Kernel Methods for Revealed Preference Analysis

Sébastien Lahaie [1]

Abstract. In classical revealed preference analysis we are given a sequence of *linear* prices (i.e., additive over goods) and an agent's demand at each of the prices. The problem is to determine whether the observed demands are consistent with utility-maximizing behavior, and if so, recover a representation of the agent's utility function. In this work, we consider a setting where an agent responds to *nonlinear* prices and also allow for incomplete price information over the consumption set. We develop two different kernel methods to fit linear and concave utilities to such observations. The methods allow one to incorporate prior information about the utility function into the estimation procedure, and represent semi-parametric alternatives to the classical non-parametric approach. An empirical evaluation exhibits the relative merits of the two methods in terms of generalization ability, solution sparsity, and runtime performance.

1 INTRODUCTION

The economic theory of demand supposes that an agent, facing prices, will choose to consume the bundle of goods that it most prefers among all bundles that it can afford, according to some well-defined (ordinal) utility function. The basic question of revealed preference analysis is to what extent this behavioral hypothesis can be validated or refuted given observed demand data. If the observations are consistent with utility maximization then the question becomes that of recovering an actual utility function explaining the behavior, perhaps with some useful structure such as concavity, in order to make welfare judgments or forecast future demands [21].

Current revealed preference techniques apply under a model where the agent responds to *linear* prices (i.e., additive over goods). Since linear prices can be succinctly described, it is also implicitly assumed that price information is complete: at each demand observation, the entire price vector is recorded. In this work, we consider a setting where an agent responds to possibly *nonlinear* prices. Since it can be costly to completely record such prices, we also allow for incomplete price information, meaning that the prices of certain bundles may simply be unavailable. Instances of nonlinear pricing abound at both the individual and firm levels, ranging from advertising rates and electricity tariffs to mailing rates, telephone tariffs and airline ticket prices, to name just a few [22].

To extend the applicability of revealed preference analysis to such instances, it is necessary to develop methods that can incorporate incomplete, nonlinear price data to fit a utility function that successfully generalizes. In this work we propose two different kernel methods for this purpose. Following the usual approach in kernel methods we first recode the bundles in terms of a set of features implicitly specified through a kernel function. Our two methods fit linear and concave utility functions, respectively, to the observations mapped into the feature space; in the original bundle space, the resulting utilities can be highly nonlinear and non-concave. We will see that the flexibility afforded by the choice of kernel function can bring substantial improvements in generalization ability, as is the case with more standard applications of kernel ideas in classification and regression.

Another advantage of kernel methods is that they provide well-understood ways to incorporate prior information on utilities through the introduction of regularization terms (among other techniques) [17]. The current nonparametric approaches in the economic literature only introduce priors on the error structure [8, 20]. In fact, we will see that a regularized kernel method trained on incomplete price data can outperform these nonparametric approaches, which draw on complete price data.

The rest of the paper is organized as follows. In the remainder of this section we survey the literature on revealed preference and related work on machine learning methods to fit utilities and ranking data. In Section 2 we provide formal background on revealed preference analysis, and explain why a straightforward adaptation of current nonparametric techniques in economics to nonlinear prices has the potential to generalize poorly. Section 3 describes our two methods and their properties. In Section 4 we report on experiments that evaluate the relative merits of the two methods and compare them to the nonparametric approach in the economics literature.

Related work. In a sequence of papers beginning with Samuelson [16], economists have examined the question of testing whether observed demand data is *rationalizable*, meaning that there exists a utility function that explains the demand behavior. This culminated in the *generalized axiom of revealed preference*, which provides a necessary and sufficient condition for data to be rationalizable [10, 15, 19]. Much has been made of this generalized axiom, because it can be checked in polynomial time using combinatorial algorithms (essentially special cases of network flow algorithms), thus providing a convenient test of rationalizability. However, we will not go into its specification here because we will not make use of it—see Varian [21], who also provides a survey of past and recent research on revealed preference. It is now understood that the equivalence between rationalizability and the generalized axiom can be seen as an instance of linear programming duality [7].

Independently, Afriat [1] provided a cyclical consistency condition which is equivalent to the generalized axiom. More importantly for our work, he formulated a system of inequalities which has a positive solution if and only if the demand data is rationalizable. A solution to the system, if feasible, also immediately defines a utility function rationalizing the data. Afriat's inequalities form the basis of both our kernel method formulations.

Beigman and Vohra [2] consider the problem of rationalizability from the viewpoint of statistical learning theory. They show that without any other assumptions on utility besides monotonicity and

[1] Yahoo! Research, New York; lahaies@yahoo-inc.com

concavity, the sample complexity of learning (in the probably approximately correct sense [12]) a demand and hence utility function is infinite. This provides a formal justification for introducing regularization terms in our methods, because without them the methods could not generalize given a finite amount of data.

