedi, Schimert, and Ghasemzadeh (2014) take a related approach to build a low-cost classifier for sensor localization: a greedy algorithm selects the best feature, in terms of contribution to prediction performance and power consumption, and adds it to the model until accuracy meets a threshold; cost in their setting reflects power consumption. Both these methods address classification, rather than the regression task we desire.

Test-time feature ordering At test time, we want to make a prediction on a new example. We have immediate access to some features but have to pay for others. We want to order the features we ask, based on known values, to improve prediction while keeping feature acquisition costs low. He, Daumé III, and Eisner (2012) consider a similar setting, where all features are available for training, and at test time we want an instance-specific subset of features for prediction, trading off feature cost with prediction accuracy. They formulate dynamic feature selection as a Markov decision process. The policy selects a feature to add; the reward function reflects the classifier margin with the next feature, penalized by feature cost. However, this method does not make sequential predictions, and instead only chooses whether to keep getting features or to stop and make a final prediction.

Dynamically ordering questions in a survey, based on previously-answered questions, can be considered an adaptive survey design (ASD). ASD attempts to improve survey quality (higher response rate, lower error) by giving respondents custom survey designs, rather than the same one (Schouten et al. 2013). Usually ASD tries to minimize non-response, and designs involve factors like number of follow-ups, which can be costly. The general technique is to maximize survey quality, while keeping costs below a budget.

Another related area is personalization, where a system suggests items from user preferences. However, the cold-start problem makes it hard to give recommendations at first, when the system knows nothing about the user, including which questions to ask. Sun et al. (2013) present a multiple-question decision tree for movie recommendation, where each tree node asks users for opinions on several movies, rather than just one. This model lets users sooner provide information about movies they have seen, but it is designed to minimize the number of questions and not the amount of effort required to answer questions of varying difficulty.

Most work in test-time feature ordering does not consider the situation of providing predictions with partial information *as* questions are answered, nor does it address the issue of giving measures of prediction uncertainty to users. Our work fills this gap to provide personalized energy estimates with limited, costly information to prospective tenants.

Data

The Residential Energy Consumption Survey (RECS) contains information about home infrastructure, occupants, and energy consumption. We can use this dataset to learn relationships between household features and energy consumption to predict energy usage for prospective tenants. The most recently released RECS was a nationally representative sample of 12,083 homes across the U.S. (U.S. Energy Information Administration 2009). For each household, RECS records fuel consumption by fuel type (*e.g.*, electricity, natural gas) and around 500 features of the home (*e.g.*, age of refrigerator, number of occupants). We restricted our analysis to homes in the same climate zone as our planned deployment location, a subset of 2470 households. We used 90% of these homes for training and the remaining 10% for testing.

Methods

In our problem setting, features have different costs of obtaining, and we want to build models and make predictions that leverage features cost-effectively. Some information is easily found in the rental listing (*e.g.*, number of bedrooms), while other information requires asking users. For example, the number of windows does not appear in listings and would require a prospective tenant to visit each site; consequently, this question has high cost. Other useful features relate to occupant behavior (*e.g.*, preferred temperature). These questions likely remain constant for each user across homes and therefore require asking only once and are cheaper. We categorize costs as low (can be automatically extracted from rental listings), medium (occupant-related; require asking only once), and high (unit-related; must be answered for each apartment and may require a site visit).

We are concerned with cost-effectiveness, both at training time, when we build a model that efficiently uses features of varying costs, and at test time, when we order the costlier questions we ask a user so that we can make confident predictions on limited information about a new instance.

Training time: cost-aware feature selection

A greedy approximation to feature selection, forward selection starts with an empty feature set and, at each iteration, adds the feature that minimizes error (Tropp 2004; Harrell 2001). For this analysis, we started with low-cost features (rather than no features, as in classic forward selection) and minimized leave-one-out cross-validation error with linear regression to add successive higher-cost features. We trained separate models for predicting electricity consumption (on all homes in our climate zone) and for predicting natural gas consumption (on the 75% of homes that use natural gas).

