Mach Learn
https://doi.org/10.1007/s10994-018-5705-5

A scalable preference model for autonomous decision-making

Markus Peters[1] · **Maytal Saar-Tsechansky**[2] · **Wolfgang Ketter**[1,3] · **Sinead A. Williamson**[2] · **Perry Groot**[4] · **Tom Heskes**[4]

Received: 31 December 2015 / Accepted: 4 April 2018

Abstract Emerging domains such as smart electric grids require decisions to be made autonomously, based on the observed behaviors of large numbers of connected consumers. Existing approaches either lack the flexibility to capture nuanced, individualized preference profiles, or scale poorly with the size of the dataset. We propose a preference model that combines flexible Bayesian nonparametric priors—providing state-of-the-art predictive power—with well-justified structural assumptions that allow a scalable implementation. The Gaussian process scalable preference model via Kronecker factorization (*GaSPK*) model provides accurate choice predictions and principled uncertainty estimates as input to decision-making tasks. In consumer choice settings where alternatives are described by few key attributes, inference in our model is highly efficient and scalable to tens of thousands of choices.

Editor: Johannes Fürnkranz.

✉ Markus Peters
peters@rsm.nl

Maytal Saar-Tsechansky
maytal@mail.utexas.edu

Wolfgang Ketter
wketter@rsm.nl

Sinead A. Williamson
sinead.williamson@mccombs.utexas.edu

Perry Groot
perry.groot@science.ru.nl

1 Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands

2 McCombs School of Business, University of Texas at Austin, Austin, USA

3 Faculty of Management, Economics and Social Sciences, University of Cologne, Cologne, Germany

4 ICIS, Radboud University Nijmegen, Nijmegen, The Netherlands

Published online: 01 May 2018

Keywords Autonomous agents · Autonomous decision-making · Bayesian inference · Discrete choice · Gaussian processes · Laplace inference · Preferences

1 Introduction

Data-driven modeling has become integral to informing a growing array of decisions, yet *autonomous* decision-making remains elusive under all but the most highly structured circumstances. Two prominent application domains, dynamic flight pricing and automated credit approvals, exemplify how automated business rule engines make most operative decisions quickly, cheaply, and reliably (Baker 2013). However, in less structured settings involving individual preferences—from planning the next vacation to trading in complex multi-echelon markets—autonomous decision-making through software agents remains an active area of research, e.g., Adomavicius et al. (2009).

A key challenge in autonomous decision-making in unstructured settings is the identification of what choices a given user *deems* best. Individuals may be unaware of the drivers underlying their own preferences (Lichtenstein and Slovic 2006), making data-driven preference models instrumental because they can elicit preferences by generalizing from *observed* choices (Bichler et al. 2010). While we will typically see global patterns in preferences (e.g. cheaper options preferred over more expensive options), we do not expect a single model to capture all behavior. Two individuals may make different choices when faced with the same options, and even a single individual may not always make consistent decisions.

To see the benefit of preference modeling consider, for example, **smart grids** (Kassakian and Schmalensee 2011), where data-driven learning is anticipated to play a key role in facilitating efficient electricity distribution and use. Particular challenges in this context are electric vehicles that are charged in varying locations, and the incorporation of intermittent and variable renewable electricity sources, such as solar and wind (Valogianni et al. 2014). Data-driven modeling of electricity consumption preferences under different incentives and contexts is essential for effectively incentivizing consumers to choose sustainable behaviors (Watson et al. 2010; Peters et al. 2013). A number of factors may influence an individual's choice to use her electric vehicle—for example, time of day, cost of electricity, or weather conditions. A preference model can learn that a user prefers not to use her electric vehicle in the morning if electricity is at a higher price point, over not using the vehicle if the cost of electricity is reduced. Such a model could be used to incentivize electric vehicle owners to make the car battery's energy available to nearby consumers when renewable energy is scarce. Electricity cost and emission reduction, informed by such data-driven preference learning and autonomous decision-making, can be significant (Kahlen et al. 2014).

Prior work on preference learning has made significant advances in generating accurate predictions from noisy observations such as electricity meter readings that are inconsistent and heterogeneous (Kohavi et al. 2004; Evgeniou et al. 2005). Recently, non-parametric Bayesian models have proved particularly advantageous. A Bayesian framework, such as that used by Guo and Sanner (2010), explicitly models uncertainty and accommodates inconsistencies in human choices rather than imposing stringent rationality assumptions. By allowing inconsistencies in observed choices to be translated to uncertainty, Bayesian models can distinguish between cases where estimates are certain enough to justify an autonomous action, and cases when the model might benefit from actively acquiring additional evidence or transfer con-

trol to a human decision-maker (Saar-Tsechansky and Provost 2004; Bichler et al. 2010). Non-parametric Bayesian models are a particularly flexible class of Bayesian models that minimize assumptions made about the structure underlying the data, instead automatically adapting to the complexity of real-world observations (Guo and Sanner 2010; Bonilla et al. 2010; Houlsby et al. 2012).

Existing non-parametric Bayesian methods that achieve state-of-the-art predictive accuracies do not scale well to a large number of users. Their prohibitive computational costs cannot be addressed with additional processing power or offline processing, making such methods impractical for modeling a large number of users. Conversely, models that are more scalable, such as the restricted value of information algorithm (Guo and Sanner 2010), tend to be parametric models that offer inferior predictive performance compared with more complex models (Bonilla et al. 2010). If non-parametric Bayesian methods are to become widely adopted in practice, progress must be made to ensure that they both scale well and achieve high-quality predictions. In particular, important domains such as energy markets and healthcare require methods that are **computationally efficient**, and that scale gracefully with respect to the number of users and observations. Contemporary electric distribution systems, for example, produce large amounts of data from up to ten million consumer meters, each transmitting data every few minutes (Widergren et al. 2004). Such large amounts of data must be processed quickly and at high granularity (i.e., unaggregated), as automated responses often rely on fine-grained, local information. It is therefore important for preference models to provide consistently fast training times, as well as to incorporate and act on new data in a timely manner.

In this paper we develop and evaluate a novel, non-parametric, Bayesian approach that offers an advantageous augmentation to the existing preference modeling toolset. Our approach, **Ga**ussian process **S**calable **P**reference model via **K**ronecker factorization (*GaSPK*), leverages common features of consumer choice settings, particularly the small set of relevant product attributes, to yield state-of-the-art scalability. *GaSPK* formulates a personalized preference model based on a shared set of trade-offs, designed in a way that facilitates the use of Kronecker covariance matrices. While covariance matrices with Kronecker structure have been used in a preference learning context (Bonilla et al. 2010), no prior work on preference learning has employed their favorable factorization and decomposition properties to produce scalable algorithms. As we will see in Sects. 3.2 and 5, this leads to improved theoretical and empirical scalability over related preference models.

We empirically evaluate *GaSPK*'s performance relative to that of key benchmarks on three real-world consumer choice datasets. For this study we collected an electricity tariff choice dataset on a commercial crowdsourcing platform for a U.S. retail electricity market. To confirm our findings we evaluated the methods on two benchmark choice datasets on political elections and car purchases. Our results establish that *GaSPK* is likely to often be the method of choice for modeling preferences of a large number of users. *GaSPK* produces state-of-the-art scalability, while often yielding favorable predictive accuracy as compared to the accuracy achieved by existing approaches.

GaSPK introduces a new benchmark to the preference modeling toolset that is particularly suitable for modeling a large number of users' preferences when alternatives can be described by a small number of relevant attributes. Its principled handling of uncertainty is also instrumental for autonomous decision-making.

2 Gaussian process scalable preference model via Kronecker factorization (*GaSPK*)

We begin with outlining the *GaSPK* learning approach. As discussed above, *GaSPK* aims to augment the non-parametric Bayesian preference modeling toolset to allow for scalability and conceptual simplicity in consumer choice settings. Our discussion begins with a description of the fundamentals of *GaSPK*. We outline our contributions to facilitate scalability and conceptual simplicity in Sect. 3.

Let $U = \{u_1, \ldots, u_{n_U}\}$ denote a set of **users** and $X = \{x_1, \ldots, x_{n_X} \mid x_i \in \mathbb{R}^{d_X}\}$ a set of **instances**, the objects or actions between which users choose. Each instance is described by d_X real-valued attributes. Data is presented as a set of observed, binary **choices**, denoted as:

$$C = \left\{(u, x^1, x^2, y) \mid u \in U, x^i \in X, y \in \{+1, -1\}\right\}$$

Here, $y \in \{-1, +1\}$ denotes the user's choice: $(u, x^1, x^2, +1)$ reflects that user u prefers the first alternative (instance x^1) to the second alternative (instance x^2), whereas $(u, x^1, x^2, -1)$ reflects the opposite preference. Table 1 summarizes the mathematical notation used throughout the paper. The goal of preference learning is to learn an order relation $\succeq_u$ over instances

Table 1 Summary of mathematical notation (symbols are in alphabet order)

Symbol	Definition
$\circ$	Hadamard (element-wise) matrix product
$\otimes$	Kronecker matrix product
$C = \left\{(u, x^1, x^2, y)\right\}$	Choice situations: when presented with alternatives x^1 and x^2, user u chose $y = +1$ (first alternative), or $y = -1$ (second alternative)
γ_u^c, Γ_u, Γ	$\Gamma_u = (\gamma_u^1, \ldots, \gamma_u^C)$ is a probability vector indicating the extent of user u's possession of each of the n_C characteristics; $\Gamma \in \mathbb{R}^{n_U \times n_c}$ collects all Γ_u
d_T, d_X	Dimensionality of elements in T, X
f_u, f^c	Functions $f : \mathbb{R}^{d_T} \to \mathbb{R}$ describing users' latent evaluation of trade-offs and characteristic evaluation of trade-offs, respectively
$\theta = \{l_d\}$	Lengthscale hyperparameters
I	Identity matrix
K	Covariance matrix, $K \in \mathbb{R}^{(n_c n_T)\times(n_c n_T)}$
$L : LL^T = W$	Lower Cholesky factor of W
n_c	Fixed number of characteristics
n_e	Number of Eigenvalues used in low-rank approximations
n_T, n_U, n_X	Number of elements in T, U, and X
N, Φ	Probability density function (PDF), and cumulative distribution function (CDF) of the standard normal distribution
$p(C\|\{f_u\})$, $\nabla p(C\|\{f_u\})$	Likelihood and its Jacobian $\frac{\partial p(C\|f)}{\partial f_i}$
t, $t^{(d)}$, T	Trade-off t, its d-th element, and set of all trade-offs, respectively
U	Set of all users
$W = -\nabla\nabla \log p(C\|\{f_u\})$	Negative Hessian of the log likelihood
X	Set of all instances
$y \in \{-1, +1\}$	Single choice: $y = +1$ (first alternative), or $y = -1$ (second alternative)
Z	Model evidence, also known as marginal likelihood

for each user, so as to predict unobserved choices, including those of previously unobserved users.

Rather than operating directly on order relations $\succeq_u$, some preference models estimate latent functions from which the order relations can be inferred. For example, the standard discrete choice models proposed by Thurstone (1927) and Bradley and Terry (1952) estimate functions $\widetilde{f_u} : X \to \mathbb{R}$ that capture the utility $\widetilde{f_u}(x)$ that user u derives from each instance x. When presented with a previously unobserved choice between instances x^1 and x^2, these models will predict that $x^1 \succeq_u x^2$ if and only if $\widetilde{f_u}(x^1) \geq \widetilde{f_u}(x^2)$.

Two key disadvantages of such approaches include the absolute interpretation of utility independently of context, and the stringent rationality assumptions that follow from this treatment. When making decisions individuals have been shown to focus on **trade-offs** resulting from their choices rather than on absolute outcomes and thus perceive alternatives within the context in which they are presented (Tversky and Simonson 1993). The assumption of utility models that individuals simply recall absolute, predetermined instance utilities $\widetilde{f_u}(x)$, and the strict transitivity of $\succeq_u$ implied by this assumption, are frequently violated in practice. Therefore, we represent our choices in terms of trade-offs $t = \tau(x^1, x^2)$, where τ is some mapping from $\mathbb{R}^{d_X} \times \mathbb{R}^{d_X}$ to $\mathbb{R}^{d_T}$, where $d_T \leq d_X$. For example, we might choose $\tau(x^1, x^2) = x^1 - x^2$.[1] If the dimensionality d_X of X is high, we might choose τ to be a dimensionality-reducing mapping, so that $d^T < d_X$; such an approach is supported by findings that, when d_X is large, consumers tend to base their decisions on a small subset of the dimensions of X (Hensher 2006).

