International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:11, 2020

Fast Approximate Bayesian Contextual Cold Start Learning (FAB-COST)

Jack R. McKenzie, Peter A. Appleby, Thomas House, Neil Walton

Abstract—Cold-start is a notoriously difficult problem which can occur in recommendation systems, and arises when there is insufficient information to draw inferences for users or items. To address this challenge, a contextual bandit algorithm – the Fast Approximate Bayesian Contextual Cold Start Learning algorithm (FAB-COST) – is proposed, which is designed to provide improved accuracy compared to the traditionally used Laplace approximation in the logistic contextual bandit, while controlling both algorithmic complexity and computational cost. To this end, FAB-COST uses a combination of two moment projection variational methods: Expectation Propagation (EP), which performs well at the cold start, but becomes slow as the amount of data increases; and Assumed Density Filtering (ADF), which has slower growth of computational cost with data size but requires more data to obtain an acceptable level of accuracy. By switching from EP to ADF when the dataset becomes large, it is able to exploit their complementary strengths. The empirical justification for FAB-COST is presented, and systematically compared to other approaches on simulated data. In a benchmark against the Laplace approximation on real data consisting of over $670,000$ impressions from autotrader.co.uk, FAB-COST demonstrates at one point increase of over 16% in user clicks. On the basis of these results, it is argued that FAB-COST is likely to be an attractive approach to cold-start recommendation systems in a variety of contexts.

Keywords—Cold-start, expectation propagation, multi-armed bandits, Thompson sampling, variational inference.

I. Introduction

WHEN making recommendations to website users, it is important to learn as efficiently as possible which content is most appropriate to display, including the 'cold-start' case when there is little or no prior history of the user and/or the content. Content recommendation systems and online advertising are examples contexts, in which there is an intrinsic trade-off between *exploiting* current knowledge by e.g. displaying adverts believed most likely to be clicked on, and *exploring* other content that might have higher rewards by e.g. displaying adverts which there is currently little information about.

Multi-armed bandits (MABs) are a class of algorithm which aim to balance the exploration-exploitation dilemma present whenever an intelligent system must make decisions in an uncertain environment [1]. For a recent and detailed overview on Multi-armed bandits see [2]. The most effective and conceptually simple MAB algorithm is known as Thompson Sampling [3], which uses sampling from a posterior distribution obtained through Bayesian inference. While Thompson's original (non-contextual) model for such inference has the benefit of being analytically tractable, it has the drawback of each action being assumed independent; when working in a large action space, it will be much more efficient to share information among similar content, which is the main motivation behind the contextual bandit. Chapelle et al. [4] propose such a contextual bandit based on Bayesian logistic regression, which is not in general analytically tractable, leading to the authors' use of the Laplace approximation.

Making use of the Bernstein-von Mises and central limit theorems [5], the Laplace approximation will have errors that are asymptotically $\mathrm{O}(T^{-1})$, where T is number of impressions observed. For extremely large datasets, these errors will therefore become negligible, however they may be very large early on in the learning process, and as is shown, these errors may be large enough to have significant practical consequences even after many thousands of observations.

The accuracy of the inference procedure is improved upon by using a combination of Expectation Propagation (EP), which was contemporaneously devised by [6] and [7], and Assumed density filtering (ADF) as presented by [8]. These are both moment projection variational inference methods, meaning that they work by iteratively projecting the intractable posterior distribution onto a tractable one (usually belonging to the exponential family) via minimisation of the forward Kullback-Leibler divergence. ADF is a one-pass, online method, and observations are processed one-by-one, updating the posterior distribution which is then approximated before processing the next observation. EP – which is an extension to ADF – iteratively refines the approximation by making additional passes through the dataset giving much better accuracy, but at the same time incurring a greater computational cost.

Another class of methods considered are Markov chain Monte Carlo (MCMC), which are capable of generating arbitrarily accurate representations of the Bayesian posterior. These are useful to provide a 'ground truth' for comparison of approximate methods in a study such as this, however they are not suitable for online use. This is because MCMC requires large amounts of computational effort, and also typically involves algorithmic parameters that currently cannot be tuned automatically, but rather need to be adjusted until convergence can be diagnosed [9].

The paper is structured as follows. Section II provides details of the multi-armed bandit approach to recommender systems, and Section III provides details of the inferential systems used in Bayesian online learning. In Section IV the

Jack R. McKenzie, is a PhD student in Applied Mathematics at The University of Manchester, Manchester, UK (Corresponding author, e-mail: jack.mckenzie@manchester.ac.uk).
Neil Walton and Thomas House are both Readers in the School of Mathematics at The University of Manchester, Manchester, UK.
Peter A. Appleby is Head of Data Science at Auto Trader, Manchester, UK.

FAB-COST algorithm is introduced, which is systematically compared to Laplace, EP, ADF and MCMC inference procedures, showing an attractive balance of accuracy and computational effort. It is then demonstrated that the increased accuracy of the posterior leads to better performance when used in the logistic bandit setting and results in more clicks generated when used in an online advertising scenario on real data from autotrader.co.uk. Finally the details are concluded on in Section V.

The code used to generate the results in this paper are available on GitHub at https://github.com/JackMack21/FAB-COST.

II. Bandit Algorithms

A. Notation and General Setup

Here and throughout, the shorthand $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the Gaussian distribution, where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the *mean vector* of first moments and $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is the *covariance matrix* of centred second moments. $\mathbb{E}_{p(x)}[\cdot]$ represents an expectation over the probability distribution $p(x)$.