Test time: cost-effective dynamic question-ordering

After learning regression models for energy usage on the selected features from our training set, we want to predict energy usage for a prospective tenant in a rental unit. Our approach considers a trajectory of prediction intervals as a user provides information. A prediction interval consists of a lower and upper bound such that the true value lies in this interval with at least some probability (Weisberg 2014). Prediction interval width corresponds to prediction uncertainty: a wider interval means less confidence. We select as the optimal next question the one whose inclusion most reduces the expected value of the prediction interval width; that is, it most reduces the expected uncertainty of the next prediction.

In this problem, there are features that are unknown (not yet supplied by the user). We use k nearest neighbors (kNN) (Cover and Hart 1967) to supply values for unanswered features in vector $x \in \mathbb{R}^d$. For each unknown feature f, we find the k data points in the training set $X \in \mathbb{R}^{n \times d}$ (n samples, each d-dimensional) that are closest to x, along dimensions $\mathcal{K}$ that are currently known. Then we estimate x_f as z_f, the mean or mode of feature f in the k nearest neighbors.

Because these z values estimate unknown features of x, we use the measurement error model (MEM) (Fuller 2009) to capture error associated with estimated features. Unlike traditional regression models, MEMs do not assume we observe each component x_f exactly; there is an error δ_f associated with the estimation: $z_f = x_f + \delta_f$, where $\mathbb{E}[\delta_f | x_f] = 0$.

Prediction $\hat{y}$ still depends on the *true, unobserved* value x: $\hat{y} = \hat{\beta}^T \bar{x} = \hat{\beta}(\bar{z} + \bar{\delta})$, where $\hat{\beta} \in \mathbb{R}^{d+1}$ is the parameter vector learned on the training set X (recall all feature values are known at training time). The notation $\bar{x}, \bar{z}, \bar{\delta}$ means vectors x, z have a 1 appended to them and δ a 0 to account for the constant term in the regression. Let $\bar{X}$ extend this notion to the training matrix: $\bar{X} = [\mathbb{1}^n X]$. We can calculate a $100(1-\alpha)\%$ prediction interval for a new point z as

$$\hat{y} \pm t_{n-d-1;\alpha/2} \cdot \sqrt{\hat{\sigma}^2 \left(1 + \bar{z}^T (\bar{X}^T \bar{X})^{-1} \bar{z} + \bar{\delta}^T (\bar{X}^T \bar{X})^{-1} \bar{\delta}\right)}, \quad (1)$$

where the $\bar{\delta}^T (\bar{X}^T \bar{X})^{-1} \bar{\delta}$ term accounts for error from estimated features and $t_{n-d-1;\alpha/2}$ is the value at which a Student's t distribution with $n-d-1$ degrees of freedom has cumulative distribution function value $\alpha/2$. We can estimate δ from training data by calculating the error of predicting each feature with kNN, from the other features. We also estimate $\hat{\sigma}^2$, the regression variance, from training data.

Then, we cycle through each candidate feature f and compute the expected prediction interval width $\mathbb{E}[w(f)]$ for asking that feature next, over each value r that feature f might take on from its range of potential values R:

$$\mathbb{E}[w(f)] = 2 \cdot t_{n-d-1;\alpha/2} \sum_{r \in R} p(z_f = r) \cdot \sqrt{\hat{\sigma}^2 \left(1 + \bar{z}_{f:=r}^T (\bar{X}^T \bar{X})^{-1} \bar{z}_{f:=r} + \bar{\delta}_{f:=0}^T (\bar{X}^T \bar{X})^{-1} \bar{\delta}_{f:=0}\right)}, \quad (2)$$

where $p(z_f = r)$, the probability that the f-th feature's value is r, is calculated empirically from the training set, and the notation $\bar{u}_{f:=q}$ means the f-th component of u is replaced with the value q. Including the feature that attains the narrowest expected prediction interval width $\mathbb{E}[w(f)]$ will reduce the uncertainty of our prediction more than any other feature. This approach allows incorporation of feature cost