Assume, for example, that electricity tariffs (i.e., rates or plans) are characterized by cost per kilowatt-hour and whether the electricity is generated from renewable sources. Given user u is presented with a choice between two alternative tariffs:

$$x^1 = \left[32\ \frac{¢}{\text{kWh}},\ 1\ \text{(renewable)}\right] \text{ and } x^2 = \left[28\ \frac{¢}{\text{kWh}},\ 0\ \text{(non-renewable)}\right]$$

our goal is to predict whether the user will prefer the first ($y = +1$ and $x^1 \succeq_u x^2$) or the second tariff ($y = -1$ and $x^2 \succeq_u x^1$). The trade-off the user faces is $t = (x^1 - x^2) = \left(4\ \frac{¢}{\text{kWh}}, 1\right)$, i.e., by choosing tariff x^1, the user pays an additional $4\ \frac{¢}{\text{kWh}}$ in exchange for a supply of renewable energy. In this formulation of the trade-off, our goal is to classify whether a given user will perceive the trade-off as favorable ($f_u(t)$ positive) and choose the first tariff, or perceive the trade-off as unfavorable ($f_u(t)$ negative) and choose the second tariff instead. Thus, we aim to classify trade-offs as favorable or unfavorable based on users' latent evaluations $f_u(t)$.

From a decision-theoretic perspective, our approach is inspired by the trade-off contrast principle (Tversky and Simonson 1993), and by case-based decision theory (Gilboa and Schmeidler 1995), which posits that a user's trade-off evaluation will resemble past evaluations of similar trade-offs. Compared with models that index functions based on raw input values, using trade-offs allows us to generalize preferences outside their original context, for example learning that a user will pay an extra $4\ \frac{¢}{\text{kWh}}$ for renewable energy, regardless of the base rate.

Given human preferences are latent and inconsistent, and because observed choices can be biased and distorted by noise (Evgeniou et al. 2005), we cast the problem in probabilistic terms, which accommodates these properties. Panel (a) in Fig. 1 outlines the generative pro-

[1] One can alternatively formulate the trade-off using percentage increases or any other relevant transformation. Such alternative transformations may increase the interpretability of the model's outputs; however, in our experiments we found them to have a negligible effect on the performance of our approach. We conjecture that it is also possible to learn the τ mapping from the data using, e.g., warped Gaussian processes (Snelson et al. 2004).

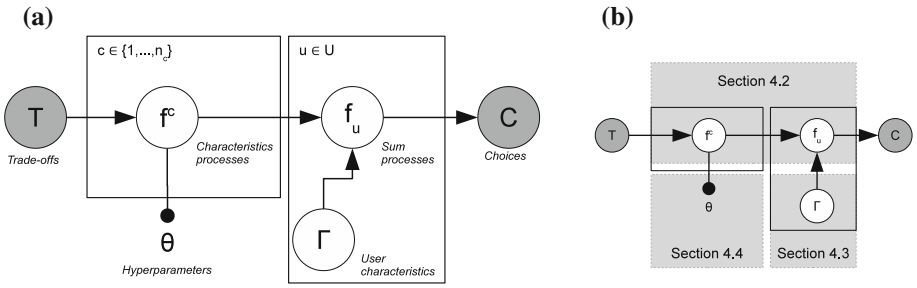

Fig. 1 Probabilistic graphical model of *GaSPK*. **a** users make choices based on their evaluations $f_u(t)$ of the associated trade-offs. Users' evaluations are linear combinations of n_c behavioral characteristics f^c which they possess to different degrees γ_u^c. Shaded circles represent observed data, white circles represent latent quantities of interest. The two **plates** in the figure denote replication of the enclosed elements for each characteristic c or user u, respectively. **b** Graphical table of contents for this article

cess underlying the **Ga**ussian process **S**calable **P**reference model via **K**ronecker factorization (*GaSPK*). Reading Panel (a) from right to left: users make observable choices C between alternatives based on their latent evaluation of trade-offs t, denoted as $f_u(t)$. Evaluations are modeled as linear combinations of n_c behavioral characteristics f^c which individuals possess to different degrees, denoted by $\gamma_u^c \in [0, 1]$:

$$f_u(t) = \sum_{c=1}^{n_c} \gamma_u^c \cdot f^c(t) \quad \text{with} \quad \sum_c \gamma_u^c = 1 \tag{1}$$

We let Γ_u denote the n_c-dimensional probability vector associated with user u, and Γ denote the $n_U \times n_c$ matrix of all γ_u^c.

The shared functions f^c can capture global patterns of behavior—e.g. frugality, environmental consciousness—that are exhibited in different quantities by different users (as determined by the weights Γ). Sharing information across users in this hierarchical manner allows us to draw statistical strength across users, leading to better predictions even when we have few choice observations for a particular user.

For now, we assume that Γ is known and focus on the problem of efficiently obtaining probabilistic estimates of the f^c. To do so in a Bayesian context, we start by placing some prior distribution $p(f^c)$ over these functions. We desire our prior to be flexible and make minimal assumptions about the functional form of the f^c. To this end, we select the nonparametric Gaussian process (GP) prior (Rasmussen and Williams 2006; MacKay 1998). A GP is a distribution over continuous functions $f : \mathbb{R}^d \to \mathbb{R}$ such that, for any finite set of locations in $\mathbb{R}^d$, the function evaluations have a joint multivariate Gaussian distribution. We write $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$, where m is a mean function (which we set to zero to reflect indifference in the absence of other information), and k is a covariance function that specifies how strongly evaluations of f at t and t' are correlated.

In our model, the input space is the product space of the d_T trade-off dimensions, and we evaluate the functions at the finite set of observed trade-offs T. A high value of the latent characteristic function f^c at trade-off t indicates that users with a large weight γ_u^c for that characteristic will tend to make a $+1$ choice at that trade-off. As is common when working with GPs, we employ squared exponential covariances of the form:

$$k(t, t') = \prod_{d=1}^{d_T} \exp\left(-\frac{\left(t^{(d)} - t'^{(d)}\right)^2}{2 \cdot l_d^2}\right) \tag{2}$$

This covariance structure captures the desirable property that evaluations $f(t)$, $f(t')$ of trade-offs t, t' become less correlated as the distance between them increases. This gives preference to smooth functions f, reflecting the intuition that individuals make similar choices when presented with similar trade-offs. The product structure of Eq. (2) corresponds to the assumption that each dimension of a trade-off contributes independently to the covariance, a property that will be crucial for efficient posterior inference. $\theta =:= \{l_d \mid d = 1, \ldots, d_T\}$ denotes the length-scale hyperparameters of the squared exponential covariance function. These length-scales characterize, in each dimension, the magnitude at which a trade-off becomes material to the users. Because the length-scales depend on the measurement of trade-offs in each dimension (e.g., dollars vs. cents), we will learn them from the data in Sect. 3.2.3.

Evaluating k at all pairs of observed trade-offs (t_1, t_2) yields the covariance (kernel) matrix K necessary for posterior inference. Importantly, the cost of many key operations on K grows cubically in the number of unique trade-offs, which presents naïve inference methods with significant scalability challenges. In Sect. 3.2 we show how the structure of our preference learning task can be exploited to substantially reduce this cost, yielding state-of-the-art scalability for our setting without significant loss in accuracy.

In order to complete our specification of the *GaSPK*, we must combine this prior distribution over functions with a realistic likelihood model, relating the latent functions f^c and the user-specific weights γ_u^c to the observed choices C. To translate the latent functions $f_u = \sum_c \gamma_u^c f^c$ into probabilities of making a given choice, we pass the f_u through a sigmoidal function, transforming the real-valued evaluations $f_u(t)$ to binomial probabilities $p_{u,t}$ and capturing the intuition that a +1 choice is more likely when the latent function takes on high values. The two most prominent candidates commonly used for such mappings are the Probit and Logit functions (Train 2003). Given that both functions can be computed efficiently and that no significant differences exist between them in terms of predictive accuracy (Rasmussen and Williams 2006), and since the marginal distribution of $f_u(t)$ is a normal distribution, we follow earlier work (Chu and Ghahramani 2005; Houlsby et al. 2012) and use the Probit likelihood:

$$p(y|f^c(t), \Gamma) = \begin{cases} \Phi\left(\overbrace{\sum_c \gamma_u^c \, f^c(t)}^{f_u(t)}\right) & \text{if } y = +1 \\ 1 - \Phi(f_u(t)) = \Phi(-f_u(t)) & \text{if } y = -1 \end{cases} \tag{3}$$

where Φ denotes the cumulative distribution function of the standard normal distribution.[2] The shape of our likelihood model is illustrated in Fig. 2.

We assume that the choices are independent conditioned on the latent functions, implying that the order in which the choices are observed is irrelevant (i.e. they are exchangeable). This exchangeability assumption is common in the preference modeling literature [see for example Thurstone (1927), Bradley and Terry (1952), Guo and Sanner (2010)], and allows us to write the joint likelihood as

[2] In some models, the Probit likelihood also includes a noise variance term, $p(y|\cdot) = \Phi\left(\frac{f_u(t)}{\sigma_n^2}\right)$. However, because our trade-off evaluation interpretation of the $f_u(t)$ is invariant under scaling, we set $\sigma_n^2 = 1$ without loss of generality.

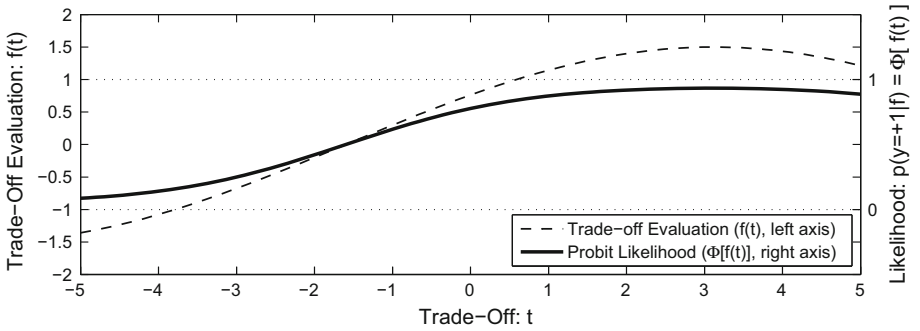

Fig. 2 Illustration of the Probit likelihood model. Probit likelihood model applied to a range of one-dimensional trade-offs t. The evaluation function $f(t)$ is fixed in the example. The bold line shows the Probit likelihood that assigns higher probabilities $p(y = +1|f(t))$ from the interval [0, 1] (dotted lines, right axis) to trade-offs with more positive evaluations

$$p(C|f^c) = \prod_i \Phi(y_i \cdot f_{u_i}(t_i))$$

Having specified both the likelihood model and the GP prior, we can now obtain a posterior distribution $p(f^c \mid C)$, reflecting our updated belief about the latent f^c. This belief can be used to predict users' choices with respect to unobserved trade-offs t_*. Specifically, the probability that a user u chooses the first alternative when presented with trade-off t_* is given by:

$$p(y_* = +1 \mid t_*, C) = \int \Phi(f_u(t_*))\, p(f_u(t_*) \mid C)\, df_u(t_*) \tag{4}$$

$$= \Phi\left(\frac{y \cdot E[f_u(t_*)]}{\sqrt{1 + Var[f_u(t_*)]}}\right) \tag{5}$$

3 Fast Bayesian inference in *GaSPK*

The modeling choices made in the basic *GaSPK* framework described in Sect. 2 are designed to give flexible and powerful modeling capacity, allowing us to obtain high-quality predictive performances. However, the goal of this work is to combine state-of-the-art performances with computational efficiency. As described in Sect. 2, a naïve implementation of the *GaSPK* will not scale well as we see more data, since GP inference typically scales cubically with the number of datapoints. Further, the non-Gaussian likelihood means we are unable to evaluate the posterior analytically, and must make judicious approximate inference choices to ensure scalability.

In this section we address these issues of scalability. We first introduce **modeling choices** that facilitate scalable inference (Sect. 3.1), then develop a **scalable approximate inference scheme** in Sect. 3.2. Our inference algorithm alternates between using Laplace's method to efficiently obtain the approximate posterior distribution $p(f^c|C) \approx q(f^c|C)$ of characteristic trade-off evaluations, and estimating the user characteristics Γ and the hyperparameters θ.