Steps in the bandit algorithm are called *iterations*, and these are indexed by integers $i = 1, \dots, T$. T is the total iterations over the learning process, and $\tau \leq T$ refers to a current, but not necessarily final, iteration. At each iteration, the bandit algorithm – also referred to as the learner, and which is assumed to serve adverts for expositional simplicity – selects an advert from the set of eligible adverts $\mathcal{A}^i$, with cardinality $|\mathcal{A}^i| = K$. This set is indexed by $j = 1, \dots, K$. Each advert has D features, examples of which in the context of automobile sales being the colour, model and age of the car advertised. As such, the options available can be represented as a matrix $\mathbf{A}^i \in \mathbb{R}^{K\times D}$, with the j-th row, $\mathbf{a}_j^\top$, having elements corresponding to the features of an eligible advert.

The observed reward of the advert selected by the learner at iteration i is the binary outcome of a non-click/click, $y_i \in \{0, 1\}$. Each y is treated as a random variable, the expected value for which at the i-th iteration corresponds to the selected advert's click-through-rate (CTR).

$\mathbf{X} = [\mathbf{x_1}, \dots, \mathbf{x_T}]^\top$ is called the *design matrix* for the entire learning process. Each row $\mathbf{x}_i^\top \in \mathbb{R}^D$ represents the features of the advert displayed at the i-th iteration. For some algorithms the matrix $\mathbf{X}^\tau = [\mathbf{x_1}, \dots, \mathbf{x}_\tau]^\top$ will be used; this contains the information about the history of adverts chosen up to the current iteration τ. The history of observations up to iteration τ is represented by the vector $\mathbf{y}_\tau \in \mathbb{R}^\tau$.

$\boldsymbol{\theta} \in \Theta$ is the vector of parameters that is to be learnt. In the non-contextual case, each advert has a *local* parameter θ_j that is learnt and so $\Theta = \mathbb{R}^K$. In the contextual case $\Theta = \mathbb{R}^D$ and $\boldsymbol{\theta}$ is a *global* parameter vector which shares information between adverts.

B. Thompson Sampling

Thompson sampling [3] is a Bayesian approach to MABs. Its use requires the quantification of belief about the CTR for each advert via a posterior probability distribution. The adverts shown and the binary reward of click/no-click observed up to the current time τ are denoted via the vectors $\mathbf{a}_\tau = [a_1, \dots, a_\tau]^\top$ and $\mathbf{y}_\tau = [y_1, \dots, y_\tau]^\top$ respectively. As data arrives, which is comprised of the tuple (a_i, y_i), the latent parameter $\boldsymbol{\theta}$ can be learnt via Bayes' rule

$$p(\boldsymbol{\theta}|\mathbf{y}_\tau, \mathbf{a}_\tau) \propto \prod_{i=1}^{\tau} p(y_i|a_i, \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

Once a posterior distribution has been calculated for each advert, an advert should be chosen which maximises the expected CTR, where the expected CTR is calculated as

$$\mathbb{E}[y_{\tau+1}|a_{\tau+1}, \mathbf{a}_\tau, \mathbf{y}_\tau] = \int \mathbb{E}[y_{\tau+1}|\mathbf{y}_\tau, a_{\tau+1}, \mathbf{a}_\tau, \boldsymbol{\theta}] p(\boldsymbol{\theta}|\mathbf{y}_\tau, a_{\tau+1}, \mathbf{a}_\tau) \mathrm{d}\boldsymbol{\theta}. \quad (1)$$

In Thompson sampling it is not necessary to evaluate this integral, but rather a sample is generated from each advert's CTR posterior, and the advert which corresponds to the maximum sample generated is displayed. Although the integral (1) could be evaluated explicitly, displaying content that simply maximises it would involve constantly exploiting existing knowledge and would never explore to learn. The sampling-based approach, however, allows exploration and exploitation to be traded off

A well studied example of Thompson Sampling is the non-contextual Beta-Bernoulli bandit. In this simple case, each advert is assumed to have an independent Bernoulli likelihood. This has a conjugate prior, the Beta distribution, meaning that Bayesian inference can be performed analytically to give a closed form solution. When working in a very large action space (i.e. with many adverts) it is, however, much more efficient to share information among similar adverts rather than assuming independence, which is the main motivation behind the contextual bandit.

C. The Contextual Bandit

Instead of learning an independent posterior distribution for each advert, the contextual bandit instead learns a *global* posterior $p(\boldsymbol{\theta}|\mathbf{X})$ where $\boldsymbol{\theta} \in \mathbb{R}^D$, with D representing the number of covariates.

As discussed above, at the current iteration τ, the learner is presented with the features of the adverts available, stored in the matrix $\mathbf{A}^\tau$. The learner then chooses an advert from $\mathcal{A}^\tau$ which is expected to have the highest CTR when combining the context received with the sample generated from the posterior

$$a_\tau = \mathrm{argmax}(\mathbf{A}^i\boldsymbol{\theta}), \quad (2)$$

i.e. if the j-th row of $\mathbf{A}^i\boldsymbol{\theta}$ as in (2) is the maximum, advert a_j is shown. After observing the reward y_i, the context of the advert chosen a_j is then added to the history of chosen adverts (corresponding to the i^{th} row of $\mathbf{X}$), and the posterior is updated either in batch using $\mathbf{X}^\tau$ or online using the chosen adverts covariates which is now denoted $\mathbf{x}_\tau$.