The application of kernel methods to recover utility information can be found sporadically in the machine learning literature. Chapelle and Harchaoui [3] and Evgeniou et al. [6] apply support vector machines (SVMs) to conjoint analysis, where the task is to estimate a utility function given choice data (which bundles were chosen when a restricted subset is offered). Their ideas are related to SVMs for ranking [4, 9]. Domshlak and Joachims [5] develop an SVM approach to fit utilities given more intricate, qualitative choice data. Conjoint analysis is a different setting than ours, because in revealed preference analysis the prevailing prices when a bundle is chosen imply a substantial amount about the underlying utilities.

2 BACKGROUND

Consider an agent endowed with a utility function $u : X \to \mathbf{R}$ that encodes its preferences over bundles in its *consumption set* X (i.e., those bundles it can feasibly consume). Throughout we will take $X = \mathbf{R}^m_+$, where m is the number of goods. At prices $p : X \to \mathbf{R}$, the agent will choose to consume a bundle in its *demand set*

$$D(p) = \arg\max_{x \in X} \{u(x) : p(x) \leq b\},$$

where b is the agent's budget. The classical theory of demand deals with linear prices, meaning that $p \in \mathbf{R}^m$ and the price of a bundle x is evaluated according to the usual scalar product $\langle p, x \rangle$. In this work we will allow for general price functions over the consumption set.

A utility function u is *strictly monotone* if $u(x') > u(x)$ for bundles $x' > x$. If utility and prices are both strictly monotone, $x \in D(p)$ implies that $p(x) = b$, meaning an agent always exhausts its budget when choosing a bundle. Throughout we will assume that utilities and prices are strictly monotone; this is a standard assumption in the theory of demand. Note that any strictly monotone transformation of a utility function leaves the underlying preferences unchanged, in the sense that $D(p)$ and hence the behavior of the agent are unaffected. In particular, we can translate utilities by a constant and scale them by a positive factor, and the preferences remain the same.

In revealed preference analysis we are given a sequence of observations $\{(x_i, p_i)\}_{i \in N}$ for $N = \{1, 2, \ldots, n\}$, where x_i is the bundle chosen by the agent when the prevailing prices are p_i. The question is whether there is a utility function u that *rationalizes* the observations, meaning that $x_i \in D(p_i)$ for all $i \in N$; by our arguments above, we can take the budget b_i at observation i to be $p_i(x_i)$. To verify this condition one needs full knowledge of the prices p_i at each observation. We adapt the condition to partial price information as follows. We say that a utility function u is *consistent* with a sequence of demand observations if for each x_i and bundle x whose price $p_i(x)$ was recorded at observation i, we have $u(x_i) \geq u(x)$ if $p_i(x) \leq p_i(x_i)$. We will also make use of the notion of *approximate consistency to within an error* δ, which simply means that $u(x_i) + \delta \geq u(x)$ must hold instead for some $\delta > 0$.

Let $c_{ij} = p_i(x_j)$ be the cost of bundle j at observation i, when x_i was chosen. In order to obtain utilities that rationalize a given set of observations, Afriat [1] introduced the following system of inequalities with variables v_i, λ_i for $i \in N$:

$$v_i - \lambda_i c_{ii} \geq v_j - \lambda_i c_{ij} \quad (i, j \in N) \tag{1}$$

with the added constraint that $\lambda_i > 0$ for all $i \in N$. To motivate how these inequalities arise, suppose that the agent's utility is concave. Then a necessary condition for $x_i \in D(p_i)$ is that there exist a Lagrange multiplier $\lambda_i \geq 0$ such that $x_i \in \arg\max_{x \in X} \{u(x) - \lambda_i [p_i(x) - b_i]\}$; it should now be clear that v_i is meant to correspond to $u(x_i)$. Furthermore, if utility is strictly monotone, we will have $\lambda_i > 0$ because the budget constraint will bind. Thus the inequalities describe observable constraints on the utilities of the demanded bundles together with the associated Lagrange multipliers (assuming utility is concave); the multipliers have an intuitive interpretation as the marginal utility of wealth at each observation [21].

It turns out that a positive solution to these inequalities is a necessary and sufficient condition not just for rationalization by a concave utility function, but by any strictly monotone utility function. The following was proved by Afriat [1].

Theorem 1 *The observations can be rationalized by a strictly monotone utility function if and only if the system of inequalities (1) has a positive solution. If (v, λ) is such a solution then the utility function*

$$v(x) = \min_{i \in N} \{v_i + \lambda_i [p_i(x) - p_i(x_i)]\} \tag{2}$$

rationalizes the observations.

Note that the condition that the solution be positive only constrains the variables λ_i, because we can always add a constant to the v_i to make them all positive and maintain feasibility. It is easy to see that if the prices p are strictly monotone and linear, then (2) defines a strictly monotone, concave utility function [7]. Thus one interpretation of Afriat's theorem is that violations of monotonicity and concavity cannot be detected with a finite amount of demand data under linear prices [19].