3.1 Structured Gaussian processes

When we condition on our finite set of trade-offs T, inferences about the f^c correspond to posterior inference in a multivariate Gaussian. Evaluating the covariance function k at all

pairs of observed trade-offs (t_1, t_2) yields the covariance (kernel) matrix K necessary for this posterior inference. Importantly, the cost of many key operations on K grows cubically in the number of unique trade-offs, which presents naïve inference methods with significant scalability challenges.

However, since our covariance structure factorizes across dimensions (Eq. 2), if we are able to arrange our inputs on a grid, we can formulate our model using Kronecker covariance matrices. Kronecker covariance matrices have favorable factorization and decomposition properties that, as we describe in this section, facilitate scalable inference. While Kronecker-structured covariances have appeared in other preference learning models (Bonilla et al. 2010; Birlutiu et al. 2013), we believe we are the first to exploit their computational advantages in this context.

In particular, as we will see later in this section, Kronecker covariance matrices are particularly appealing when our input space can be expressed as a low-dimensional grid with a fairly small number of possible values along each dimension. Our problem setting is well suited to the use of such a structure. Consumer and econometric research has established that consumers focus on relatively small subsets of attributes as well as few possible values thereof when choosing amongst alternatives, e.g., Caussade et al. (2005) and Hensher (2006). Motivated by this, we consider settings in which (1) the number of users, instances, and observed choices is large and naïve methods are therefore computationally infeasible; (2) trade-offs can be represented by a small number of attributes; and (3) each attribute has a small number of values, or can be discretized. We show that when alternatives can be represented by a small number of attributes and values, it is possible to obtain matrices K which are large, but on which important operations can be performed efficiently. In the empirical evaluations that follow, we demonstrate that this approach yields computational advances but also, despite introducing approximations, produces predictive performance that is often superior to what can be achieved with current scalable approaches.

Concretely, we assume that trade-offs can be arranged on a d_T-dimensional grid, and let T_d denote the set of unique values that occur on the dth attribute in T. In our electricity tariffs example, trade-offs can be characterized by (1) price differences per kWh, and (2) differences in renewable sources, so that we may have the following unique trade-off values: $T_1 = \{-0.10, -0.09, \ldots, 0.09, 0.10\}$ and $T_2 = \{-1, 0, 1\}$. Not all possible combinations of trade-offs are always observed ($|T| < |T_1| \cdot |T_2| = 63$), and the covariance matrix $\widetilde{K} = \left[k(t, t')\right]_{t,t' \in T}$ is therefore significantly smaller than 63×63. A Gaussian process applied to such a structured input space is known as a structured GP (Saatci 2011).

The key notion of structured GPs is that, rather than working directly with $\widetilde{K}$, we can instead work with a larger matrix of the form (Saatci 2011):

$$K = K_1 \otimes \cdots \otimes K_{d_T}$$

where $\otimes$ denotes the Kronecker product.[3] The entries K_d hold the covariance contributions of the d-th dimension and they are generally much smaller than $\widetilde{K}$ (in our example, $K_1 \in \mathbb{R}^{21 \times 21}$ and $K_2 \in \mathbb{R}^{3 \times 3}$). The Kronecker matrix K, on the other hand, holds the covariances between

[3] For two arbitrarily sized matrices A, B, the Kronecker product is defined as:

$$A \otimes B := \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

all trade-offs in the Cartesian product $\times_d T_d$, and it is thus much larger (in our example, $K \in \mathbb{R}^{63\times 63}$).

The significant computational savings that the Kronecker structure of K outlined above enables follow from the fact that, instead of explicitly generating and manipulating $\widetilde{K}$, it is now possible to operate on the smaller, K_d. In this setting, several key matrix operations involving K can be performed efficiently. Most importantly:

- Matrix-vector products of the form Kb can be computed at a cost that is linear in the size of b, in contrast to the quadratic cost entailed by standard matrix-vector products. This follows from the fact that $\left(K_i \otimes K_j\right) \mathrm{vec}(B) = \mathrm{vec}\left(K_i B K_j^T\right)$, where $b = (B)$; since the number of nonzero elements of B is the same as the length of b, this operation is linear in the length of b.
  As we will see in Algorithm 2, such products are required to find the posterior mode of our GPs and in general dominate the overall computational budget; this speed-up means that they are no longer the dominant computational cost.
- Eigendecompositions of the form $K = Q^T \Lambda Q$ can be computed from the Eigendecompositions of the K_d:

$$Q = \bigotimes_{d=1}^{D} Q_d \quad \Lambda = \bigotimes_{d=1}^{D} \Lambda_d$$

  at cubic cost in the size of the largest K_d. This is a consequence of the mixed product property of Kronecker products, that states that $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ and therefore,

$$(Q_i \Lambda_i Q_i^T) \otimes (Q_j \Lambda_j Q_j^T) = \left((Q_i \Lambda_i) \otimes (Q_j \Lambda_j)\right)(Q_i^T \otimes Q_j^T) = (Q_i \otimes Q_j)(\Lambda_i \otimes \Lambda_j)(Q_i \otimes Q_j)^T.$$

  In particular, this allows us to efficiently determine the Eigenvectors to the n_e largest Eigenvalues of K, allowing us to obtain computational speed-ups by replacing K with a low-rank approximation.

Furthermore, note that all operations can be implemented by considering only the set of unique, observed or predicted trade-offs. This reduces the region under consideration, from the large space covered by K to a manageable superset of T. Unobserved trade-offs can be modeled through infinite noise variances in Eq. (3). The corresponding likelihood terms then evaluate to indifference ($p = 0.5$), and their derivatives to zero. The latter yield even sparser matrices W and L in Algorithms 2 and 4 below, which can be directly exploited via standard sparse matrix operations.

3.2 Approximate inference in *GaSPK*

The Kronecker structure described above has proved useful in a regression context, but requires careful algorithmic design to ensure its benefits are exploited in the current context. In Sect. 3.2.1, we develop a scalable inference algorithm using Laplace's method to estimate the posterior distributions $p(f^c|C, \Gamma)$.

In a full Bayesian treatment of *GaSPK*, we would consider Γ another latent quantity of interest, and infer its posterior distribution. Previous work has addressed similar challenges by either imposing a Gaussian or a Dirichlet process prior on Γ (Houlsby et al. 2012; Abbasnejad et al. 2013). However, these approaches are computationally expensive, and it can be hard to

interpret the resulting joint distribution over weights and characteristics. Instead, we treat Γ as a parameter to be estimated; in Sect. 3.2.2 we show that we can either find the maximum likelihood value by optimization, or find a heuristic estimator that we show in Sect. 5 performs well in practice at a much lower computational cost.

We combine these two steps in an EM-type algorithm (Dempster et al. 1977) that jointly learns Γ and the posterior distribution over the f^c. The algorithm is outlined in Algorithm 1.

Algorithm 1 An EM-type algorithm for learning f^c and Γ. GAMMAESTIMATOR can be either the ML estimator for Γ, or the heuristic estimator described in Algorithm 4.

function EMWRAPPER(covariance matrix K, choices C, # characteristics n_c)
  $\Gamma \leftarrow$ random user characteristics
  repeat
    $\mathrm{E}\left[\hat{f}^c\right] \leftarrow$ LAPLACEMODE(K, C, Γ)
    $\hat{\Gamma} \leftarrow$ GAMMAESTIMATOR($C, E[\hat{f}^c]$)
  until $\Gamma, E[\hat{f}^c]$ converge
return user characteristics Γ, characteristic function modes $E[\hat{f}]$
end function

In the E-step, we use Laplace's method to approximate the conditional expectation $E[f^c|C, \Gamma]$ with the posterior mode $E[\hat{f}^c|C, \Gamma]$, as described in Sect. 3.2.1. We then obtain one of the two estimators for Γ described in Sect. 3.2.2—an optimization-based estimator that corresponds to the exact M-step but is slow to compute, or a heuristic-based estimator that is significantly faster to compute. In practice, we suggest using the heuristic-based estimator; as we show in Sect. 5 this approach strikes a good balance between predictive performance and computational efficiency.

3.2.1 Learning the latent functions f^c conditioned on Γ

Inferring the f^c is complicated by the fact that the posterior $p(f^c|C, \Gamma)$ is analytically intractable under the Probit likelihood. Discrete choice models often use sampling-based methods to approximate the posterior (Allenby and Rossi 1998; Train 2003). However, sampling is slow, particularly for high-dimensional models based on GPs. Alternatives include Laplace's method, Expectation Propagation, and Variational Bayesian methods, all of which seek to approximate $p(f^c|C)$ with a similar distribution $q(f^c|C)$ that can be computed and represented efficiently (Bishop 2006).

In this paper we use Laplace's method, because it is computationally fast and conceptually simple. Laplace's method is a well known approximation for posterior inference in regular GPs (Rasmussen and Williams 2006) and simpler preference learning scenarios (Chu and Ghahramani 2005). Laplace's method aims to approximate the true posterior p with a single Gaussian q, centered on the true posterior mode $\hat{f}^c$, and with a variance matching a second-order Taylor expansion of p at that point (see Fig. 3). Approximating the posterior with a single multivariate Gaussian allows us to conveniently re-use it as the prior in subsequent Bayesian updates which is important for online and active learning from user interactions (Saar-Tsechansky and Provost 2004). While the approximation can become poor if the true posterior is strongly multi-modal or skewed, prior work has shown this limitation has no significant impact in the preference learning context, e.g., Chu and Ghahramani (2005).

In principle, we could directly apply the Laplace mode and variance calculations used by Chu and Ghahramani (2005), which assume a full covariance matrix. However, doing so

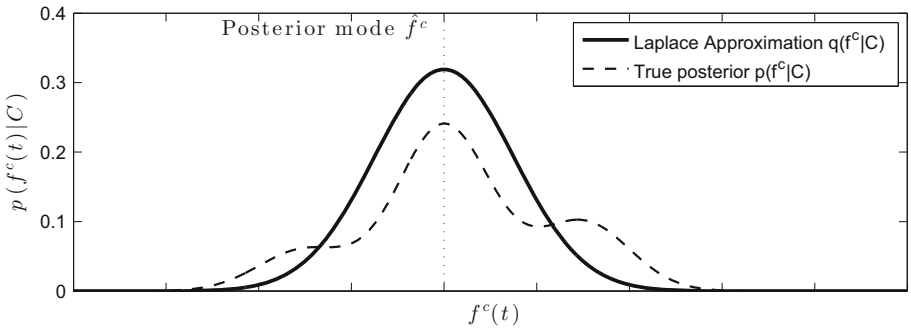

Fig. 3 Laplace approximation $q(f^c|C)$ of the true posterior $p(f^c|C)$ The solid line shows the approximation $q(f^c|C)$ of the true posterior $p(f^c|C)$ for a one-dimensional marginal distribution. The approximation is centered on the mode $\hat{f}^c$ of the true posterior and its variance is matched to a second-order Taylor expansion of the true posterior at that point

would negate the benefit of using a structured covariance function. Instead, we formulate our calculations to exploit properties of our covariance matrix, yielding an algorithm which, as we show later in this section, has better scaling properties directly applying the algorithms in (Chu and Ghahramani 2005).

Our development of Laplace inference in *GaSPK* proceeds in two steps. First, we describe an efficient procedure for finding the posterior mode $\hat{f}$ (Algorithm 2). We then describe how the posterior variance and predictions for new trade-offs t_* can be computed (Algorithm 3). Additional mathematical details are provided in "Appendix A".