The global parameter vector $\boldsymbol{\theta}$ therefore acts as a projection of the features onto the real numbers. Since the outcome is a binary reward $y_i \in \{0, 1\}$, an approach is needed that relates a continuous projection to such an outcome. Bayesian logistic regression fulfils this requirement, which is the next point of discussion.

D. Bayesian Logistic Regression

Logistic regression has proven to be a very popular and effective two class 'soft' classification method [10], [11]. The goal of logistic regression is to find the best fitting model to describe the relationship thus far observed between the binary response variables $\mathbf{y} \in \mathbb{R}^\tau$, and the design matrix $\mathbf{X} \in \mathbb{R}^{\tau \times D}$. Although in statistics literature it is referred to as logistic regression, by virtue of modelling a binary outcome it is also a method for classification [12]. While there has recently been much interest in learning techniques based on e.g. neural networks, as [13] points out, interpretability and reproducibility are two very important issues that need to be addressed, and are important advantage of using well-understood statistical techniques.

Logistic regression is a likelihood-based method in which the parameter vector $\boldsymbol{\theta}$ describes the probabilistic relationship between the input vector $\mathbf{x}_i$, and a binary response $y_i \in \{0, 1\}$. Due to the response being binary, a Bernoulli model is used for probabilities:

$$\Pr(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \pi(\mathbf{x}_i, \boldsymbol{\theta})^{y_i}(1 - \pi(\mathbf{x}_i, \boldsymbol{\theta}))^{(1-y_i)}, \tag{3}$$

where $\pi(\mathbf{x}_i, \boldsymbol{\theta}) = \mathbb{E}[y_i|\mathbf{x}_i, \boldsymbol{\theta}] = p(y_i = 1|\mathbf{x}_i, \boldsymbol{\theta})$. As $\pi(\mathbf{x}_i, \boldsymbol{\theta})$ is the probability of observing a positive outcome, the inner product $\boldsymbol{\theta}^\top \mathbf{x}_i$ is mapped from the real line to the interval $[0, 1]$ via the (sigmoidal) *logistic* function, $\sigma(x) = 1/(1+\exp(-x))$,

$$\pi(\mathbf{x}_i, \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^\top \mathbf{x}_i) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^\top \mathbf{x}_i)}. \tag{4}$$

Assuming conditional independence at each iteration, the likelihood function for the current iteration is obtained as

$$p(\mathbf{y}_\tau|\mathbf{X}^\tau, \boldsymbol{\theta}) = \prod_{i=1}^{\tau} \Pr(y_i|\mathbf{x}_i, \boldsymbol{\theta}), \tag{5}$$

where the historical observations are represented as $\mathbf{y}_\tau = [y_1, \ldots, y_\tau]^\top$ and $\mathbf{X}^\tau = [\mathbf{x}_1, \ldots, \mathbf{x}_\tau]^\top$. Equations (3), (4) and (5) between them define logistic regression.

To select adverts via Thompson sampling, $\boldsymbol{\theta}$ must be estimated, including an appropriate quantification of uncertainty, which motivates the use of a Bayesian treatment of logistic regression. Here, it is assumed that there is a distribution over $\boldsymbol{\theta}$ that is sequentially learnt as data arrives via Bayes' rule:

$$p(\boldsymbol{\theta}|\mathbf{y}_\tau, \mathbf{X}^\tau) \propto p(\mathbf{y}_\tau|\mathbf{X}^\tau, \boldsymbol{\theta})p(\boldsymbol{\theta}), \tag{6}$$

where; $p(\boldsymbol{\theta})$ is a probability density function representing the *prior* belief about the parameters, $p(\boldsymbol{\theta}|\mathbf{y}_\tau, \mathbf{X}^\tau)$ is a probability density function representing the *posterior* beliefs about the parameters, and all other quantities are as defined above. Since $\boldsymbol{\theta}$ is passed through a non-linear mapping, inference is not straightforward; more explicitly, the logistic likelihood function does not permit a conjugate prior. This leads onto the next section which explains methods for dealing with such situations.

III. Inferential Methodology

Suppose that one is trying to solve (6) for the posterior distribution – dependence on $\mathbf{y}_\tau, \mathbf{X}^\tau$ is suppressed the exact distribution is denoted $p(\boldsymbol{\theta})$. When there is a conjugate prior, the constant of proportionality in (6) can be calculated exactly, but otherwise it must be approximated.

Markov chain Monte Carlo (MCMC) methods are very popular in Bayesian statistics; and provided there is sufficient computational resources, they guarantee an arbitrarily accurate approximation to the posterior distribution. The need for large computational resources can, however, be problematic, especially when working at scale. An overview of MCMC is provided by [9].

Other approaches that will be used and detail below make a Gaussian approximation, which is justified via the Bernstein Von Mises Theorem. This states that the posterior converges to a Gaussian asymptotically [14], however different methods will achieve different accuracies at a given amount of data.

A. The Laplace Approximation

The Laplace approximation is a very popular inference method in Bayesian statistics. Its popularity is due to its simplicity: given a target distribution $p(\boldsymbol{\theta}) = \exp(-\xi(\boldsymbol{\theta}))$, which in this case will be the posterior distribution, a tractable Gaussian approximation $q(\boldsymbol{\theta})$ is made, centred at the mode of the original target density with variance equal to the curvature of the negative log-target. Explicitly, let

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \xi(\boldsymbol{\theta}), \qquad \boldsymbol{\Lambda}^* = \left.\frac{\partial^2 \xi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}=\boldsymbol{\mu}^*},$$

and then the two moments of the Gaussian approximation are matched to the above:

$$p(\boldsymbol{\theta}) \approx q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Lambda}^{*-1}).$$

The Laplace approximation is used as an inference procedure in the logistic contextual bandit described by [4] and very commonly when making an approximation in Bayesian GLMs. Asymptotically, the errors are expected to be $O(T^{-1})$ [15].