into the question selection, by weighting the expected prediction interval width by the cost of acquiring the feature:

$$f^\star = \arg\min_f \left(\mathbb{E}[w(f)] + \lambda \cdot c_f\right),$$

where c_f is the cost of feature f and $\lambda \in \mathbb{R}$ trades off feature cost with reduced uncertainty. A high-cost feature might not be chosen, if another feature can provide enough improvement at lower cost. We ask for this information, update our vector of known data with the response (and estimate the unknown features again, now including the new feature in the set for kNN prediction), and repeat the process until all feature values are filled in (or the user stops answering). Algorithm 1 formalizes this dynamic question-ordering (DQO) process.

Algorithm 1 Dynamically choosing a question-ordering $\mathcal{A}$ and making a sequence of predictions $\hat{y}$ at the current feature values and estimates as feature values are provided

Input: $X \in \mathbb{R}^{n\times d}$, $x \in \mathbb{R}^d$, $\mathcal{K} \subseteq \{1,...,d\}$, $k \in \mathbb{Z}^+$, $\delta \in \mathbb{R}^d$, $\alpha \in [0,1]$, feat_ranges, feat_proportions, $\hat{\beta} \in \mathbb{R}^{d+1}$, $\hat{\sigma}^2 \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $c \in \mathbb{R}^d$
Output: $\mathcal{A} \subseteq \{1,...,d\}$, $\hat{y} \in \mathbb{R}^{|\mathcal{A}+1|}$

function DQO_ALL(X, x, $\mathcal{K}$, k, δ, α, feat_ranges, feat_proportions, $\hat{\beta}$, $\hat{\sigma}^2$, λ, c)
  $\mathcal{A} \leftarrow \{\}$, $\hat{y} \leftarrow \{\}$
  for $i \in \{1,...,d-|\mathcal{K}|\}$ **do**
    $z \leftarrow$ ESTIMATE_FEATURES$(X, x, \mathcal{K}, k)$
    $\hat{y}_i \leftarrow \hat{\beta}^T z$ ▷ Predict on features and estimates
    $E \leftarrow$ EXPECTED_INTERVAL_WIDTH(X, $\mathcal{K}$, z, δ, $\hat{\sigma}^2$, α, feat_ranges, feat_proportions)
    $f^\star \leftarrow \arg\min_{f\notin\mathcal{K}} (E_f + \lambda \cdot c_f)$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{f^\star\}, \mathcal{K} \leftarrow \mathcal{K} \cup \{f^\star\}$
    $z_{f^\star} \leftarrow x_{f^\star}$ ▷ Ask and receive value for $f^\star$
    $\delta_{f^\star} \leftarrow 0$ ▷ No more uncertainty in $f^\star$
  end for
  $z \leftarrow$ ESTIMATE_FEATURES$(X, x, \mathcal{K}, k)$
  $\hat{y}_{d-|\mathcal{K}|+1} \leftarrow \hat{\beta}^T z$ ▷ Make final prediction
  return $\mathcal{A}, \hat{y}$
end function

More generally, this algorithm can be seen as a framework that predicts on partial information and selects which feature to query next by 1) estimating values for unknown features (here with kNN) and 2) asking for the feature that will most reduce the expected uncertainty of the next prediction (here measured by prediction interval width). With this approach, we strategically order questions, tailored to previous information, to give accurate predictions while minimizing user burden of answering many or difficult questions that will not provide a substantial reduction in prediction uncertainty.