The mode $\hat{f}$ of the posterior is the maximizer of the log posterior $\log p(f^c|C, \Gamma) \propto \log p(C|f^c, \Gamma) + \log p(f^c)$ which can be found by setting the first derivative of $\log p(f^c|C, \Gamma)$ to zero and solving for f^c. Because the Probit likelihood is log concave, there exists a unique maximum $\hat{f}$, which we obtain iteratively by using the Newton–Raphson method (Press et al. 2007) with the update step

$$\begin{aligned} f^{new} &= (K^{-1} + W)^{-1} \underbrace{(Wf + \nabla \log p(C|f))}_{b} \\ &= K(b - L(I + L^T K L)^{-1} L^T K b). \end{aligned} \tag{6}$$

We repeatedly assign $f \leftarrow f^{new}$ and recompute Eq. (6) until f converges. The matrix W in the first line of Eq. (6) denotes the negative Hessian of the log likelihood, $W = -\nabla\nabla \log p(C|f^c, \Gamma)$, a sparse matrix consisting of $n_c \times n_c$ diagonal sub-matrices of size $n_T \times n_T$. W is computed using Eq. (10) described in "Appendix A.1", along with additional computational details regarding our Probit likelihood. The sparsity of W allows us to compute its Cholesky decomposition $W = LL^T$ in $O(n_T n_c^3)$ time, rather than the $O(n_c^3 n_T^3)$ time that would be typical of a dense matrix. We use this decomposition instead of W in the second line of Eq. (6), eliminating the numerically unstable K^{-1} and the unwieldy inverse of the first factor in the previous line. All matrices in the second line of Eq. (6) are of size $(n_c n_T) \times (n_c n_T)$ and therefore usually large. However, as we discuss in Sect. 3.1, L has at most $\frac{n_T n_c (n_c - 1)}{2}$ non-zero elements (less if not all possible trade-offs from T are observed), and thus it is never necessary to generate K explicitly.

Using Eq. (6), we can efficiently compute the posterior mode by following the steps outlined in Algorithm 2. Note, that all operations in the algorithm are simple matrix operations available in most programming environments. Furthermore, the operations in lines 6 through

Algorithm 2 Laplace mode finding

1: **function** LAPLACEMODE(covariance matrix K, choices C, user characteristics Γ)
2: $f = \mathbf{0}$
3: **repeat**
4: $W \leftarrow -\nabla\nabla \log p(C|f, \Gamma)$
5: $L \leftarrow \text{CHOLESKY}(W)$
6: $b \leftarrow Wf + \nabla \log p(C|f, \Gamma)$
7: $a \leftarrow b - L(I + L^T KL)^{-1} L^T Kb$ ▷ using conjugate gradients
8: $f \leftarrow Ka$
9: **until** f converges
10: **return** posterior mode $\hat{f}$
11: **end function**

8 are all matrix-vector operations which generate vectors as intermediate results. Rather than calculating the inverse in line 7 explicitly, we use conjugate gradients (Press et al. 2007) to solve the system $(I + L^T KL)x = L^T Kb$ by repeatedly multiplying the parenthesized term with candidates for x, as in Cunningham et al. (2008).

Because K has Kronecker structure and L consists only of diagonal sub-matrices, multiplications with K and L have linear time and space complexity, hence the overall computational cost is dominated by the $O(n_T n_c^3)$ cost of the Cholesky decomposition. Without the Kronecker structure, these multiplications would be $O(n_T^2 n_c^2)$, and their cost would therefore dominate when $n_T > n_c$.

We next compute the variance $V_q(f)$ of the approximate posterior q, which can be written as (Rasmussen and Williams 2006):

$$V_q(f) = \text{diag}(K) - \text{diag}(KL(I + L^T KL)^{-1} L^T K) \tag{7}$$

The computations in Eq. (7) involve full matrix operations, and are therefore more expensive than the matrix-vector operations used for mode-finding. However, we can limit the computations to points of interest t_* only, which reduces the number of rows in K being considered. To further reduce the size of the involved matrices, we approximate K via a low-rank decomposition with exact diagonal given by:

$$K \approx QSQ^T + \Lambda, \text{ where } \Lambda = \text{diag}(K) - \text{diag}(QSQ^T) \tag{8}$$

Importantly, the decomposition can be efficiently computed when K has Kronecker structure, as discussed in Sect. 3.1. Specifically, the matrix S in Eq. (8) is a diagonal matrix with the n_e largest Eigenvalues of K on its main diagonal. Q contains the corresponding Eigenvectors, and it has the same number of rows as K but only n_e columns. Λ is a diagonal matrix of the same size as K, making the low-rank approximation of K exact on the diagonal (Quiñonero-Candela and Rasmussen 2005; Vanhatalo et al. 2010). The number of Eigenvalues n_e in the approximation is a user-defined input and it can be used to balance computing time against accuracy of the approximated posterior variance. As we will show below, even choices of small numbers of Eigenvalues n_e often yield posterior variances close to those obtained with the full matrix K. Under this low-rank approximation, Eq. (7) can be re-written as:

$$\begin{aligned} V_q(f) &\approx \text{diag}(K) - \text{diag}(KL(I + L^T(QSQ^T + \Lambda)L)^{-1} L^T K) \\ &= \text{diag}(K) - \text{diag}(K\Pi K) + \text{diag}(K\Pi Q(\underbrace{S^{-1} + Q^T \Pi Q}_{P})^{-1} Q^T \Pi K) \end{aligned} \tag{9}$$

where P is a small matrix of size $n_e \times n_e$, and where $\Pi = L(I + L^T \Lambda L)^{-1} L^T$ can be computed efficiently, because L is sparse and Λ is diagonal. Π itself is also sparse, consisting

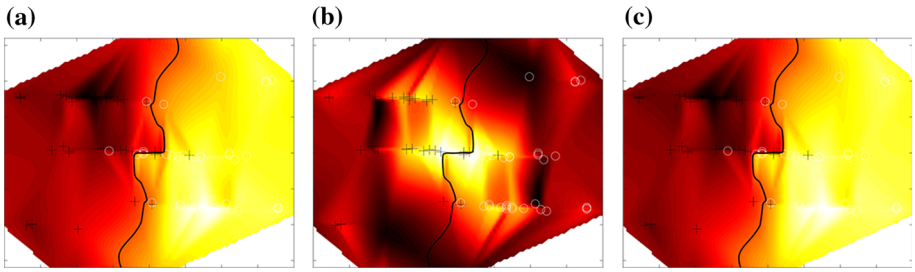

Fig. 4 Outputs of Algorithms 2 and 3 for a single user from a popular preference benchmark dataset. Observed choices are represented by black pluses (favorable trade-offs) and white circles (unfavorable trade-offs); darker colors represent higher values; bold lines represent the boundaries where $E[f_u] = 0$ (**a**, **b**), or $p(y = +1|f) = 0.5$ (**c**). The diamond-like shape of the plot results from mapping the four-dimensional trade-off space to two dimensions using its first two principal components. **a** Posterior mode, **b** posterior variance and **c** prediction (Color figure online)

of $n_c \times n_c$ diagonal blocks like W. Because K has Kronecker structure, the first two terms in Eq. (9) can be computed efficiently and without resorting to approximations. We address the computation of the third term next.

Algorithm 3 Laplace prediction

1: **function** LAPLACEPREDICT(covariance matrix K, choices C, user characteristics Γ, posterior mode $\hat{f}$, trade-offs T_*, # Eigenvalues n_e, Cholesky factor L)
2:
3: $QSQ^T + \Lambda \leftarrow$ LOWRANKAPPROXIMATION(K, n_e) ▷ Equation (8)
4: $\Pi \leftarrow L \cdot$ FORWARDSOLVE($I + L^T \Lambda L, L^T$)
5: $C \leftarrow$ CHOLESKY($S^{-1} + Q^T \Pi Q$)
6: $V = K_* \cdot$ BACKWARDSOLVE($\Pi Q, C$)
7: $v_* \leftarrow \text{diag}(K_*) - (K \circ K) \cdot \text{diag}(\Pi)\big|_* + \sum_j [V \circ V]_{i,j}$ ▷ Equation (9)
8:
9: $p_* \leftarrow \Phi\left(\frac{\hat{f}_*}{\sqrt{1+v_*}}\right)$ ▷ Equation (4)
10: **return** posterior variances v_*, predictive probabilities $p_* = p(y = +1|f, T_*)$
11: **end function**

In Algorithm 3, we first calculate the Cholesky factor C of P (line 5), which is subsequently used in solving[4] the system ΠQC^{-1}. The product V in line 6 is equivalent to n_e matrix-vector products with a Kronecker matrix and is computationally inexpensive when n_e is sufficiently small. In line 7, we exploit the symmetry of the third term in Eq. (9), and the fact that only its diagonal is needed, to reduce calculations to an efficient element-wise product of the smaller V. Finally, in line 9, we use the posterior variances to calculate the predictive probabilities p_* at the trade-off points T_* using Eq. (4).

Figure 4 illustrates the output of Algorithms 2 and 3 for the choices of a single user, using data from a popular preference benchmark dataset (Kamishima and Akaho 2009). Panel (a) shows the posterior mode $\hat{f}_u = E[f_u]$, which is expectedly high in regions of the trade-off space perceived as favorable, and low otherwise. The bold line indicates the zero boundary $\hat{f}_u = 0$, and it is sufficient as a predictor of future choices when predictive certainty estimates

[4] FORWARDSOLVE denotes the operation that solves the linear system $Ax = b$ for x. BACKWARDSOLVE similarly solves $xA = b$.

are not required. Importantly, it can be computed using only Algorithm 2 and is therefore very fast.

The key distinguishing feature of our probabilistic approach are the variance estimates shown in Panel (b). As shown, the algorithm correctly identifies the region at the center of the panel where the decision boundary already follows a closely determined course to match earlier observations (pale yellow coloring, low variance). If additional observations were to be acquired for the purpose of improving predictions, they should be located in the upper or lower regions of the decision boundary instead, where fewer evidence is currently available (dark red coloring, high variance). Panel (c) shows the combination of both outputs to compute the predictive probabilities $p(y = +1|f)$. While the decision boundary at $p(y = +1|f) = 0.5$ is the same as the one in Panel (a), this panel also incorporates predictive variances by shrinking the predictive probabilities towards indifference ($p = 0.5$) in high-variance regions [see Eq. (4)]. Consequently, the corridor in which *GaSPK* is indifferent (intermediate intensity orange coloring, intermediate probabilities) is narrower in areas with extensive evidence from the data, and wider towards the edges of the panel. This information is an important input to subsequent decision-making tasks which require information on whether existing evidence is conclusive enough to make an autonomous decision.

3.2.2 Learning user characteristics

To complete our EM-type algorithm, we must estimate the user characteristics $\Gamma = \left[\gamma_u^c\right]_{u,c}$ from the data. Recall from Sect. 2 that γ_u^c denotes the fraction of user u's behavior explained by characteristic c, that is, $f_u(t) = \sum_c \gamma_u^c \cdot f^c(t)$ with $\sum_c \gamma_u^c = 1$. An exact M-step estimator for Γ, that returns $\arg\max \prod_i \Phi\left(y_i \sum_c \gamma_{u_i}^c \cdot f^c(t_i)\right)$ s.t. $\sum_c \gamma_u^c = 1, u = 1, \ldots, U$, can be obtained using an interior-point optimizer. This yields a (local) optimum for Γ, but is more computationally expensive.

As an alternative, we propose a heuristic approximation to this M-step, described in Algorithm 4. We note that if $\gamma_u^c > \gamma_u^{c'}$, then f_u is likely to be closer to f^c than to $f^{c'}$. Therefore, approximating f_u with f^c is likely to give a higher likelihood than approximating f_u with $f^{c'}$. The heuristic "M-step" in line 3 computes an approximation to the likelihood that characteristic c alone generated the observed choices. Each iteration of the surrounding loop calculates one column of the Γ matrix, corresponding to one characteristic. The resulting user characteristics are then re-scaled so that they add to one in line 5.

As we will see in Sect. 5, while it lacks the theoretical justification of the exact M-step, empirically the heuristic Algorithm 4 obtains good results with much lower computational cost. Finally, while the number of user characteristics n_c has to be set manually, we find that consistent with prior work, our method is insensitive to the choice of this parameter when it is not excessively small, e.g., Houlsby et al. (2012).

Algorithm 4 A heuristic estimator for the user characteristics Γ

1: **function** HEURISTICGAMMAESTIMATOR(function modes $\mathrm{E}\left[\hat{f}^c\right]$, choices C, # characteristics n_c)
2: **for** $c = 1 : n_c$ **do**
3: $\Gamma_{*,c} \leftarrow \prod_i \Phi(y_i \cdot \mathrm{E}\left[\hat{f}_{t_i}^c\right])$
4: **end for**
5: NORMALIZEROWS(Γ)
6: **return** user characteristics Γ
7: **end function**

3.2.3 Learning hyperparameters

As in the case of Γ, a full Bayesian treatment the hyperparameters, $\theta = \{l_d\}$, is prohibitively expensive. Prior work has often resorted to either gradient-based optimization of the marginal likelihood Z, e.g., Chu and Ghahramani (2005), or to heuristics, e.g., Stachniss et al. (2009) to learn the hyperparameters from the data. In the experiments that follow, we employ a heuristic and set the length-scales to the median distance between trade-offs t. This has been found in prior work to be a computationally fast heuristic yielding consistently good empirical results (Houlsby et al. 2012).