B. Variational Inference

Variational approximations turn what was an inference problem into one of optimisation. They work by minimising a distance measure $\mathcal{D}$ between the target distribution $p(\boldsymbol{\theta})$ and an approximation $q(\boldsymbol{\theta}) \in \mathcal{Z}$, where $\mathcal{Z}$ is the family of densities the approximation is to be restricted to, so that

$$q^*(\boldsymbol{\theta}) = \underset{q \in \mathcal{Z}}{\operatorname{argmin}}\ \mathcal{D}(q(\boldsymbol{\theta}), p(\boldsymbol{\theta}|\mathbf{x})). \tag{7}$$

The more generality there is in $\mathcal{Z}$, the more complex the optimisation procedure becomes, and in this application, the approximating family is restricted to the set of Gaussians.

The distance measure used is typically the Kullback-Leibler divergence [16], also known as the relative entropy, which is non-symmetric. Here, the *forward* KL-divergence, defined as

$$\begin{aligned}\mathrm{KL}(p(\boldsymbol{\theta}|\mathbf{x})||q(\boldsymbol{\theta})) &= \int p(\boldsymbol{\theta}|\mathbf{x}) \log\left\{\frac{p(\boldsymbol{\theta}|\mathbf{x})}{q(\boldsymbol{\theta})}\right\} d\boldsymbol{\theta} \\ &= \mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[\log\left(p(\boldsymbol{\theta}|\mathbf{x})\right) - \log\left(q(\boldsymbol{\theta})\right)\right]\end{aligned} \tag{8}$$

is used as the distance measure, which when used in the objective in (7), gives a solution known as the *moment projection*.

Note that the *reverse* KL-divergence is obtained by swapping p and q in (8) and when used as the objective in (7), the solution is known as the *information projection*. This form is not necessarily convex in $\boldsymbol{\theta}$ and can therefore yield different solutions depending on how the optimisation procedure is initialised. While the information projection is most commonly used in variational inference due to its simplicity, there are problems with reliability and accuracy, and no advantage was found by using this approach – see [12] and [17] for further discussion.

Unlike the information projection, the objective (8) is convex in $\boldsymbol{\theta}$, and will give a unique solution when minimised. This minimum corresponds to an approximation centred at the mean of the target which is why it is known as *mean seeking*. The moment projection plays an important role in asymptotic theory and as explained by [18], its minimisation has the desirable effect of minimising the expected loss.

Although it is the 'correct' KL-divergence measure to use, the forward version has the drawback that it is much harder to compute; it is intractable as it requires the calculation of the expectation over the target $p(\boldsymbol{\theta}|\mathbf{x})$. EP overcomes the intractability of the target by forming what is known as a tilted distribution; this will be discussed in the next section.

C. Exponential Families

A distribution belongs to the set of exponential families if its density can be written as

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \exp\left(\boldsymbol{\lambda}^T\phi(\boldsymbol{\theta}) - \Phi(\boldsymbol{\lambda})\right), \qquad \boldsymbol{\lambda} \in \boldsymbol{\Theta}$$
$$\Phi(\boldsymbol{\lambda}) = \log \int \exp\left(\boldsymbol{\lambda}^T\phi(\boldsymbol{\theta})\right) \mathrm{d}\boldsymbol{\theta},$$

where $\phi(\boldsymbol{\theta})$ is known as the sufficient statistics, $\boldsymbol{\lambda}$ the natural parameters and $\Phi(\boldsymbol{\lambda})$ is the log-partition function which ensures normalisation.

Assuming that the representation of the exponential family is minimal (i.e. there are no dependencies between the components of $\phi(\boldsymbol{\theta})$ and $\boldsymbol{\lambda}$), then the following properties hold:

1) Product of Exponentials: A product of densities belonging to the set exponential families is also a member of the exponential families:

$$\prod_{i=1}^{T} q(\boldsymbol{\theta}|\boldsymbol{\lambda}_i) = q\left(\boldsymbol{\theta}\Big|\sum_{i=1}^{T}\boldsymbol{\lambda}_i\right)\exp\left(\Phi\left(\sum_{i=1}^{T}\boldsymbol{\lambda}_i\right) - \sum_{i=1}^{T}\Phi(\boldsymbol{\lambda}_i)\right),$$

given that $\sum_{i=1}^{T}\boldsymbol{\lambda}_i \in \boldsymbol{\Theta}$.