Results

Training time: cost-aware feature selection

We can reasonably assume all low-cost features are available (since these can be extracted from the rental listing). However, for higher-cost groups, we need to know how many features are needed in practice. We analyzed the predictive power of selecting features based on cost. Figure 1 summarizes the fraction of explained variance for various feature groups, on the 20% of the training set used for feature analysis. Although low-cost features explain only 21-35% of variance, the full feature set explains 89-98% of the variance. Figure 2 shows how the degree of variance explained changes as higher-cost features are added in forward selection. The steep trajectory means the first few features added have a big impact on the explained variance, with about 10 costlier features needed to explain 60% of the variance, and 28 costlier features to explain 70% of the variance of electricity usage, on the feature selection subset. Natural gas performance is similar to electricity, as shown in Figure 2.

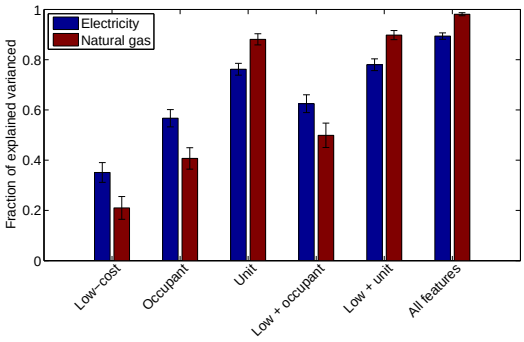

Figure 1: The fraction of variance captured by regression models on sets of RECS features. Occupant features are "mid-cost," and unit features are "high-cost."

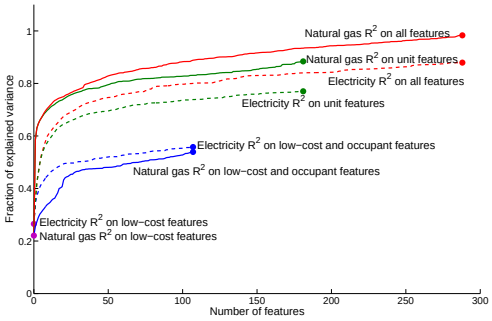

Figure 2: The fraction of explained variance for electricity (dashed) and natural gas (solid) usage as features are added in forward selection, starting with the low-cost features.

Based on this analysis, we chose 60 features (30 each for electricity and natural gas), plus the 18 low-cost features. We selected 30 for each prediction task both because our exploration demonstrates that that many can explain a substantial amount of the variance, and also to avoid overfitting. Table 1 gives examples of features that are most predictive in each cost category for electricity prediction. We then used tenfold cross-validation on the remaining 80% of the training set to learn regression coefficients for the chosen features.

Table 1: Examples of highly-ranked features selected via forward selection to predict electricity consumption.

Extractable (low-cost)	**Occupant (mid-cost)**	**Unit (high-cost)**
Type of housing	Number of TVs	Size of freezer
Square footage	Householder age	Who pays elec.
Cool. deg. days	Number of lights	Elec. heating

Test time: cost-effective dynamic question-ordering

We evaluated the performance of our cost-effective *DQO* algorithm in making sequential predictions with partial, evolving information on a held-out test set. For comparison, we implemented several baselines. The *Oracle* chooses the next best feature according to the minimum true prediction interval width (calculated on the test sample using true feature values, rather than the *expected* width as in Algorithm 1). We tested two versions of DQO and oracle: ordering additional features *without cost* and *with cost* (implemented as $\lambda = 0$ and $\lambda = 0.05$). In addition, we implemented a *Random* algorithm which chooses a random question ordering for each sample; a *Fixed Decreasing* algorithm, which asks questions in decreasing order of feature measurement error δ_f (identical ordering for all samples); and a *Fixed Selection* algorithm, which asks questions in the order of forward selection in the training phase (also identical for all samples).

We calculated several metrics related to the trajectory of prediction performance and cost, for orderings given by the seven algorithms: *DQO* and *Oracle with* and *without cost, Random, Fixed Decreasing*, and *Fixed Selection*. We summarized prediction performance with the width of the current prediction interval (prediction *certainty*) and the absolute value of the difference between the current prediction and the truth (prediction *error*); we also measured the cumulative cost of all features asked at each step (prediction *cost*). We report on performance for two separately trained regressions (for electricity and for natural gas); due to similarity between the two and space constraints, our plots show only electricity. Tables 2 and 3 summarize the metric trajectories for electricity and natural gas as areas under the curve—smaller values are better because they mean the algorithm spent less time in high uncertainty, error, and cost.