4 Related work

Machine learning research has produced preference models based on a broad variety of learning frameworks (Fürnkranz and Hüllermeier 2011). Of particular interest to this research is work on *probabilistic* preference models that derive principled uncertainty estimates from noisy preference observations. Chu and Ghahramani (2005) were the first to model preference learning using Gaussian processes. However, that approach does not capture heterogeneity across users—an essential property for modeling large, heterogeneous sets of users. More recent work (Bonilla et al. 2010; Birlutiu et al. 2013; Houlsby et al. 2012) has alleviated this shortcoming; however, these contributions have focused on solutions for incorporating heterogeneous preferences, rather than ensuring scalability.

More specifically, the **Hierarchical GP** model (Birlutiu et al. 2013) is derived from a semi-parametric Bayesian formulation that builds on the framework proposed by Bradley and Terry (1952). The authors model each user's utility function using a GP, which they represent using a basis decomposition $f_u(x) = w_u^T \phi(x)$. A hierarchical Gaussian prior on the base weights w_u induces correlations between users (hence our choice of name for the approach). An EM-type algorithm is then used for learning, which iteratively refines the parameters of the hierarchical prior. While the Hierarchical GP model offers state-of-the-art accuracy, inference is computationally expensive since we need to effectively learn a Gaussian process for each user.

The **Collaborative GP** method of Houlsby et al. (2012) also builds on Bradley and Terry (1952). Like *GaSPK*, it represents users' utility functions using a weighted superposition of globally shared GPs. Unlike *GaSPK*, the weights are unnormalized; this adds a redundant degree of freedom which makes interpretability harder. Further, the weights are treated as random variables to infer rather than parameters to optimize, increasing the computational burden. Another key distinction between *GaSPK* and Collaborative GP is that the latter operates on pairs of alternative instances (x^i, x^j) instead of the associated trade-offs $t = \tau(x^i, x^j)$, and it estimates instance utilities rather than trade-off evaluations. This makes inference in the model significantly more demanding, and the authors employ a combination of Expectation Propagation and Variational Bayes to address this challenge. As shown in Sect. 5, this design choice yields comparable accuracies to those produced by the Hierarchical GP method at lower computational cost, likely due to the smaller number of GPs. However, in general the Collaborative GP will not scale as well as *GaSPK*: inference still scales cubically in the number of distinct trade-offs, and approximating the full posterior over weights adds computational complexity.

In the limit of a single latent characteristic, Collaborative GP reduces to regular **GP classification** with a specific preference kernel (Rasmussen and Williams 2006; Houlsby et al. 2012). Inference in this model is fast and conceptually simple, and as such GP classification

constitutes the strongest computational benchmark for *GaSPK*. As shown in Sect. 5, *GaSPK* is valuable as it can both achieve GP classification's computational performance as well as a substantial improvement in predictive accuracy.

Bonilla et al. (2010) presented an earlier GP approach that aimed to accommodate heterogeneity amongst users. However, their approach has been shown to be inferior in both predictive accuracy and computational cost relative to other state-of-the-art approaches (Houlsby et al. 2012). Note, that Bonilla et al. (2010) make use of a single Kronecker product to multiply one item-covariance matrix with one user-covariance matrix. The purpose of this product is fundamentally different from the manner in which the Kronecker product is used in *GaSPK*, namely to deal with the growing item-covariance matrix. Furthermore, in Bonilla et al. (2010), capturing heterogeneity yields an even larger matrix, making the resulting method extremely slow, as noted and demonstrated in Houlsby et al. (2012). By contrast, *GaSPK* uses $(D-1)$ Kronecker products, where D refers to the dimensionality of the data, and the Kronecker product is used to factor the trade-off-covariance matrix, thereby effectively addressing its growth.

Marketing and Econometrics research considered preference measurement methods such as conjoint analysis, Logit and Probit models, and other discrete choice prediction techniques (Greene 2012). Early preference measurement was limited to population-level estimates, but more recent techniques accommodate heterogeneity across consumer segments (Allenby and Rossi 1998; Evgeniou et al. 2007). The primary objective of these models is to inform human decision makers, and thus their outputs are interpretable coefficients. By contrast, *GaSPK* work focuses on preference learning for use in autonomous decision-making settings, and has to consider scalability, incremental updates, and other practical issues that arise when moving from passive preference measurements to autonomous decision-making (Netzer et al. 2008). In Sect. 5, we illustrate these differences by benchmarking *GaSPK* against the Mixed Logit model, a well-established standard in Marketing and Econometrics. The **Mixed Logit** model estimates $f_u^i = w_u x_u^i + \varepsilon_u^i$ where ε_u^i is extreme-value distributed, and the w_u are drawn from a hierarchical prior. Like the other benchmarks, Mixed Logit accommodates random variations in taste among users. This makes inference more difficult than in the standard Logit model—a challenge that is addressed by a computationally expensive sampling procedure. Moreover, as demonstrated in our empirical evaluations, Mixed Logit is less flexible to adapt to the data in comparison to the non-parametric models.

5 Empirical evaluation

In the empirical evaluations that follow we consider learning preferences in consumer choice settings and aim to evaluate whether *GaSPK* offers a valuable addition to the existing set of non-parametric Bayesian approaches that similarly provide principled uncertainly estimates. To do so, we compare *GaSPK* to three non-parametric Bayesian approaches shown to yield state-of-the-art performance either in scalability or predictive accuracy. We compare the heuristic estimator of Algorithm 4 with an exact M-step, and show that the heuristic method offers comparable accuracy with much lower computational cost. Further, we show that, compared with other methods, *GaSPK* (using this heuristic estimator) offers an impressive combination of predictive accuracy and computational efficiency. Our evaluations are performed on an electricity tariff preference dataset collected specifically for this work, as well as two benchmark datasets used earlier in the literature. In our implementation of *GaSPK* we

used the GP toolbox (Vanhatalo et al. 2013), and we make both our code and data publicly available at https://bitbucket.org/gtmanon/gtmanon.

5.1 Datasets and benchmark methods

The first benchmark with which we compare *GaSPK* is **GP classification** (Rasmussen and Williams 2006; Houlsby et al. 2012), a non-parametric Bayesian method that exhibits state-of-the-art scalability and which *GaSPK* aims to approximate. Inference was performed using expectation propagation. Because scalability often comes at the cost of predictive accuracy, this evaluation aims to establish whether *GaSPK* is able to match GP classification for an increasing number of users, and to produce better predictive accuracy, thereby offering an advantageous augmentation of the non-parametric Bayesian toolset for a large number of users.

We also compare GP classification and *GaSPK* to non-parametric Bayesian approaches that are shown to yield state-of-the-art predictive performance but to be computationally intensive. These comparisons aim to assess the relative loss in accuracy incurred by *GaSPK* and GP classification to produce their respective scalabilities. To do so, we evaluate the performance of **Hierarchical GP** (Birlutiu et al. 2013) and **Collaborative GP** (Houlsby et al. 2012), both shown to yield state-of-the-art accuracy but to be computationally more expensive. As done in prior work (Houlsby et al. 2012), to allow evaluations with these computationally intensive methods, the data sets used in the empirical evaluations include a moderate number of users. As we will see, the differences in the scalabilities are clearly apparent for these data sets, and the performances differ in order of magnitude.[5]

We also contrast the non-parametric Bayesian approaches with the well-established, parametric **Mixed Logit** model (see above). These results will aim to establish the benefits from adopting a non-parametric Bayesian framework in our setting. Mixed Logit estimates $f_u^i = w_u x_u^i + \varepsilon_u^i$ where ε_u^i is extreme-value distributed, and the w_u are drawn from a hierarchical prior. Like the other benchmarks, Mixed Logit accommodates variations in taste among users. This makes inference more difficult than in the standard Logit model, a challenge that is addressed by a computationally expensive sampling procedure. Moreover, in comparison to the non-parametric models, Mixed Logit is also less flexible to adapt to the data. In the evaluations reported below we used the implementation by Train (2003).

We evaluated the methods on three preference datasets collected from human decision-makers. Recall that a key motivation for this work is the need for computationally fast and scalable preference models towards contemporary applications, such as to automate decisions in dynamic energy markets. One application domain of significant global importance is the modeling of electricity tariff choices of smart grid consumers. In future smart grids, tariffs may be revised frequently and automatically to reflect changes in the cost and availability of renewable energy (such as solar or wind); consequently, tariff choice is expected to become a near-continuous process in which both retailers and customers will rely on automated, data-driven decision agents. The ability to predict and act upon tariff choices quickly and with adequate accuracy is therefore an important challenge. To evaluate our approach in this setting, we used data on real electricity tariffs from the Texas retail electricity market. This retail market is the most advanced in the United States, and it provides daily information on

[5] Even for such small size data sets it was not possible to evaluate the method proposed by Bonilla et al. (2010)—as noted by the authors, this method is not suitable for modeling a large number of users. This is in agreement with the findings of Houlsby et al. (2012), who show that the method of Bonilla et al. (2010) is both slower and achieves lower predictive performance than the Collaborative GP. This limitation underscores the practical significance of scalable approaches with respect to the number of users.

Table 2 Characteristics of the datasets used in this study

Dataset	Instances	Users	Trade-offs stated preferences	Orig. dim.	Sel. dim.	Grid size
Tariffs	261	61	610	9	9	12,288
Cars	10	53	2362	5	5	216
Elections	8	264	7392	20	8	30,375

Instances, **Users**, and **Trade-Offs** refer to the number of elements in X, U, and T, respectively. **Orig. Dim.** and **Sel. Dim.** refer to the number of trade-off dimensions (the size of each t) before and after feature selection (note that feature selection is only done before *GaSPK*, not the comparison methods). And **Grid Size** refers to the number $|T|$ of points on *GaSPK*'s grid

available tariffs (see http://www.powertochoose.org). Using the Amazon Mechanical Turk crowdsourcing platform, we acquired data on American participants' choices between pairs of tariffs offered in Austin, Texas in February 2013. Tariff preferences were acquired on randomly drawn tariff pairs from a set of 261 tariffs, and all modeling techniques were evaluated on predicting consumers' preference for the same pairs of tariffs.

"Appendix B" provides complete details on the Tariffs data set collection, as well as an example of tariff choice (Table 3). The Tariffs data set reflects important characteristics of many real world applications where the data correspond to many choice alternatives (tariffs), but relatively few observed choices per individual user (see Table 2). As commonly encountered in practice, choices of different users likely correspond to different alternatives and are thus sparsely distributed and difficult to generalize from. This property of the Tariffs dataset is common in real world applications, but is not reflected in other benchmark datasets.

Our evaluations on the Tariffs data set are complemented with two benchmark datasets. Specifically, we used the **Cars** dataset that contains stated preferences for automobile purchases (Abbasnejad et al. 2013), and the **Elections** dataset compiled by Houlsby et al. (2012), which captures revealed voters' preferences over eight political parties in the United Kingdom. The full Elections dataset contains 20 trade-off dimensions, resulting in a Kronecker covariance matrix that was too large to hold in memory. As described in Sect. 3.1, our trade-off function $\tau(x^1, x^2)$ need not involve all dimensions of X, and indeed prior research (Hensher 2006) indicates that, when the number of dimensions are large, users tend to base their choices on a smaller subset of dimensions. We therefore applied greedy forward feature selection to reduce the dimensionality of this datasets to a subset of important predictive features, such that the accuracies after feature selection were comparable to those reported by Houlsby et al. (2012) on the complete feature set using the most accurate benchmark method (Birlutiu et al. 2013).

Since our comparison methods do not involve the large Kronecker matrix, we ran them on two versions of the Elections dataset: one with all 20 covariates, and one with the 8 covariates used by *GaSPK*. As we will see in Sect. 5.2, for each model, using the full set of covariates only yields a fairly modest improvement in performance. That much of the information content was maintained by the feature selection procedure reaffirms prior findings which *GaSPK* exploits, namely, that a subset of relevant dimensions often effectively informs human choices. A summary of the key characteristics of these datasets is presented in Table 2.