2) Moments: Moments can be found by differentiating the log-partition function with respect to its natural parameters:

$$\mathbb{E}_{q(\boldsymbol{\theta})}[\phi(\boldsymbol{\theta})] = \nabla_{\boldsymbol{\lambda}}\Phi(\boldsymbol{\lambda}), \quad \mathrm{Var}_{q(\boldsymbol{\theta})}[\phi(\boldsymbol{\theta})] = \nabla^2_{\boldsymbol{\lambda}}\Phi(\boldsymbol{\lambda}).$$

3) Bijective Mapping: There is a bijective mapping from the natural parameters $\lambda(\boldsymbol{\eta})$ and the moment parameters $\boldsymbol{\eta}(\boldsymbol{\lambda})$. The log-partition function $\Phi(\boldsymbol{\lambda})$ is strictly convex and has a Legendre dual

$$\Psi(\boldsymbol{\eta}) = \mathbb{E}_{\eta(\boldsymbol{\lambda})}[\log p(\lambda(\boldsymbol{\eta})|\boldsymbol{\theta})]$$

Conversion between the natural and moments parameters is done via:

$$\boldsymbol{\eta}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}}\Phi(\boldsymbol{\lambda}), \qquad \theta(\boldsymbol{\eta}) = \nabla_{\boldsymbol{\eta}}\Psi(\boldsymbol{\eta})$$

These properties are very useful for message passing algorithms which both EP and ADF are a subset of – see [19]. Seeger [20] gives an in-depth discussion of the properties of exponential families.

IV. FAB-COST

In this section a description of the FAB-COST algorithm is given, along with a demonstration of its performance on real automotive website data (AT) taken from the autotrader.co.uk website.

A. The Logistic Contextual Bandit

Having introduced multi-armed bandits, Thompson sampling for these via Bayesian logistic regression, and multivariate normal approximations to the posterior in such regressions, the overall structure of the FAB-COST approach, as in Algorithm 1, can now be provided. It is worth mentioning that the pseudocode for the logistic regression bandit provided is the same as the linear bandit introduced by [21] and [22]; however, the linear case is conjugate and can be solved exactly, meaning that it does not require the work to update moments accurately that have been carried out.

Algorithm 1: Logistic Regression Thompson Sampling

for $i = 1 \dots T$ **do**
  1. Generate a sample from the approximated posterior:
    $\tilde{\boldsymbol{\theta}}_i \sim \mathcal{N}(\boldsymbol{\mu}_{i-1}, \boldsymbol{\Sigma}_{i-1})$
  2. Select an advert:
    $a_i = \underset{\mathrm{j}\in\mathcal{A}}{\mathrm{argmax}}(\mathbf{A}^i\tilde{\boldsymbol{\theta}}_i)$
  3. Update moments:
    $\boldsymbol{\mu}_i = \mathbb{E}[\boldsymbol{\theta}|\mathbf{X}_i, \mathbf{y}_i]$
    $\boldsymbol{\Sigma}_i = \mathbb{E}[(\boldsymbol{\theta} - \boldsymbol{\mu}_i)(\boldsymbol{\theta} - \boldsymbol{\mu}_i)^\top|\mathbf{X}_i, \mathbf{y}_i]$
end

Describing this algorithm in long form, it is initialised with a prior belief on $\boldsymbol{\theta}$. At each iteration, the learner is presented with an available set of adverts and random sample is generated from the posterior distribution (or prior at the first iteration), which corresponds to step 1 in Algorithm 1. An advert is then selected from the set via Thompson Sampling by choosing a_j which maximises the linear combination of the sample generated $\tilde{\boldsymbol{\theta}}_i$ and the eligible adverts covariates $\mathbf{A}^i$; this corresponds to step 2 in Algorithm 1 – note that the monotonicity of the logistic function means that the sampled CTR is maximised when the linear combination is maximised. This is followed by observing a binary reward of a user clicking or not clicking on the chosen advert, at which point the posterior $p(\boldsymbol{\theta})$ is updated (i.e. within the Gaussian approximation, the first and second moments are updated). This corresponds to step 3 in Algorithm 1.

B. Expectation Propagation

Expectation Propagation (EP) is an iterative approach to minimising the forward KL-divergence between the posterior that is to be approximated, and the Gaussian approximation. It was first generalised by [6] and [7], but has roots further back in Statistical Physics [23]. EP belongs to a group of message passing algorithms and works by essentially propagating the moments of an exponential family - which in the Gaussian case are the mean and variance - between the factors of the posterior. Finding the moments of the target is obviously problematic as the target is intractable - if it weren't then a closed form solution could be found analytically. EP's solution to this is to form what is known as the *tilted distribution* $t_i(\boldsymbol{\theta})$ (whose moments are much easier to find) and iteratively project the moments from this, onto the tractable approximation $q(\boldsymbol{\theta})$.

First, it is assumed that the true posterior factorises into a product of T factor terms or *sites*:

$$p(\boldsymbol{\theta}|\mathbf{x}) \propto \prod_{i=1}^{T} p_i(\boldsymbol{\theta}).$$

EP approximates each of these true sites by a Gaussian distribution $q_i(\boldsymbol{\theta})$, which in natural parameters is expressed as

$$p_i(\boldsymbol{\theta}) \approx q_i(\boldsymbol{\theta}|\boldsymbol{\lambda}_i) \propto \exp\left\{\mathbf{h}_i\mathbf{x}_i - \boldsymbol{\Lambda}_i\frac{\mathbf{x}_i^2}{2}\right\}.$$

Then due to property 1 in of exponential families in §III-C, the global approximation is expressed as

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_{i=1}^{T} q_i(\boldsymbol{\theta}|\boldsymbol{\lambda}_i),$$

and the natural parameters of our global approximation can be calculated as the product of the natural parameters of each site approximation $\mathbf{h} = \sum_{i=1}^{T}\mathbf{h}_i$, $\boldsymbol{\Lambda} = \sum_{i=1}^{T}\boldsymbol{\Lambda}_i$. It is this ability to simply add and subtract natural parameters of the sites that motivates the use of an exponential family approximation.

The EP algorithm sweeps through the data set, with steps described below.