Certainty metrics For certainty, we calculated widths of 90% prediction intervals as features were answered. Since narrower prediction interval widths correspond to more certain predictions, we expect DQO interval widths to be less than the baselines, particularly in the early stages. Figure 3 plots the *actual* prediction interval widths as questions are asked (calculated with Equation 1, using known feature values and imputed values for unknown features), averaged across the test dataset, for the question sets from the seven orderings. DQO results in the narrowest (or near-narrowest) prediction intervals, compared to baselines, with improvements most notable in the first 10 questions answered—the situation when users do not answer all questions.

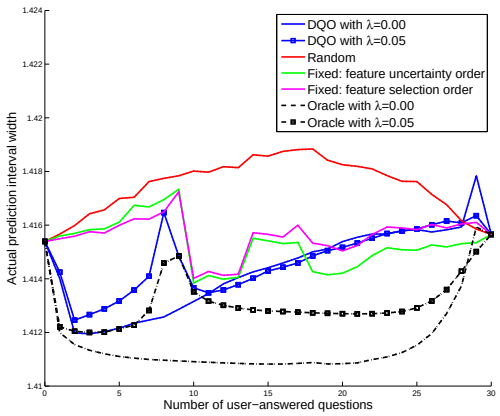

Figure 3: Prediction interval widths as questions are asked.

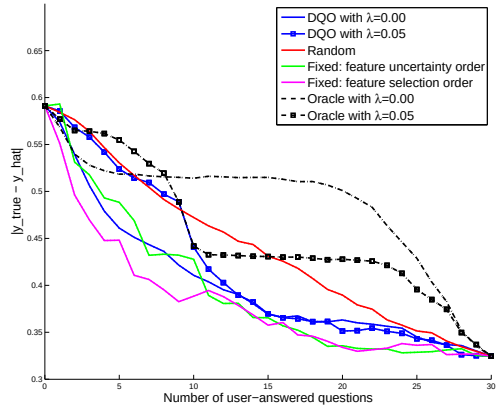

Figure 4: Absolute value of differences between true values and current predictions as questions are asked.

Error metrics For error, we calculated the absolute value of differences between predictions and truth as questions are answered, plotted in Figure 4. For all orderings, predictions near the true value as questions are answered. Once about 10 questions have been asked, *DQO with cost* reaches similar performance as *DQO without cost* and the fixed-order baselines (*Fixed decreasing* and *Fixed selection*).

Cost metrics Progressive total feature costs as features are asked and their true values are used in the models are plotted in Figure 5. Cumulative feature costs are lower for orderings that penalize feature cost (*DQO, Oracle with cost*). The other orderings have similar cost trajectories to each other.

Overall, these metrics show that our test-time DQO approach quickly achieves accurate, confident predictions: by asking around 10 questions, DQO (with and without cost) reaches similar accuracy as the fixed-order baselines, but the sequential predictions by the fixed orderings are less confident than DQO until about 20 questions have been asked.

Table 4 shows how frequently the oracle asked features in

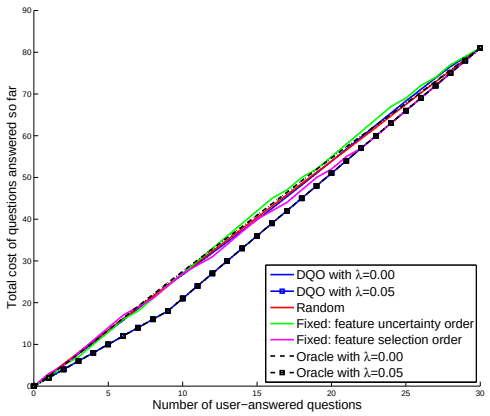

Figure 5: Total feature costs as questions are asked.