GaSPK was applied to versions of the datasets in which the continuous attributes were discretized to between 5 and 25 levels, with the objective of minimizing information loss

while keeping the resulting grid size manageable.[6] All other methods ran on the original, non-discretized datasets. We employed the Natural Breaks algorithm (Jenks and Caspall 1971) to identify bins for discretization. Natural Breaks is a univariate variation of the k-means algorithm, which selects bin boundaries such that within-bin variances are minimized while between-bin variances are maximized.

In the empirical evaluations below we will aim to evaluate whether *GaSPK* offers advantageous improvements over the existing, scalable GP classification. Simultaneously, the state-of-the-art predictive accuracies exhibited by Hierarchical GP and Collaborative GP allow us to assess the reduction in predictive accuracy that GP classification and *GaSPK*'s computational benefits entail.

5.2 Model scalability and predictive accuracy

The learning curves reported below show averages over repeated-sampling cross-validation, using 20 random splits of the data into training and test sets. Error bars reflect 90% confidence intervals. In all *GaSPK* runs, the number of characteristics was set to $n_c = 10$, and we used the Eigenvectors corresponding to the $n_e = 100$ largest Eigenvalues in the sparse approximation. For all experiments the length-scale hyperparameters were heuristically set to the median distance between feature vectors, as proposed by Houlsby et al. (2012).

Figure 5 shows the training time incurred by each approach for increasing training set sizes. Training times correspond to running Algorithms 2 through 4. As expected, GP classification has the fastest training time, since there is only a single GP to be learned. While the cost of matrix inversion increases cubically with the number of distinct trade-offs, the simplicity of the GP classification model means this cost remains small relative to other operations and we see little change in the training time as we increase the number of observations. By contrast, the more sophisticated comparison methods display a clear increase in computational cost as we increase the size of the training set.

Looking at the two *GaSPK* implementations, we immediately see that using interior point optimization to perform the M-step is significantly slower than the heuristic approach. As shown, the heuristic version of *GaSPK*'s training efficiency matches that of GP classification, thereby achieving two key goals. First, it trains significantly faster than the Hierarchical GP, Collaborative GP, or the Mixed Logit models. Second, *GaSPK*'s training times barely increase with the size of the training set. *GaSPK*'s fast training times and scalability as a function of the number of training observations follow directly from our proposed use of the Kronecker-structured covariance matrices. Once a given grid size is set for inference, new observations entail merely a modest increase in training time through additional likelihood terms. In contrast, in the computationally intensive methods we compare with, additional observations increase the size of the covariance matrices. The added (typically cubic) costs of matrix operations are the primary factor undermining scalability in the number of observations.

We now aim to establish whether *GaSPK* can yield improved accuracy over GP classification, thereby offering an advantageous addition to the set of highly scalable methods, and whether the time-saving heuristic EM algorithm is effective in practice. Figure 6 presents the proportion of correctly predicted test choices as a function of the training set sizes shown in Fig. 5. We note that in all cases, the heuristic version of the *GaSPK* algorithm shows comparable performance to the optimization-based version, motivating its use. In all future experiments, we will consider only this heuristic-based algorithm.

[6] On our hardware, we restricted overall grid sizes to $10^4 - 10^5$ points.

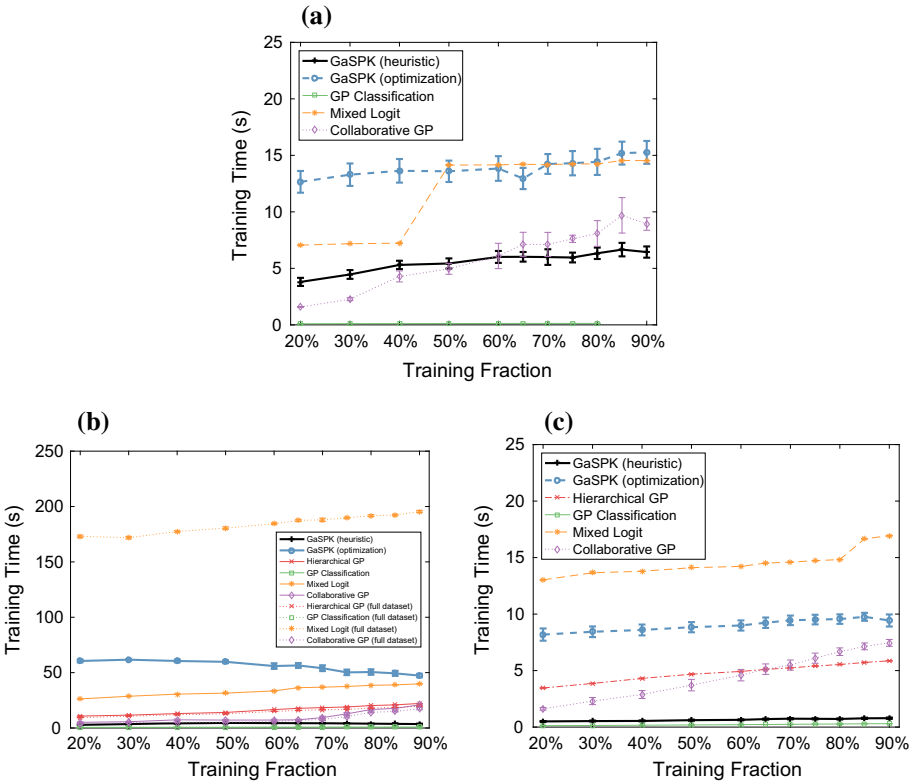

Fig. 5 Training time. Training times with respect to the fraction of data used for training the model. Error bars show 90% confidence intervals. The available hierarchical GP implementation failed to predict on the Tariffs dataset due to numerical errors. **a** Tariffs, **b** elections and **c** cars

As shown, for the Tariffs dataset, the two *GaSPK* variants exhibit the highest predictive accuracy relative to the accuracy offered by GP classification and that of the computationally slower approaches. It is useful to recall here that, similar to common real-world choice settings, the Tariffs dataset contains many alternatives but relatively few choice observations for each user (see Table 2). *GaSPK*'s focus on estimating the f^c is likely instrumental in this setting relative to other methods' focus on determining user characteristics Γ. As compared to GP classification, *GaSPK* yields comparable scalability as well as improved accuracy on the Tariffs and Elections data sets. On the Cars data set, GP classification performs well, suggesting that the problem is fairly simple and does not benefit from the additional modeling flexibility afforded by the personalized models. Here, the optimization-based *GaSPK* performs as well as the best competing algorithm, while the heuristic *GaSPK* performs slightly less well than the other GP-based methods. We hypothesize that, in this simple setting, where the modeling flexibility of the personalized models does not seems to yield significant advantages, the approximation induced by the heuristic has a more notable effect. Importantly, as compared to the computationally intensive approaches' predictive performances, when we use the heuristic EM Algorithm 4, *GaSPK*'s fast training and scalability are accompanied by predictive accuracies that are consistently good across domains, and which are not significantly worse than the most accurate and computational intensive methods. In the Elections

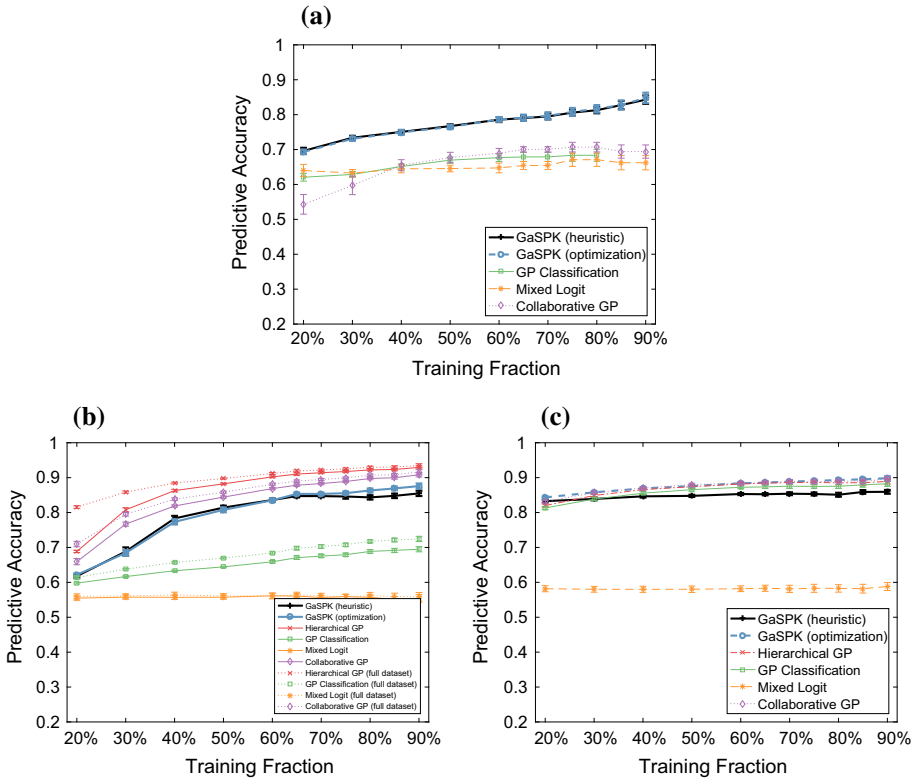

Fig. 6 Predictive accuracy. Predictive accuracy on held-out test data corresponding to the training time measurements in Fig. 5. Solid lines indicate results trained on 8 dimensions; dotted lines indicate results trained on the full dataset (20 dimensions). **a** Tariffs, **b** elections and **c** cars

dataset, we note that the *GaSPK* performs comparably with the competing methods even when the competing methods have access to the full set of 20 predictors, rather than the subset of 8 used by *GaSPK*. Indeed, in support of the hypothesis that users tend to base their choices on a smaller subset of dimensions, the comparison methods only report a modest improvement when using 20 rather than 8 predictors.

By contrast, GP classification's scalability comes at the cost of highly inconsistent predictive performance—GP classification yields particularly poor predictions on the Elections dataset, where it is unable to benefit from additional training data.

Key to our discussion is that additional training data allows non-parametric methods to capture more predictive structure in the data, as reflected by the inclining accuracy curves (see, in particular, Fig. 6b). In sharp contrast, the parametric Mixed Logit fails to benefit from additional data because its fixed set of parameters underfits larger training sets. A related effect can be observed in GP classification's performance on the Elections dataset. On this revealed preference dataset, the Hierarchical and Collaborative GP methods benefit substantially from additional training data early on in the learning curve. As shown, once a representative training sample is available, these methods are *able* to exploit more observations to capture the heterogeneity in the data. GP classification, however, benefits less from additional training

data—its single latent characteristic yields a significant speed-up in computation, but it also undermines the flexibility to capture the rich heterogeneity inherent in the Elections data.

In summary, *GaSPK* strikes a new and advantageous balance relative to existing approaches by offering a combination of the scalability of GP classification with the modeling flexibility and expressiveness of more complicated and computationally costly non-parametric GP approaches. *GaSPK* effectively adapts to the complexity in the data while scaling gracefully as more data becomes available. *GaSPK*'s scalability along with its consistently good predictive performance suggests that *GaSPK* can often be the method of choice in large-scale applications involving a large number of users and observations.

5.3 Dimensionality characteristics

GaSPK aims to produce state-of-the-art scalability in the number of observations to accommodate real-world applications with a large number of observed choices. Our solution is inspired by prior findings that human choices are determined by a small number of dimensions. *GaSPK* has thus been designed to provide superior scalability for learning and inference when trade-offs can be characterized by a small number of dimensions. Our experiments also demonstrate that dimensionality reduction in these settings incurs only a modest loss in predictive accuracy. The trade-off inherent in *GaSPK*'s ability to offer state-of-the art scalability in the number of observations and consistently good predictive performance is typical of structured GP methods: *GaSPK* is fast and highly scalable with respect to the number of observations for low-dimensional settings, while it is unsuitable in domains with high dimensions as this yields exponential growth in its grid size.