The problem at hand from here on is that of finding a utility function consistent with observations when price information is *nonlinear* and possibly incomplete. Now, Afriat's Theorem holds even if prices are nonlinear; the assumption of linear prices is needed only to establish that (2) is concave. Also, inequalities (1) can be formulated and solved even with partial price information; we simply discard those where the prices are not available. Thus if all we care for is to check whether the observations are consistent with utility maximization, nonlinearity poses no problem. However, recovering a utility function is a different matter. With full price information, we could still construct (2) to forecast the utilities of other bundles even if prices are nonlinear; we will henceforth refer to this as the *full-information method*. This method in fact has the potential to generalize very poorly. To see why, consider Figure 1 (following page).

In the figure both the utility u and prices p_i are nonlinear; recall that λ_i is the Lagrange multiplier at observation i. The chosen bundle at prices p_i is x_i because it maximizes $u - \lambda_i p_i$. Now assume that for x the minimum in (2) is attained at $i \in N$. Then the forecasted utility for bundle x will be $v(x)$, which here could be arbitrarily far off from the real utility $u(x)$ due to the nonlinearity of p_i. In general, the flaw with the approach is that with nonlinear prices, the structure of prices may bear no connection with the structure of the utility function; for instance, the price structure may be much more complex than the utility structure, so it is not sensible to formulate utility in terms of the observed prices as in (2). A better approach would be to first identify linear prices $p'_i \leq p_i$ as in the figure and use those in (2) instead of p_i to forecast utilities.[2]

[2] Incidentally, Figure 1 also gives the simple intuition behind the fact that

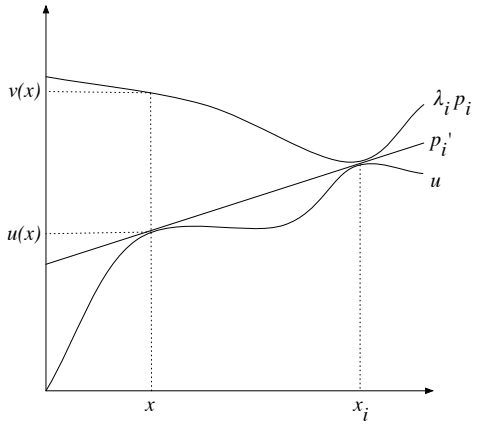


Figure 1. Utility recovery example.

If price information is incomplete, then it may not even be possible to evaluate (2) at a given bundle x. A naive approach would be to first fit a price function to the available price data using an appropriate machine learning method, and then evaluate (2). Again, this is not a sensible approach for the same reason as just mentioned: prices can be more complicated and bear no relation to the utility function, so fitting the price data may amount to solving a much more difficult problem than that of recovering the utilities.

In the remainder we will assume that the available information consists *only* of the demanded bundles x_i for $i \in N$ as well as the matrix consisting of costs $c_{ij} = p_i(x_j)$ for all $i, j \in N$. We will not have available the prices of any non-demanded bundles. In fact, it will be clear that our methods can be applied with whatever partial price information is available (i.e., prices of some demanded bundles may be missing and prices of some non-demanded bundles may be present). We assume that we have exactly the cost matrix c only to simplify the exposition.

3 KERNEL METHODS

The idea behind kernel methods in machine learning is the "kernel trick": training examples are recoded in terms of a set of features and then linear classification or regression algorithms are applied to the examples in this new encoding. The choice of an encoding amounts to the choice of a function class from which to fit a classifier or regressor [17].

Here our examples are bundles $x \in X$. We map the bundles into a *feature space* $Y = \mathbf{R}^M$ via a mapping $\phi : X \to Y$. Typically we will have $M \gg m$; in fact M may be infinite. Consequently, to develop a workable kernel method with respect to a mapping ϕ, one must find a way to fit a utility function without ever explicitly working with bundle representations in Y. The trick is to instead formulate the problem purely in terms of scalar products $\langle \phi(x), \phi(x') \rangle$. What make this practical is that, for many useful mappings, the scalar products can be evaluated in time that does not depend on M.

A *kernel function* κ computes the scalar product of the images of two bundles in feature space: $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$. A feature space can be equivalently specified either through a mapping ϕ or a kernel function κ. In our experiments we will focus on the *polynomial* kernel, defined by $\kappa(x, x') = (\langle x, x' \rangle + 1)^d$, where d is a parameter. The corresponding mapping ϕ maps bundles into a feature space that has a dimension for every monomial $x_1^{i_1} x_2^{i_2} \cdots x_m^{i_m}$ of degree at most d—see [17] or any other standard textbook on kernel methods for a treatment of the polynomial kernel. With $d = 1$, we essentially recover the *linear* kernel corresponding to the mapping $\phi(x) = x$; we would use this kernel to try to fit linear or concave functions in the original bundle space.