Table 2: Electricity: Areas under the curves for the certainty, error, and cost metrics from various methods.

Method	Int. width	$\lvert y - \hat{y} \rvert$	Cost
DQO without cost	42.43	12.06	1212.91
DQO with cost	42.44	12.53	1120.50
Random	42.53	13.18	1213.30
Fixed decreasing	42.46	11.85	1233.50
Fixed selection	42.47	11.45	1190.50
Oracle without cost	42.35	14.62	1222.91
Oracle with cost	42.39	13.63	1120.50

one position, across test instances. Most features are chosen fairly uniformly at each position in the ordering. This indicates that there is no single best order to ask questions across all households, which is why DQO is so valuable.

Future Work

Currently, our DQO algorithm assumes that 1) users are able to answer the next question we ask and 2) their answers are accurate. However, situations could arise where these assumptions do not hold. For example, in utility prediction, a prospective tenant may want energy estimates for a home before visiting—they could still answer occupant-related features. DQO can be easily extended to this case by offering a "don't know" option for answering questions and removing unknown features from consideration in later iterations. Breaking the second assumption, that user answers are accurate, would allow people to give estimates for features. Incorporating this element into DQO would require a way to estimate the error of user-provided feature estimates. Another area for future work is refining feature costs. We categorized costly features as occupant- or unit-related, but other metrics could reflect time or effort to answer a question.

Generalizing DQO framework to other techniques

DQO as presented here uses kNN to estimate values for unknown features and calculates expected prediction uncertainty as prediction interval width to choose the next question to ask. However, we can use other techniques for estimation of yet-unasked questions and calculations of expected uncertainty. For example, to do classification instead of regression, we could use distance from the decision boundary to measure certainty for SVM classification (farther away from boundary is a more certain prediction).

Table 3: Natural gas: Areas under the curves for the certainty, error, and cost metrics from various methods.

Method	Int. width	$\lvert y - \hat{y} \rvert$	Cost
DQO without cost	48.46	13.61	1198.63
DQO with cost	48.49	13.57	1142.00
Random	48.79	13.42	1227.96
Fixed decreasing	48.49	13.27	1224.00
Fixed selection	48.46	12.28	1239.00
Oracle without cost	48.29	13.68	1242.71
Oracle with cost	48.37	13.77	1142.00

Table 4: Frequency features are chosen in a single position.

Threshold (%)	Number of features in one position more than Threshold% of the time	
	Electricity	*Natural gas*
10	30	29
15	15	23
20	3	15
25	0	5
30	0	3
35	0	2
55	0	1
70	0	0

Broadening DQO to adaptive survey design

Dynamically ordering questions can benefit survey design. Unlike paper surveys, online surveys can support adaptive questions, where later questions depend on previous responses. Past work in adaptive questions has taken a rule-based, question-specific approach that means a certain response to a certain question leads to a new set of questions, uniformly across all respondents (Pitkow and Recker 1995; Bouamrane, Rector, and Hurrell 2008). A richer interpretation of adaptive questions would use a dynamic question order, personalized to individual respondents, based on their previous answers. Such an approach could increase engagement and response rate, as well as imputation quality.

Conclusion

Providing personalized energy estimates to prospective tenants with limited, costly information is a challenge. Our solution uses an established dataset to build cost-effective predictive models and, at test time, dynamically orders questions for each user. At training time, we use a cost-based forward selection algorithm to select relevant features from RECS and combine low-cost features that are extractable

from rental advertisements with relevant higher-cost features related to occupant behavior and home infrastructure. At test time, when we want to make a personalized estimate for a new renter-home pair, we present a cost-effective way to choose questions to ask a user about their habits and a rental unit, based on which feature's inclusion would most improve the certainty of our prediction, given the information we already know. Our experiments show that, for predicting electricity and natural gas consumption, we achieve prediction performance that is equally accurate, but more certain, than two fixed-order baselines by asking users only 21% of features (26% of the cost of the full-feature model). This setting, where we know all feature values at training time but must acquire individual features at cost during test time, shows up in other applications, such as conducting tests to make a medical diagnosis, giving personalized recommendations, and administering adaptive surveys.