To demonstrate the implications of this trade-off, we studied the performance of *GaSPK* and the two fastest benchmarks, GP classification and Collaborative GP, on synthetic choice datasets for which we can directly control the number of users (n_U) and dimensions (d_X). Specifically, for each user, we randomly constructed a utility plane in a d_X-dimensional instance space from which we read utilities for $n_X = 15$ randomly drawn instances. These instance utilities were distorted with Gaussian noise, and then used to compute each user's choices between all $\frac{15 \cdot 14}{2} = 105$ instance pairs. Eighty percent of these choices were used to train the model while the remaining 20% were held out for evaluation. Note, that our synthetic generation procedure closely follows the key assumption underlying the Hierarchical GP and Collaborative GP, namely that users make choices based on their predetermined, latent utility functions. Our synthetic generation procedure should therefore work in the favor of these methods.

Figure 7 shows the resulting training times for several dimensionalities (panels) and levels of discretization (three *GaSPK* lines per plot). *GaSPK*'s computational costs are dominated by the fixed cost associated with a given grid size. In particular, because *GaSPK*'s grid grows exponentially in the number of dimensions, this fixed cost outgrows the variable cost of other methods as the data's dimensionality increases (see Panel (c) for 9 dimensions and 7 levels). At the same time, as shown in Fig. 7, *GaSPK*'s training curves are relatively flat; thus, it scales better for large numbers of users and choices in the consumer choice settings for which it is designed.

5.4 Sparse approximation quality

Another parameter affecting *GaSPK*'s overall computational cost is the number of Eigenvectors n_e used in the sparse approximation in Algorithm 3. In our experiments, we used $n_e = 100$ throughout, and we now illustrate that *GaSPK*'s output is relatively unaffected by

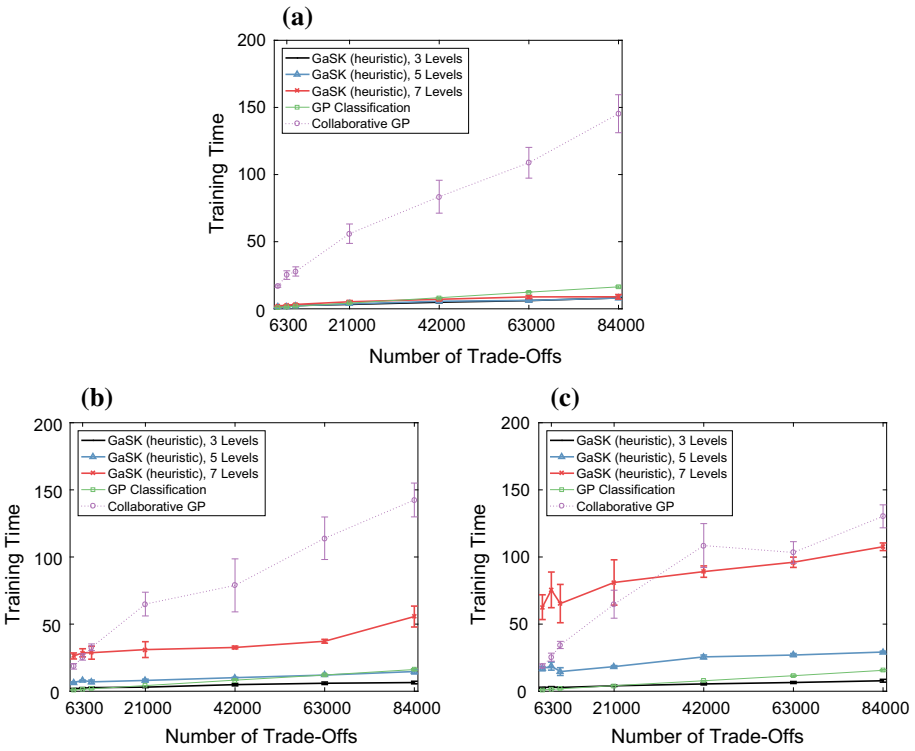

Fig. 7 Scalability in the number of dimensions. Training times for *GaSPK* at various levels of discretization, relative to GP Classification and Collaborative GP. Experiments are based of synthetic data of varying dimensionalities. Half of the dimensions are binary and the other half are continuous. Continuous dimensions were discretized to the indicated number of levels for *GaSPK* only. Error bars show 90% confidence intervals based on ten random repetitions. **a** 6 Dimensions, **b** 8 dimensions and **c** 9 dimensions

this choice as long as n_e is not excessively low. Note, that n_e has no bearing on predictive accuracy as the sparse approximation is only used in the posterior variance computation. The posterior mode, and therefore also the iterative procedure for learning user characteristics Γ (Algorithm 4), are unaffected by n_e.

Figure 8 depicts the posterior variance for the first user from a popular preference benchmark dataset, and for varying numbers of Eigenvectors. Note, that the general shape of the posterior variance is similar in all three panels, which indicates that our sparse Laplace approach delivers reasonable results starting from small n_e values. Differences between panels are primarily limited to the step from $n_e = 10$ [Panel (a)] to $n_e = 100$ [Panel (b)]. In Panel (b), the low-variance area at the center of the panel is noticeably larger than in Panel (a). Surrounding areas similarly shift to lower variances. The subsequent step to $n_e = 1000$ [Panel (c)] entails almost no further change in posterior variance. A quantitative analysis supports this interpretation: when the model was learned on a randomly selected training set of 80% of the data and evaluated on the remainder, the log predictive likelihood (two standard errors) was -0.4988 (0.0067) for $n_e = 10$, and -0.4992 (0.0065) for both $n_e = 100$ and $n_e = 1000$.

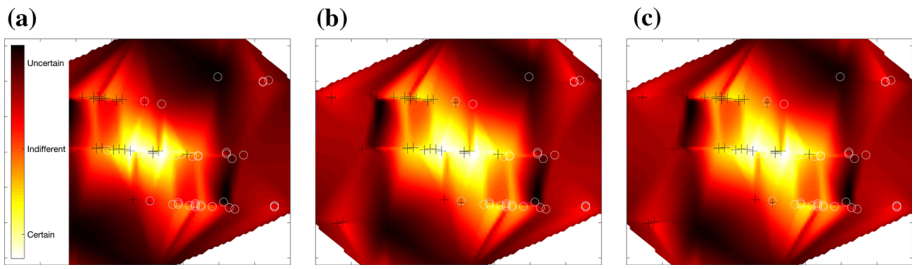

Fig. 8 Posterior variance for different numbers of Eigenvectors (n_e) in the low-rank approximation. Panels show increasingly finer posterior variance estimates for a single user from a popular preference benchmark dataset. Between **a** and **b**, the low-variance area in the center expands, and the adjacent regions shift towards lower variance, reflecting the better estimate. The addition of more Eigenvalues in (**c**) has no noticeable effect, however. **a** 10 Eigenvectors, **b** 100 Eigenvectors and **c** 100 Eigenvectors

6 Discussion and conclusions

The *GaSPK* preference model we develop here aims to offer a novel and advantageous balance between computational scalability and predictive performance targeted at preference learning, such as in consumer choice settings, involving a large number of users and alternatives. *GaSPK* provides state-of-the-art scalability when human choices are informed by a small set of dimensions, allowing it to accommodate data on a large number of users and observations. These properties are particularly critical in important emerging applications, including modeling preferences in smart electric grids and in complex Business-to-Business marketplaces, where preference models must be learned in real-time from a large number of users and observations. *GaSPK* provides principled probabilistic uncertainty estimates that are fundamental for automated, data-driven decisions.

GaSPK exploits common characteristics of consumer choice settings to yield good performance in settings where the number of users, instances, and observed choices is very large and excellent scalability is critical. Because users have been shown to approximate when evaluating alternatives (Caussade et al. 2005), *GaSPK* takes advantage of settings in which trade-offs can be captured by a small number of attributes and levels for each attribute. Our empirical evaluations demonstrate that exploiting these properties allows *GaSPK* to offer order-of-magnitude performance improvements over state-of-the-art computationally intensive approaches, making it possible to deploy preference modeling in a wide variety of contemporary, large-scale consumer choice domains. Our empirical evaluations also demonstrate that *GaSPK*'s computational benefits allow for consistently good predictive performance as compared to the scalable GP Classification, and that the scalability improvements incur only a modest reduction in predictive accuracy as compared to computationally intensive alternatives.

Figure 9 summarizes the settings under which different approaches are beneficial, and when *GaSPK* constitutes a new benchmark and an advantageous trade-off. We show that for settings with large numbers of users and choice observations where choices can be effectively characterized with few dimensions and levels, *GaSPK* offers good scalability as well as consistently good predictive performance. Thus, *GaSPK* offers a new benchmark that can often be the method of choice in these settings. In settings where both the dimensionality and the number of observations is high, GP classification provides similarly fast predictions as does *GaSPK* in lower-dimensional settings, but its predictive performance remains inconsistent due to its limited expressive power to capture complex patterns. When the number of users

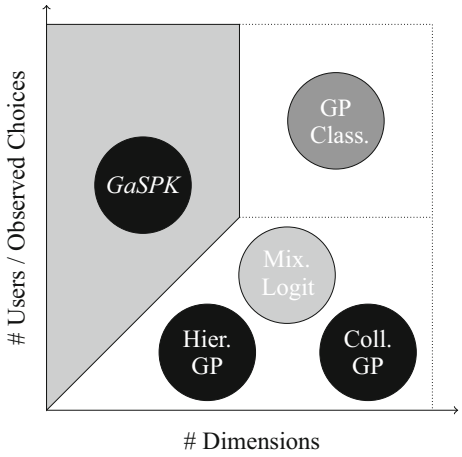

Fig. 9 Summary of empirical results. Darker colors indicate higher predictive accuracy. *GaSPK* provides high predictive accuracy and higher scalability than existing methods for consumer choice settings with few dimensions and attribute levels. In higher-dimensional settings, the Hierarchical and Collaborative GP models are more efficient, but their scalability with respect to the number of users and choices is limited. GP classification scales to high dimensions and large numbers of observations, but its predictive accuracy is inconsistent across datasets due to its limited expressive power

and observations is small, Hierarchical GP and Collaborative GP are feasible, and they offer state-the-art predictive performance.

The research we present here focuses on fast learning of probabilistic trade-off evaluations f^c that characterize different segments of the user population. We solved the related problem of learning what combination of these evaluations describes each user through a simple, yet effective iterative scheme. We find that existing alternatives to this simple iterative scheme entail significantly higher computational costs, making them impractical for the settings we consider. It would be valuable for future work to explore alternatives that learn the number of characteristics n_c from the data at a reasonable cost.

GaSPK learns from pairwise choices of the form "User u prefers alternative a to alternative b," which are objective and cognitively less demanding for humans to express, but which are also more difficult to learn from than learning from ratings. However, the natural separation between model and observations inherent in Bayesian modeling makes it possible to adapt *GaSPK* to learn from other data types, in addition to pairwise choices. In particular, Jensen and Nielsen (2014) provide likelihood models for ordinal ratings that are compatible with the framework underlying *GaSPK*, and that would allow *GaSPK* to learn from heterogeneous observations of pairwise choices and ratings simultaneously.

The contributions presented here towards efficient and scalable inference also generalize to other important classification problems such as those arising in credit scoring, quality assurance, and other impactful practical challenges. As such, *GaSPK* offers meaningful contributions to a broad range of domains, where its reliable and consistent computational and predictive performance make it suitable for supporting users' autonomous decision-making.

Acknowledgements Funding was provided by National Science Foundation (Grant No. IIS-1447721).

A Derivations used in fast inference

A.1 Probit likelihood

The Laplace mode finding procedure (Algorithm 2) requires computation of the log likelihood $\log p(C|f^c)$, and of its first two derivatives with respect to the values f_t^c of the characteristic evaluations f^c at all trade-offs t (i.e., the Jacobian $\nabla \log p(\cdot)$, and the Hessian $\nabla\nabla \log p(\cdot)$). The Probit likelihood of a single observation is given by Eq. (3) as:

$$\begin{aligned}\log p(y|f_t^c) &= \log \Phi(y \cdot f_{u,t}) \\ &= \log \Phi\left(y \cdot \left[\sum_{c=1}^{n_c} \gamma_u^c \cdot f_t^c\right]\right)\end{aligned}$$

where Φ denotes the cumulative distribution function (CDF) of the standard normal distribution. The argument to Φ is sometimes scaled by a precision factor σ^{-2}. But because our interpretation of the f^c is invariant under scaling, we can set $\sigma^{-2} = 1$ without loss of generality. The Jacobian and the Hessian of the log likelihood are given by:

$$\nabla \log p(C|f^c) = \frac{\partial \log p(y|f_t^c)}{\partial f_{t_i}^{c_1}} = \frac{y_i \gamma_u^{c_1} N(f_{t_i})}{\Phi(y f_{t_i})} \tag{10}$$

$$\nabla\nabla \log p(C|f^c) = \frac{\partial^2 \log p(y|f_t^c)}{\partial f_{t_i}^{c_1} \partial f_{t_i}^{c_2}} = \frac{-y f_{t_i} \gamma_u^{c_1} \gamma_u^{c_2} N(f_{t_i})}{\Phi(y f_{t_i})} - \frac{\gamma_u^{c_1} \gamma_u^{c_2} N^2(f_{t_i})}{\Phi^2(y f_{t_i})} \tag{11}$$

where the derivatives are with respect to the evaluations of characteristics c_1 and c_2 for trade-off t_i, and u denotes the user making choice y.