1) The Tilted Distribution: At each iteration of the algorithm, the current *global approximation* $q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_{i=1}^{T} q_i(\boldsymbol{\theta}|\boldsymbol{\lambda}_i)$ is augmented by replacing one of the sites with a true site $p_i(\boldsymbol{\theta})$. This can be thought of in two steps: firstly the *cavity distribution* is defined as

$$q^{\setminus i}(\boldsymbol{\theta}|\boldsymbol{\lambda}^{\setminus i}) = \prod_{j\neq i} q_j(\boldsymbol{\theta}|\boldsymbol{\lambda}_j),$$

which is the global approximation with a site removed, then the *tilted distribution* is defined as

$$t_i(\boldsymbol{\theta}) \propto p_i(\boldsymbol{\theta}) \prod_{j\neq i} q_j(\boldsymbol{\theta}|\boldsymbol{\lambda}_j),$$

which is the cavity with its 'hole' filled in with a true site.

2) The Moment Projection: EP proceeds to iteratively project the tilted onto the global approximation

$$q^*(\boldsymbol{\theta}) = \underset{\mathrm{q}\in\mathcal{Z}}{\operatorname{argmin}}\ \mathrm{KL}(t_i(\boldsymbol{\theta})||q(\boldsymbol{\theta}|\boldsymbol{\lambda})). \tag{9}$$

Equivalently, the first two moments of the tilted can be computed $\mathbb{E}_{t_i}[\phi(\boldsymbol{\theta})] = [\boldsymbol{\mu}, \boldsymbol{\Sigma}]^\top$ and the moments of the global approximation are equated to these: $q^*(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The moments of the local site approximation $q_i(\boldsymbol{\theta}_i)$ must then be updated which is generally done by division

$$q_i^*(\boldsymbol{\theta}|\boldsymbol{\lambda}_i) = \frac{q^*(\boldsymbol{\theta}|\boldsymbol{\lambda})}{q^{\setminus i}(\boldsymbol{\theta}|\boldsymbol{\lambda}^{\setminus i})}.$$

Using property 1 in of exponential families in §III-C, this results in a simple subtraction of the natural parameters.

3) Mapping from Natural to Moment Parameters: As the objective in (9) is found via moment matching, yet the Gaussian approximation is parameterised by its natural parameters, the two different parameterisations but be alternated between at each iteration. Due to property 3 of exponential families in §III-C, there is a bijective mapping between the two. In the Gaussian case these mappings are:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}, \qquad \boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\mathbf{h}, \tag{10}$$

where the moment parameters – the mean and the variance – $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively, and the natural parameters – the shift and the precision – are given by $\mathbf{h}$ and $\boldsymbol{\Lambda}$ respectively.

4) Overall Structure: As shown in Algorithm 2, EP makes multiple sweeps through the dataset, iteratively forming the tilted distribution, matching the moments of the global approximation, and finally updating the site parameters. This is done until convergence which has been shown to usually be after 3-4 sweeps (which coincides with the experiments carried out in this work) although a stopping rule described by [24] can also be used.

Algorithm 2: The Expectation Propagation algorithm

while *not converged* **do**
  for $i = 1 \dots T$ **do**
    1. Form the tilted distribution:
      $t_i(\boldsymbol{\theta}) = \frac{1}{Z}p_i(\boldsymbol{\theta})q^{\setminus i}(\boldsymbol{\theta}|\boldsymbol{\lambda})$.
    2. Minimise the forward KL-divergence between the tilted distribution and the global approximation:
      $q^*(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \underset{\mathrm{q}\in\mathcal{Z}}{\operatorname{argmin}}\ \mathrm{KL}(t_i(\boldsymbol{\theta})||q(\boldsymbol{\theta}|\boldsymbol{\lambda}))$.
    3. Update the approximating site:
      $q_i^*(\boldsymbol{\theta}|\boldsymbol{\lambda}_i) = \frac{q^*(\boldsymbol{\theta}|\boldsymbol{\lambda})}{q^{\setminus i}(\boldsymbol{\theta}|\boldsymbol{\lambda}^{\setminus i})}$.
  end
end

5) Error Analysis: While EP has shown tremendous empirical success in many applications – particularly Bayesian General Linear Models and Gaussian Process regression [25] – the theoretical understanding of it is less comprehensive than for other approaches. There has, however, been important progress due to Dehaene and Barthelmé who show that EP

behaves like iterations of Newtons algorithm for finding the mode of a function [26]. Under what they describe in later work as 'unrealistic assumptions on the model', Dehaene and Barthelmé also showed that EP can converge at a rate of $O(T^{-2})$ [27]. The experiments carried out in this work suggest that while such a rate is indeed likely to be optimistic in more realistic settings, EP is often significantly more accurate than the Laplace approximation.

C. Assumed Density Filtering

1) Algorithm: EP is a batch method, requiring multiple sweeps through a complete dataset, potentially limiting its usefulness online. Assumed Density filtering (ADF) is a sequential inference method and can be used to work online. As with EP, ADF iteratively minimises the forward KL-divergence between the tilted distribution and the approximation, the difference being how this tilted distribution is formed.