We propose two different kernel methods to recover a utility function from demand data. The inequalities (1) form the basis of both. The first method attempts to fit a linear utility function in feature space to the demand data. The second attempts to fit a concave function in feature space. Note that in the original bundle space X this yields functions that can be highly nonlinear or non-concave.

3.1 Linear utility

The first method directly ascribes some structure to the utility function and then leverages that structure to generalize across the consumption set. This is of course a standard approach in econometrics, where the usual assumption is that utilities are linear in the goods (e.g., in discrete choice analysis [18]). The utility function will be represented as a vector $v \in Y$ in feature space; the utility of a bundle x therefore becomes $\langle v, \phi(x) \rangle$.

The problem of fitting the utility function to the data is formulated as the following quadratic program (PL).

$$\begin{aligned} \min_{v,\lambda,\epsilon} \quad & \frac{\mu}{2} \|\epsilon\|^2 + \frac{1}{2} \|v\|^2 \\ \text{s.t.} \quad & \langle v, \phi(x_i) \rangle - \lambda_i c_{ii} + \epsilon_{ij} \geq \langle v, \phi(x_j) \rangle - \lambda_i c_{ij} \quad (i, j \in N) \\ & \lambda_i \geq 1 \quad (i \in N) \end{aligned}$$

We see that the constraints are completely analogous to inequalities (1) together with the constraint that λ be positive.[3] We have introduced slack variables ϵ to ensure feasibility. Accordingly, we introduce a penalty term $\frac{\mu}{2} \|\epsilon\|^2$ on slack in the objective; if this term is zero, the resulting utility function will be exactly consistent with the data. The role of the $\frac{1}{2} \|v\|^2$ term will become clear shortly.

In the program ϵ is a vector of dimension n^2 whereas v is a vector of dimension M in feature space. Thus we cannot solve the program directly and instead work with the dual (DL), which is as follows.

$$\begin{aligned} \max_{\alpha \geq 0, s \geq 0} \quad & \sum_{i \in N} s_i - \frac{\nu}{2} \|\alpha\|^2 \\ & - \frac{1}{2} \left\| \sum_{i,j \in N} \alpha_{ij} [\phi(x_i) - \phi(x_j)] \right\|^2 \\ & s_i = \sum_{j \in N} \alpha_{ij} (c_{ii} - c_{ij}) \quad (i \in N) \end{aligned} \tag{3}$$

We have $\nu = 1/\mu$. Here α is a vector of dimension n^2 and s is a vector of dimension n. The squared norm in (3) evaluates to $\alpha' K \alpha$ where K is an $n^2 \times n^2$ matrix with rows and columns indexed by pairs (i, j) for $i, j \in N$. The entry corresponding to row (i, j) and column (k, l) is

$$\kappa(x_i, x_k) - \kappa(x_j, x_k) - \kappa(x_i, x_l) + \kappa(x_j, x_l).$$

only concave functions can be recovered when prices are linear. With linear prices, bundles in between x and x_i can never be demanded, because the prices can never be tangent at those bundles. Thus the most that could be recovered is the upper envelope of u and p_i'. With nonlinear prices, it may be possible to recover the utility function between x and x_i.

[3] We cannot simply write $\lambda > 0$ because the feasible set of a quadratic program must be closed to ensure a solution exists. But note that if inequalities (1) have a positive solution, then we can obtain a solution with $\lambda \geq \mathbf{1}$ by rescaling all the variables v and λ by a positive constant.

Thus the dual can be solved in time independent of M. Nonetheless, the fact that K has on the order of n^4 nonzero entries means that large-scale optimization techniques such as delayed column generation would be needed in the presence of large numbers of observations [11].

The properties of the method's solution are captured in the following result. Its proof consists of a straightforward appeal to strong duality and the Karush-Kuhn-Tucker (KKT) conditions.

Theorem 2 *For sufficiently large μ (small ν), there is a linear utility function v over Y consistent with the data to within an error of δ if and only if (DL) has an optimal solution (α, s) with $\max_{i,j \in N} \alpha_{ij} < \mu\delta$. In this case v can be evaluated as*

$$v(x) = \sum_{i,j \in N} \alpha_{ij}[\kappa(x_i, x) - \kappa(x_j, x)]. \tag{4}$$

(In the above μ depends on δ.) The theorem only guarantees approximate consistency with the data rather than rationalization as in Afriat's Theorem, because we are dealing with partial price information. However, even with full price information available, our empirical evaluation will demonstrate that this method can generalize better than the full-information method. The theorem also does not guarantee monotonicity across the consumption set, but it is easy to show that if x_i, x_j are two bundles in the data such that $x_i > x_j$, then $v(x_i) + \delta > v(x_j)$ assuming the prices were strictly monotone. Thus we do achieve approximate monotonicity over the data.