A.2 Laplace mode finding

In Laplace mode finding, we approximate the posterior $p(f|C)$ using a single Gaussian

$$q(f|C) = N(f|\hat{f}, A^{-1})$$

centered on the true mode $\hat{f} = \arg\max_f p(f|C)$, and with a precision of $A = -\nabla\nabla \log p(C|f)\big|_{f=\hat{f}}$ obtained through a second-order Taylor expansion (Rasmussen and Williams 2006). This mode is unique for the Probit because the Hessian of the log-likelihood is negative definite, and we can find it by setting the first derivative $\nabla\Psi$ of the unnormalized log posterior $\Psi = \log p(C|f) + \log p(f)$ to zero:

$$\nabla\Psi = \nabla \log p(C|f) - K^{-1} f \stackrel{!}{=} 0 \tag{12}$$

The second term in Eq. (12) results from differentiating the GP prior $p(f)$. The mode can then be found using the Newton–Raphson algorithm (Press et al. 2007) with the update step:

$$\begin{aligned}f^{new} &= f - (\nabla^2\Psi)^{-1}\nabla\Psi \\ &= (K^{-1} + W)^{-1} \underbrace{(Wf + \nabla \log p(y|f))}_{b}\end{aligned}$$

Table 3 Example choice situation used for collecting the tariffs dataset

Imagine having to choose between the following two tariffs for the household that you currently spend most of your time in. Which one would you prefer?
1. A fixed tariff with 100% renewable energy content. Your monthly cost of electricity will be
57.00$ if you consume 500 kWh,
106.00$ if you consume 1000 kWh, and
204.00$ if you consume 2000 kWh
under this tariff. You pay your monthly electricity bill at the end of each month. A 12 months notice period applies before you can cancel this tariff
2. A variable tariff with 0% renewable energy content. The cost of electricity in the first month will be
54.50$ if you consume 500 kWh,
101.00$ if you consume 1000 kWh, and
194.00$ if you consume 2000 kWh
under this tariff. After the first month, the price of electricity may go up or down in accordance with the tariff's terms (and within legal bounds). You will have to pre-pay your monthly electricity bill at the beginning of the month. You can cancel your tariff anytime.

$$= K \underbrace{(b - L(I + L^T K L)^{-1} L^T K b)}_{a}$$

The last step uses the matrix inversion lemma (Petersen and Pedersen 2008), and is valid for any symmetric decomposition $W = LL^T$.

B Tariffs dataset collection

For this study, we collected a dedicated set of pairwise choice data on Amazon Mechanical Turk (MTurk, http://www.mturk.com), a commercial crowdsourcing platform. Several scholars have studied the demographics of MTurk workers, and have proposed guidelines for assuring the quality of data collected through MTurk tasks (Paolacci et al. 2010). These studies give reason to believe that (1) MTurk data can be of equal or better quality than data selected through channels such as student surveys, (2) MTurk workers are highly diverse (increasing external validity), and (3) the unsupervised nature of MTurk tasks may reduce the risk of experimenter bias (increasing internal validity), all if proper precautions are taken against distractions and random responses.

Eighty adult American participants were invited to fill in an academic survey about their electricity tariff preferences in exchange for a payment of $0.30. All American MTurk workers could theoretically preview our survey through the MTurk platform, and 80 workers ultimately self-selected to participate. The survey consisted of three parts:

1. First, we reviewed basic electricity tariff concepts: fixed, variable, indexed tariffs, and those guaranteeing that a certain percentage of delivered electricity is produced from renewable sources.
2. Next, participants were asked to make ten choices between pairs of tariffs (see Table 3 for an example). Each pair was randomly generated from a total of 261 tariffs offered

in Austin, Texas in February 2013. Texas has one of the most advanced retail electricity markets in the United States and provides daily information on available tariffs, see http://www.powertochoose.org.

3. Finally, we asked participants to answer ten questions on their demographics and electricity consumption behavior, some of which were attention checkers for which the correct answer had to be consistent with an answer given to another question.

Participants had a maximum of thirty minutes to fill out all questions, but could submit their results before that time. Participants could also withdraw, allowing another MTurk worker to fill out the survey instead. Next to the given answers, we recorded the time between self-selecting for participation and the submission of results. In pretests among colleagues, we had established that it took a quick reader at least three minutes to process all provided information. We therefore discarded surveys submitted before that time. As a further safeguard against random answers, we asked two pairs of attention check questions in the demographics section where the answers to one question depended on the answer of the other. We also discarded surveys where at least one of the attention check pairs was answered inconsistently, leaving us with a total of 61 surveys that met our quality standards.

References

Abbasnejad, E., Sanner, S., Bonilla, E., & Poupart, P. (2013). Learning community-based preferences via Dirichlet process mixtures of Gaussian processes. In *Proceedings of the 23rd international joint conference on artificial intelligence* (pp. 1213–1219). AAAI Press.

Adomavicius, G., Gupta, A., & Zhdanov, D. (2009). Designing intelligent software agents for auctions with limited information feedback. *Information Systems Research*, *20*(4), 507.

Allenby, G. M., & Rossi, P. E. (1998). Marketing models of consumer heterogeneity. *Journal of Econometrics*, *89*(1), 57–78.

Baker, E. W. (2013). Relational model bases: A technical approach to real-time business intelligence and decision making. *Communications of the Association for Information Systems*, *33*(1), 23.

Bichler, M., Gupta, A., & Ketter, W. (2010). Designing smart markets. *Information Systems Research*, *21*(4), 688–699.

Birlutiu, A., Groot, P., & Heskes, T. (2013). Efficiently learning the preferences of people. *Machine Learning*, *90*, 1–28.

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). New York: Springer.

Bonilla, E. V., Guo, S., & Sanner, S. (2010). Gaussian process preference elicitation. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel & A. Culotta (Eds.), *Advances in neural information processing systems* (Vol. 23, pp. 262–270). MIT Press.

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, *39*, 324–345.

Caussade, S., Ortúzar, J., Rizzi, L. I., & Hensher, D. A. (2005). Assessing the influence of design dimensions on stated choice experiment estimates. *Transportation Research Part B: Methodological*, *39*(7), 621–640.

Chu, W., & Ghahramani, Z. (2005). Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on machine learning* (pp. 137–144). ACM.

Cunningham, J. P., Shenoy, K. V., & Sahani, M. (2008). Fast Gaussian process methods for point process intensity estimation. In *International conference on machine learning* (pp. 192–199). ACM.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, *39*, 1–38.

Evgeniou, T., Boussios, C., & Zacharia, G. (2005). Generalized robust conjoint estimation. *Marketing Science*, *24*(3), 415–429.

Evgeniou, T., Pontil, M., & Toubia, O. (2007). A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. *Marketing Science*, *26*(6), 805–818.

Fürnkranz, J., & Hüllermeier, E. (2011). *Preference learning*. New York: Springer.

Gilboa, I., & Schmeidler, D. (1995). Case-based decision theory. *The Quarterly Journal of Economics*, *110*(3), 605–639.

Greene, W. (2012). *Econometric analysis* (7th ed.). Upper Saddle River: Prentice Hall.

Guo, S., & Sanner, S. (2010). Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. In *International conference on artificial intelligence and statistics* (pp. 289–296).
Hensher, D. A. (2006). How do respondents process stated choice experiments? Attribute consideration under varying information load. *Journal of Applied Econometrics*, *21*(6), 861–878.
Houlsby, N., Huszar, F., Ghahramani, Z., & Hernández-Lobato, J. M. (2012). Collaborative Gaussian processes for preference learning. In F. Pereira, C. Burges, L. Bottou & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25, pp. 2096–2104). MIT Press.
Jenks, G. F., & Caspall, F. C. (1971). Error on choroplethic maps: Definition, measurement, reduction. *Annals of the Association of American Geographers*, *61*(2), 217–244.
Jensen, B. S., & Nielsen, J. B. (2014). Pairwise judgements and absolute ratings with Gaussian process priors. Technical report IMM6151, Technical University of Denmark.
Kahlen, M., Ketter, W., & van Dalen, J. (2014). Balancing with electric vehicles: A profitable business model. In *Proceedings of the 22nd European conference on information systems* (pp. 1–16). Tel Aviv, Israel.
Kamishima, T., & Akaho, S. (2009). Efficient clustering for orders. In D.A. Zighed, S. Tsumoto, Z.W. Ras, H. Hacid (Eds.), *Mining complex data.Studies in Computational Intelligence* (Vol. 165). Berlin, Heidelberg: Springer.
Kassakian, J. G., & Schmalensee, R. (2011). The future of the electric grid: An interdisciplinary MIT study. Technical report, Massachusetts Institute of Technology. ISBN: 978-0-9828008-6-7.
Kohavi, R., Mason, L., Parekh, R., & Zheng, Z. (2004). Lessons and challenges from mining retail e-commerce data. *Machine Learning*, *57*(1–2), 83–113.
Lichtenstein, S., & Slovic, P. (2006). *The construction of preference*. Cambridge: Cambridge University Press.
MacKay, D. J. (1998). Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, *168*, 133–166.
Netzer, O., Toubia, O., Bradlow, E. T., Dahan, E., Evgeniou, T., Feinberg, F. M., et al. (2008). Beyond conjoint analysis: Advances in preference measurement. *Marketing Letters*, *19*(3), 337–354.
Paolacci, G., Chandler, J., & Ipeirotis, P. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, *5*(5), 411–419.
Petersen, K. B., & Pedersen, M. S. (2008). *The matrix cookbook*. Technical report, Technical University of Denmark.
Peters, M., Ketter, W., Saar-Tsechansky, M., & Collins, J. E. (2013). A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Machine Learning*, *92*, 5–39.
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge: Cambridge University Press.
Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, *6*, 1939–1959.
Rasmussen, C. E., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge: MIT Press.
Saar-Tsechansky, M., & Provost, F. (2004). Active sampling for class probability estimation and ranking. *Machine Learning*, *54*(2), 153–178.
Saatci, Y. (2011). Scalable inference for structured Gaussian process models. Ph.D. thesis, University of Cambridge.
Snelson, E., Rasmussen, C. E., & Ghahramani, Z. (2004). Warped Gaussian processes. *Advances in Neural Information Processing Systems*, *16*, 337–344.
Stachniss, C., Plagemann, C., & Lilienthal, A. J. (2009). Learning gas distribution models using sparse Gaussian process mixtures. *Autonomous Robots*, *26*(2–3), 187–202.
Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, *34*(4), 273.
Train, K. (2003). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.
Tversky, A., & Simonson, I. (1993). Context-dependent preferences. *Management Science*, *39*(10), 1179–1189.
Valogianni, K., Ketter, W., Collins, J., & Zhdanov, D. (2014). Effective management of electric vehicle storage using smart charging. In *Proceedings of 28th AAAI conference on artificial intelligence*.
Vanhatalo, J., Pietiläinen, V., & Vehtari, A. (2010). Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine*, *29*(15), 1580–1607.
Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., & Vehtari, A. (2013). GPstuff: Bayesian modeling with Gaussian processes. *The Journal of Machine Learning Research*, *14*(1), 1175–1179.
Watson, R. T., Boudreau, M. C., & Chen, A. J. (2010). Information systems and environmentally sustainable development: Energy Informatics and new directions for the IS community. *Management Information Systems Quarterly*, *34*(1), 23.
Widergren, S. E., Roop, J. M., Guttromson, R. T., & Huang, Z. (2004). Simulating the dynamic coupling of market and physical system operations. In *IEEE power engineering society general meeting* (pp. 748–753). IEEE.