In the setting of data arriving sequentially, the posterior distribution at data point τ is given as

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{\prod_{i=1}^{\tau} p(\mathbf{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int \prod_{i=1}^{\tau} p(\mathbf{x}_i|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})\mathrm{d}\boldsymbol{\vartheta}}.$$

ADF takes $q(\boldsymbol{\theta})$ as the prior on $\boldsymbol{\theta}$ and iterates through the data, incorporating each point into the approximate posterior. The conditional distribution of $\boldsymbol{\theta}$ given the first τ data points can be expressed as

$$p(\boldsymbol{\theta}|x_{1:\tau}) = \frac{p(\mathbf{x}_\tau|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}_{1:\tau-1})}{\int p(\mathbf{x}_\tau|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}|\mathbf{x}_{1:\tau-1})\mathrm{d}\boldsymbol{\vartheta}}. \tag{11}$$

Then assuming that at the previous iteration the approximation $q^{(\tau-1)}(\boldsymbol{\theta})$ is made to the true posterior $p(\boldsymbol{\theta}|\mathbf{x}_{1:\tau-1})$, (11) can be rewritten to give a new *tilted distribution*

$$t_\tau(\boldsymbol{\theta}|\mathbf{x}_\tau) = \frac{p(\mathbf{x}_\tau|\boldsymbol{\theta})q^{(\tau-1)}(\boldsymbol{\theta})}{\int p(\mathbf{x}_\tau|\boldsymbol{\vartheta})q^{(\tau-1)}(\boldsymbol{\vartheta})\mathrm{d}\boldsymbol{\vartheta}}.$$

Due to there being neither a site update nor a cavity there is no need to map between the natural and moment parameters and therefore no need to perform the expensive matrix inversion seen in (10).

2) Error analysis: Fig. 1 shows that ADF makes a very poor approximation to the posterior unless it is trained on sufficient data. EP on the other hand makes an accurate approximation even on the first $1,000$ data points.

Theoretical results given by [8] calculate the asymptotic convergence of ADF by showing that the inverse of the covariance matrix approaches the fisher information matrix as $T \to \infty$. By assuming that the difference between the change in the covariance matrix between time points is negligible, he models its evolution as a matrix differential equation to give asymptotic accuracy of $O(T^{-1})$ for the mean, although no convergence rate is given for the variance.

These empirical and theoretical considerations align with the intuition that multiple sweeps through a dataset as in EP are expected to mitigate against inaccuracies from sites that lead to tilted distributions that are poorly approximated better than in a one-sweep algorithm such as ADF. Discussion by [25] is relevant in this context.

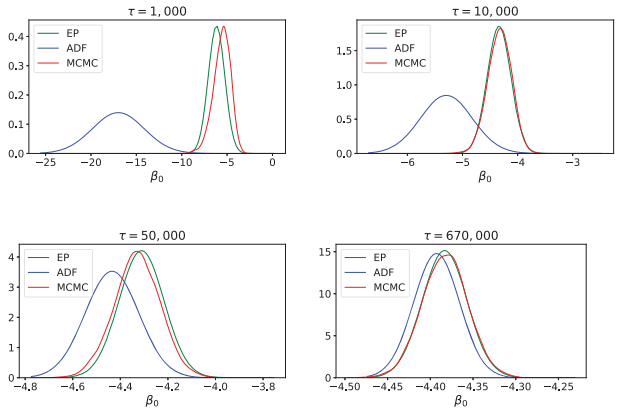

Fig. 1 Convergence of both EP and ADF on AT dataset. ADF is slow to converge, motivating its combination it with EP, which is typically much more accurate

D. Gaussian Filtering

1) Moment Matching: As mentioned earlier, (9), which takes the form:

$$q^*(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \underset{\mathrm{q}\in\mathcal{Z}}{\operatorname{argmin}}\ \mathrm{KL}(t_i(\boldsymbol{\theta})||q(\boldsymbol{\theta}|\boldsymbol{\lambda})),$$

can be solved at each iteration via moment matching

$$\mathbb{E}_q\left[\phi(\boldsymbol{\theta})\right] = \mathbb{E}_{t_i}\left[\phi(\boldsymbol{\theta})\right].$$

Due to the approximation being restricted to the set of Gaussian distributions, this requires only the first two moments of the tilted distribution to be found: the mean $\boldsymbol{\mu}$ and the variance $\boldsymbol{\Sigma}$.

Note that if all moments associated with $q(\boldsymbol{\theta})$ existed and were equal to those associated with $t(\boldsymbol{\theta})$, then the KL-divergence would be zero and the approximation would be exact. As the approximation used is Gaussian, it is only characterised by the first two moments; any difference in higher moments (e.g. skew, kurtosis) between the two is will lead to errors since the Gaussian lacks the flexibility to capture these.

In the logistic regression case, the tilted function is

$$t_i(\boldsymbol{\theta}) = \frac{1}{\tilde{Z}(\tilde{\boldsymbol{\lambda}}_i)} p_i(\boldsymbol{\theta}) \prod_{j\neq i} q_j(\boldsymbol{\theta}|\boldsymbol{\lambda}_j),$$

where the true site distribution functions are $p_i(\boldsymbol{\theta}) = \sigma(y_i\boldsymbol{\theta}^\top\mathbf{x}_i)$ and the approximating site functions are $q_i(\boldsymbol{\theta}|\boldsymbol{\lambda}_i) = \mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$. The normalisation constant is

$$\tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) = \int p_i(\boldsymbol{\theta}) \prod_{j\neq i} q_j(\boldsymbol{\theta}|\boldsymbol{\lambda}_j)\mathrm{d}\boldsymbol{\theta}.$$

Because of product of exponentials property (see §III-C1), the cavity can be written as $\prod_{j\neq i} q_j(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu}^{\setminus i},\boldsymbol{\Sigma}^{\setminus i})$ where the complement notation $\boldsymbol{\mu}^{\setminus i} := (\boldsymbol{\mu}_j)_{j\neq i}$ has been used.