The first reason for introducing $\frac{1}{2}\|v\|^2$ into the primal objective is practicality. Without it the term (3) in the dual would appear as a set of M hard constraints rather than a penalty term, which must be avoided. The second reason is more principled. It is well-known that a regularization term on the fitted function can be interpreted as a prior over the function [17]. Suppose that our prior states that the utility function is drawn according to a zero-mean Gaussian in feature space, meaning that $\text{Prob}(v) \propto e^{-\|v\|^2}$, while the error (slack) terms are also drawn independently according to a zero mean Gaussian, so that $\text{Prob}(\epsilon) \propto e^{-\|\epsilon\|^2}$. Then $-\log \text{Prob}(v, \epsilon)$ is the objective in (PL) for some μ, and the program computes the maximum a posteriori estimate given the prior and the data. If a mean of zero utilities seems odd, recall that only relative utilities matter. Zero utilities simply mean that the agent is indifferent across all bundles. In the absence of any other information, this seems like a natural prior.

3.2 Concave utility

The second method fits a concave utility function to the data in feature space Y, in analogy to the full-information method (2), which constructs a concave function in the original bundle space X. The problem of fitting the concave function to the data is formulated as the following quadratic program (PC).

$$\begin{aligned}
\min_{v,\lambda,p,\epsilon,\bar{\epsilon}} \quad & \frac{\mu}{2}\|\epsilon\|^2 + \frac{\mu}{2}\|\bar{\epsilon}\|^2 + \frac{1}{2}\sum_{i \in N}\|p_i\|^2 & \\
\text{s.t.} \quad & v_i - \langle p_i, \phi(x_i)\rangle + \epsilon_{ij} \geq v_j - \langle p_i, \phi(x_j)\rangle & (i,j \in N) \\
& \langle p_i, \phi(x_i)\rangle - \lambda_i c_{ii} + \bar{\epsilon}_{ij} \geq \langle p_i, \phi(x_j)\rangle - \lambda_i c_{ij} & (i,j \in N) \\
& \lambda_i \geq 1 & (i \in N)
\end{aligned}$$

Note that if the first and second constraints for $i, j \in N$ are added together, we recover the constraints (1). The motivation for this formulation follows the intuition in Figure 1: rather than using the original prices to forecast the utility of bundles, we first lower bound the prices with vectors drawn from the feature space Y.

In the program v and λ are vectors of dimension n, ϵ is a vector of slack variables of dimension n^2, and each p_i is a vector of dimension M in feature space. As before, the slack variables ensure feasibility. If the penalty terms on $\|\epsilon\|^2$ and $\|\bar{\epsilon}\|^2$ in the objective are zero, the program will have identified a function that is exactly consistent with the data.

Since the primal is explicitly formulated in terms of vectors in Y, we must again work with the dual (DC), which is as follows.

$$\begin{aligned}
\max_{\alpha \geq 0, \bar{\alpha} \geq 0, s \geq 0} \quad & \sum_{i \in N} s_i - \frac{\nu}{2}\|\alpha\|^2 - \frac{\nu}{2}\|\bar{\alpha}\|^2 & \\
& -\frac{1}{2}\sum_{i \in N}\left\|\sum_{j \in N}(\bar{\alpha}_{ij} - \alpha_{ij})[\phi(x_i) - \phi(x_j)]\right\|^2 & (5) \\
\text{s.t.} \quad & s_i = \sum_{j \in N} \bar{\alpha}_{ij}(c_{ii} - c_{ij}) & (i \in N) \\
& \sum_{j \in N} \alpha_{ij} = \sum_{j \in N} \alpha_{ji} & (i \in N)
\end{aligned}$$

We have $\nu = 1/\mu$. Here α and $\bar{\alpha}$ are vectors of dimension n^2, while s is of dimension n. The squared norm in (5) for each $i \in N$ can be written as $(\bar{\alpha}_i - \alpha_i)' K_i (\bar{\alpha}_i - \alpha_i)$ where K_i is an $n \times n$ matrix with the entry corresponding to $j, k \in N$ being

$$\kappa(x_i, x_i) - \kappa(x_i, x_j) - \kappa(x_i, x_k) + \kappa(x_j, x_k).$$

Thus the Hessian in this program has on the order of n^3 nonzero entries, which compares favorably to the linear utility method.

The properties of the concave method's solution are captured in the following result. Again, its proof consists of a straightforward appeal to strong duality and the KKT conditions.