References

Bouamrane, M.-M.; Rector, A.; and Hurrell, M. 2008. Gathering precise patient medical history with an ontology-driven adaptive questionnaire. In *21st IEEE International Symposium on Computer-Based Medical Systems*, 539–541.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research*.

Cover, T. M., and Hart, P. E. 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* 13(1):21–27.

Davis, J. V.; Ha, J.; Rossbach, C. J.; Ramadan, H. E.; and Witchel, E. 2006. Cost-sensitive decision tree learning for forensic classification. In *ECML 2006*. Springer. 622–629.

Dietz, T.; Gardner, G. T.; Gilligan, J.; Stern, P. C.; and Vandenbergh, M. P. 2009. Household actions can provide a behavioral wedge to rapidly reduce us carbon emissions. *Proceedings of the National Academy of Sciences* 106(44):18452–18456.

Douthitt, R. A. 1989. An economic analysis of the demand for residential space heating fuel in Canada. *Energy* 14(4):187–197.

Elkan, C. 2001. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 973–978.

Fuller, W. A. 2009. *Measurement Error Models*. John Wiley & Sons.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3:1157–1182.

Haas, R., and Schipper, L. 1998. Residential energy demand in OECD-countries and the role of irreversible efficiency improvements. *Energy Economics* 20(4):421–442.

Harrell, F. E. 2001. *Regression Modeling Strategies*. Springer Science & Business Media.

He, H.; Daumé III, H.; and Eisner, J. 2012. Cost-sensitive dynamic feature selection. In *ICML Inferning Workshop*.

Kaza, N. 2010. Understanding the spectrum of residential energy consumption: a quantile regression approach. *Energy Policy* 38(11):6574–6585.

Labandeira, X.; Labeaga, J. M.; and Rodríguez, M. 2005. A residential energy demand system for Spain. *MIT Center for Energy and Environmental Policy Research Working Paper*.

Pandey, D.; Agrawal, M.; and Pandey, J. S. 2011. Carbon footprint: current methods of estimation. *Environmental Monitoring and Assessment* 178(1-4):135–160.

Pitkow, J. E., and Recker, M. M. 1995. Using the web as a survey tool: Results from the second WWW user survey. *Computer Networks and ISDN Systems* 27(6):809–822.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.

U.S. Census Bureau. 2013. American housing survey for the United States.

U.S. Energy Information Administration. 2009. Residential energy consumption survey 2009. www.eia.gov/consumption/residential/data/2009/.

Saeedi, R.; Schimert, B.; and Ghasemzadeh, H. 2014. Cost-sensitive feature selection for on-body sensor localization. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 833–842. ACM.

Schouten, B.; Calinescu, M.; Luiten, A.; et al. 2013. Optimizing quality of response through adaptive survey designs. *Survey Methodology* 39(1):29–58.

Seryak, J., and Kissock, K. 2003. Occupancy and behavioral effects on residential energy use. In *Proceedings of the Solar conference*, 717–722. American Solar Energy Society.

Sun, M.; Li, F.; Lee, J.; Zhou, K.; Lebanon, G.; and Zha, H. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, 445–454. New York, NY, USA: ACM.

Swan, L. G., and Ugursal, V. I. 2009. Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable and Sustainable Energy Reviews* 13(8):1819–1835.

Tropp, J. A. 2004. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on* 50(10):2231–2242.

Van Raaij, W. F., and Verhallen, T. M. 1983. A behavioral model of residential energy use. *Journal of Economic Psychology* 3(1):39–63.

Weisberg, S. 2014. *Applied Linear Regression*. John Wiley & Sons, 4th edition.