Theorem 3 *For sufficiently large μ (small ν), there is a concave utility function v over Y consistent with the data to within an error of δ if and only if (DC) has an optimal solution $(\alpha, \bar{\alpha}, s)$ with $\max_{i,j \in N}(\alpha_{ij} + \bar{\alpha}_{ij}) < \mu\delta$. In this case v can be evaluated as*

$$v(x) = \min_{i \in N}\{v_i + p_i(x) - p_i(x_i)\} \tag{6}$$

where, for each $i \in N$,

$$p_i(x) = \sum_{j \in N}(\bar{\alpha}_{ij} - \alpha_{ij})[\kappa(x_i, x) - \kappa(x_j, x)]. \tag{7}$$

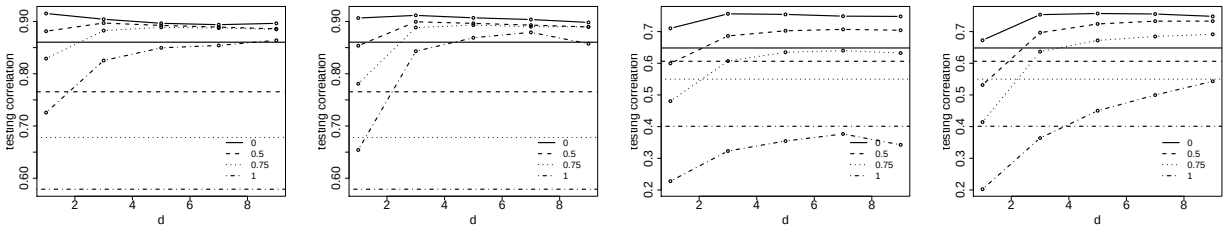
(Again, μ depends on δ.) According to the theorem, the p_i can be evaluated given the α and $\bar{\alpha}$ from the dual solution. However, we also need to back out the v_i's from the primal. The KKT conditions imply that if $\alpha_{ij} > 0$ then the corresponding constraint binds:

$$v_i - \langle p_i, \phi(x_i)\rangle + \epsilon_{ij} = v_j - \langle p_i, \phi(x_j)\rangle. \tag{8}$$

The KKT conditions also imply that $\epsilon_{ij} = \nu\alpha_{ij}$. Therefore, with the dual solution at hand, we can formulate the system of equalities corresponding to (8) for each $\alpha_{ij} > 0$ and solve it to obtain the v_i's from the primal solution.

As with the linear utility method, this method guarantees approximate consistency with the data, but not rationalization. Our empirical evaluation will show that this is not a drawback. It is easy to show that the derived utility function is approximately monotone over the observed data, assuming the prices were originally strictly monotone.

Again, there are practical and principled reasons to introduce the penalty terms $\|p_i\|^2$ in the primal objective. First, they ensure that



(a) regions, linear (b) regions, concave (c) scheduling, linear (d) scheduling, concave

Figure 2. Generalization ability of the linear and concave utility methods under the regions and scheduling distributions. The horizontal lines depict the testing correlation of the full-information method. All data points are averaged over 50 instances.

the dual is formulated in terms of scalar products between bundles, so that we obtain a kernel method. Second, as each p_i can be interpreted as the gradient (or more precisely, a supergradient) to the utility function at x_i, the penalty terms can be construed as prior information on the gradient of the concave function at each bundle in the data. Specifically, we have a zero-mean Gaussian prior for each gradient; this amounts to the belief that utilities do not change, meaning that the agent is indifferent among all bundles. Thus our prior here is analogous to the prior used for the linear utility method.[4]

4 EMPIRICAL EVALUATION

In this section we report on experiments run to evaluate the performance of our two kernel methods in terms of their ability to generalize, the sparsity of their solutions, and their scalability. We used the CATS suite of distributions to generate utility functions [13]. CATS represents utility functions in the XOR language [14]. We denote an XOR instance by a pair (u', Z) where $Z \subseteq X$ is a finite subset of bundles and $u' : Z \to \mathbf{R}$ is a utility function defined over this restricted set. The utility function u corresponding to the XOR instance is given by $u(x) = \max_{\{z \leq x : z \in Z\}} u'(z)$. The original purpose of CATS was to generate valuation functions to test winner-determination algorithms for combinatorial auctions, so the magnitudes of the utilities are supposed to be meaningful; for our purposes, we treat them simply as ordinal utilities. Also, the goods in CATS are indivisible, while in our model so far goods have been divisible; our methods are perfectly applicable to observations of demanded bundles with indivisible goods only so this is not an issue.

The obtain demand observations given an XOR instance, prices were generated as follows. We first fix a $\gamma \in [0, 1]$ that controls the degree to which the prices should be nonlinear. We then draw a linear price vector uniformly at random from the price simplex $\{p \in \mathbf{R}^m_+ : p'\mathbf{1} = 1\}$. Finally, we construct the XOR instance (Z, u'') where $u''(z) = \beta_z p(z)$ for β_z draw uniformly at random from $[1 - \gamma, 1 + \gamma]$. Our final prices are the function represented by this XOR instance. Note that with $\gamma = 0$ we obtain linear prices, while with $\gamma = 1$ the linear prices are highly perturbed; note also that under XOR semantics utilities and prices are always monotone.

[4] Many other interesting priors could be incorporated. For instance, we could introduce a term $w_{ij} \|p_i - p_j\|^2$ in the objective to specify correlation between the two gradients p_i and p_j. If we believe the utility function to be continuously differentiable, then it is natural to take w_{ij} inversely proportional to $\|\phi(x_i) - \phi(x_j)\|^2$. The latter can be evaluated using the kernel function κ. We leave a deeper investigation of these approaches to future work.

We considered four different distributions provided by the CATS suite: arbitrary, paths, regions, and scheduling. To create a problem instance, we first generate a utility function with one of these distributions; throughout all our experiments the XOR instances were of size $|Z| = 200$ using 20 goods. We then repeatedly generate prices p_i as described above (using a fixed γ). For each draw of p_i we pick a bundle $z_i \in Z$ uniformly at random, and set the budget for this observation to $b_i = p_i(z_i)$. Finally, we record the demanded bundle under prices p_i and budget b_i. (It is not necessarily the case that this bundle is z_i; however, with an XOR instance, it is necessarily the case that a demanded bundle is drawn from Z.) Once we have collected $n = 50$ demanded bundles, we record the cost matrix $c_{ij} = p_i(x_j)$ for each i, j in our set of observations N and disregard any other price information from there on. The observations do not typically consist of 50 distinct demanded bundles; in our experiments the number of unique bundles observed was on average 28 with a standard deviation of 5.6.

In our experiments we restrict our attention to the polynomial kernel previously introduced, varying the complexity parameter d. The kernel methods were implemented in Python 2.5, and the quadratic programs were solved using the `cvxopt` module.[5] We used $\nu = 1.0$ throughout so that each method is evaluated on the same footing. The experiments were run on a 2.13 GHz, Intel Core 2, 2GB machine running Linux.

Generalization. To assess the generalization ability of our kernel methods we consider the Spearman rank correlation between the fitted utility function v and actual utility function u over the bundles in Z. Rank correlation is the correct measure of agreement here because utility is ordinal. Specifically, let $Z' \subseteq Z$ be the bundles that have been observed (i.e., demanded at some observation), and let $Z'' = Z \backslash Z'$ be the unobserved bundles—the prices of the latter were not input into our methods. We define the training correlation to be the rank correlation between the vectors $(u(z) : z \in Z')$ and $(v(z) : z \in Z')$. The testing correlation is analogously defined with Z'' replacing Z'.

Figure 2 exhibits the testing correlation of the linear and concave utility methods for the regions and scheduling distributions, varying d and γ, as well as the full-information method as a benchmark. We see that the testing correlation usually improves with increased d, confirming that flexibility in the choice of kernel function can bring significant advantages. With $\gamma = 0$, increasing d does not make much difference, but this is expected because with linear prices $d = 1$ should suffice. Of course, it is possible for the methods to begin to overfit, as we observe in Figures 2(b) and 2(c) for $\gamma = 1$.

[5] `http://abel.ee.ucla.edu/cvxopt/`

In general, the linear utility method generalizes better at lower d but the best generalization is achieved by the concave method at higher d. Significantly, both methods universally outperform the full-information method when $d \geq 5$, with the exception of the concave method at $\gamma = 1$ in Figure 2(c). The full-information method's performance degrades as γ increases. This bears out our original intuition that incorporating the price structure into the estimated utility can be detrimental, even though it leads to a utility function that technically rationalizes the data. Table 1 provides an alternate view of the training and testing correlation of both methods.

		training		testing		sparsity	
distribution	d	lin.	*con.*	lin.	*con.*	lin.	*con.*
arbitrary	1	.92	*.83*	.81	*.75*	.17	*.55*
	5	.93	*.94*	.91	*.91*	.18	*.60*
	9	.92	*.92*	.90	*.91*	.13	*.18*
paths	1	.78	*.64*	.11	*.11*	.27	*.65*
	5	.81	*.80*	.24	*.33*	.26	*.62*
	9	.81	*.81*	.26	*.38*	.26	*.72*
regions	1	.93	*.85*	.83	*.78*	.18	*.60*
	5	.93	*.94*	.89	*.89*	.19	*.58*
	9	.93	*.93*	.88	*.89*	.15	*.20*
scheduling	1	.84	*.75*	.48	*.41*	.22	*.61*
	5	.84	*.86*	.63	*.67*	.22	*.63*
	9	.83	*.86*	.63	*.69*	.19	*.28*

Table 1. Training correlation, testing correlation, and utility function sparsity for the linear and concave utility methods under $\gamma = 0.75$. All data points are averaged over 50 instances.

Sparsity. The sparsity of a utility function derived by the linear utility method is defined as the number of nonzero α_{ij} coefficients in (4) over n^2. Similarly, the sparsity of a utility function derived by the concave utility method is defined as the number of nonzero $(\bar{\alpha}_{ij} - \alpha_{ij})$ coefficients in (7), summed over all i, over n^2. Sparse solutions are desirable because they are faster to evaluate, and capture the utility function succinctly. Table 1 provides some sample sparsities when $\gamma = 0.75$ (sparsities were comparable for other γ). We see that the linear utility method consistently generates the sparsest utility functions across all d and distributions. The concave utility method usually generates sparser solutions as d is increased. The reasons for this remain unclear, as we had initially expected smaller d (which reflect simpler structure in the fitted gradients) to lead to sparser solutions. Nevertheless, this is a welcome finding because the concave method generalizes better at higher d in the range considered.

Runtime. Given that the generalization abilities of both methods are comparable, one could conclude that the linear utility method is slightly preferred since it generates sparser solutions and is conceptually simpler. However, as mentioned earlier, the size of the quadratic program is on the order of n^4 for the linear method whereas it is on the order of n^3 for the concave method. This translates into a significant difference in runtime performance, as Table 2 shows. We find an order of magnitude difference between the runtimes of the two methods. Essentially, the linear method as implemented cannot scale to even a moderate number of observations such as 100, whereas the concave method can readily handle such problem sizes. It should be possible to draw on techniques from large-scale optimization such as delayed column generation to improve the runtime of the linear method [11]. We leave this to future work, although the concave method already offers a satisfactory alternative.

observations	10	20	30	40	50
linear	0.4	2.7	14	64	221
concave	0.5	2.2	5.6	12	22

Table 2. Runtime performance of the linear and concave utility functions, in *seconds*, scaling the number of observations; regions distribution, $\gamma = 0.75$, $d = 5$. All data points are averaged over 50 instances.

REFERENCES

[1] Sidney N. Afriat, 'The construction of utility functions from expenditure data', *International Economic Review*, **8**(1), 67–77, (February 1967).

[2] Eyal Beigman and Rakesh Vohra, 'Learning from revealed preference', in *Proc. of the 7th ACM Confererence on Electronic Commerce (EC)*, pp. 36–42, (2006).

[3] Olivier Chapelle and Zaïd Harchaoui, 'A machine learning approach to conjoint analysis', in *Advances in Neural Information Processing Systems*, 17. MIT Press, (2005).

[4] Koby Crammer and Yoram Singer, 'Pranking with ranking', in *Advances in Neural Information Processing Systems*, pp. 641–647. MIT Press, (2002).

[5] Carmel Domshlak and Thorsten Joachims, 'Unstructuring user preferences: Efficient non-parametric utility revelation', in *Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 169–177, (2006).

[6] Theodoros Evgeniou, Constantinos Boussios, and Giorgos Zacharia, 'Generalized robust conjoint estimation', *Marketing Science*, **24**(3), 415–429, (2005).

[7] Ana Fostel, Herbert E. Scarf, and Michael J. Todd, 'Two new proofs of Afriat's theorem', *Economic Theory*, **24**, 211–219, (2004).

[8] John Gross, 'Testing data for consistency with revealed preference', *Review of Economics and Statistics*, **77**(4), 701–710, (1995).

[9] Ralf Herbrich, Thore Graepel, and Klaus Obermayer, 'Large margin rank boundaries for ordinal regression', in *Advances in Large Margin Classifiers*, pp. 115–132. MIT Press, (2000).

[10] Hendrik S. Houthakker, 'Revealed preference and the utility function', *Economica*, **17**(66), 159–174, (May 1950).

[11] Thorsten Joachims, 'Training linear SVMs in linear time', in *Proc. of the 12th ACM International Conference on Knowldge Discovery and Data Mining (KDD)*, pp. 217–226, (2006).

[12] Michael J. Kearns and Umesh V. Vazirani, *An Introduction to Computational Learning Theory*, MIT Press, 1994.

[13] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham, 'Towards a universal test suite for combinatorial auction algorithms', in *Proc. of the second ACM Conference on Electronic Commerce (EC)*, pp. 66–76, (2000).

[14] Noam Nisan, 'Bidding and allocation in combinatorial auctions', in *Proc. second ACM Conference on Electronic Commerce (EC)*, pp. 1–12, (2000).

[15] Marcel K. Richter, 'Revealed preference theory', *Econometrica*, **34**(3), 635–645, (July 1966).

[16] Paul A. Samuelson, 'Consumption theory in terms of revealed preference', *Economica*, **15**(60), 243–253, (November 1948).

[17] Bernhard Schölkopf and Alex J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2001.

[18] Kenneth E. Train, *Discrete Choice Methods with Simulation*, Cambridge University Press, 2009.

[19] Hal R. Varian, 'The nonparametric approach to demand analysis', *Econometrica*, **50**(4), 945–973, (July 1982).

[20] Hal R. Varian, 'Goodness-of-fit in optimizing models', *Journal of Econometrics*, **46**, 125–140, (1990).

[21] Hal R. Varian, 'Revealed preference', in *Samuelsonian Economics and the Twenty-First Century*, Oxford University Press, (2006).

[22] Robert B. Wilson, *Nonlinear Pricing*, Oxford University Press, 1993.