Using the moments property from §III-C2, the moments of the tilted can be found via differentiation of the log partition function as

$$\mathbb{E}_{t_i}\left[\phi(\boldsymbol{\theta})\right] = \nabla_{\tilde{\boldsymbol{\lambda}}_i} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}_i).$$

The remaining problem is that the natural parameters $\tilde{\boldsymbol{\lambda}}_i$ are not known; if they were the moments could be found by a simple bijective mapping. Fortunately there is a recursive formulation derived by [28], which enables the moments of the tilted to be calculated as

$$\mathbb{E}_{t_i}\left[\phi(\boldsymbol{\theta})\right] = \mathbb{E}_{q^{\setminus i}}\left[\phi(\boldsymbol{\theta})\right] + \nabla_{\boldsymbol{\lambda}^{\setminus i}} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}), \tag{12}$$

meaning that the cavity natural parameters $\boldsymbol{\lambda}^{\setminus i}$ are needed instead of the tilted natural parameters $\tilde{\boldsymbol{\lambda}}_i$.

After some calculations, (12) can be used to give an iterative update formula for the first and second Gaussian moments of the global approximation:

$$\boldsymbol{\mu} = \boldsymbol{\mu}^{\setminus i} + \boldsymbol{\Sigma}^{\setminus i}\boldsymbol{\alpha}_i, \quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\setminus i} - \boldsymbol{\Sigma}^{\setminus i}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top - 2\mathbf{B}_i\right)\boldsymbol{\Sigma}^{\setminus i}, \tag{13}$$

where

$$\boldsymbol{\alpha}_i = \nabla_{\boldsymbol{\mu}_i} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) \in \mathbb{R}^D \quad \text{and} \quad \mathbf{B}_i = \nabla_{\boldsymbol{\Sigma}_i} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) \in \mathbb{R}^{D\times D}. \tag{14}$$

2) Linear Subspace Property: The normalising constant to the tilted distribution $\tilde{Z}(\tilde{\boldsymbol{\lambda}}_i)$ is, in general, intractable – its moments can be evaluated via numerical quadrature or MCMC, however in this work an approximation described by MacKay [29] is used. In MacKay's estimation framework, known as the 'evidence framework' or 'moderated output', the normalising constant is approximated as

$$\tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) = \int \sigma(y_i\boldsymbol{\theta}^\top\mathbf{x}_i)\mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu}^{\setminus i},\boldsymbol{\Sigma}^{\setminus i})\mathrm{d}\boldsymbol{\theta} \approx \sigma(\kappa(s_i^2)a_i), \tag{15}$$

where

$$\kappa(s_i^2) = \left(1+(\pi s_i^2/8)\right)^{-1/2}, \quad s_i^2 = \mathbf{x}_i^\top\boldsymbol{\Sigma}^{\setminus i}\mathbf{x}_i, \quad a_i = \mathbf{x}_i^\top\boldsymbol{\mu}^{\setminus i}.$$

Using the moderated output given in (15), $\boldsymbol{\alpha}_i$ and $\mathbf{B}_i$ are expressed as

$$\begin{aligned}\boldsymbol{\alpha}_i &= \nabla_{\boldsymbol{\mu}^{\setminus i}} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) = \boldsymbol{\rho}_i \frac{\sigma'(\boldsymbol{\rho}_i\boldsymbol{\mu}^{\setminus i})}{\sigma(\boldsymbol{\rho}_i\boldsymbol{\mu}^{\setminus i})},\\ \mathbf{B}_i &= \nabla_{\boldsymbol{\Sigma}^{\setminus i}} \log \tilde{Z}(\tilde{\boldsymbol{\lambda}}_i) = -\frac{\pi}{16\kappa^2}\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\mu}^{\setminus i}\boldsymbol{\alpha}_i^T,\end{aligned} \tag{16}$$

where

$$\boldsymbol{\rho}_i = y_i\kappa_i(s^2)\mathbf{x}_i^T, \; \sigma(z) = \frac{1}{1+\exp(-z)}, \; \sigma'(z) = \frac{\exp(-z)}{(1+\exp(-z)}$$

3) Filtering Algorithm: Using the results (16) along with the iterative update equations (13), Gaussian approximations to the posterior in Bayesian logistic regression can be computed for both EP and ADF. These are shown in Algorithms 3 and 4 respectively.

Algorithm 3: Gaussian Expectation Propagation

Initialise the global approximation $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
while *not converged* **do**
  for $i = 1 \ldots T$ **do**
    1. Form the cavity expected moments:
      $\boldsymbol{\Sigma}^{\setminus i} = \left(\boldsymbol{\Sigma}^{-1} - (\boldsymbol{\Sigma}_i)^{-1}\right)^{-1}$
      $\boldsymbol{\mu}^{\setminus i} = \boldsymbol{\Sigma}^{\setminus i}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i\right)$
    2. Project the moments of the tilted distribution onto the global approximation:
      $\boldsymbol{\mu} = \boldsymbol{\mu}^{\setminus i} + \boldsymbol{\Sigma}^{\setminus i}\boldsymbol{\alpha}_i$
      $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\setminus i} - \boldsymbol{\Sigma}^{\setminus i}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top - 2\mathbf{B}_i\right)\boldsymbol{\Sigma}^{\setminus i}$
    3. Update the expected moments of site i:
      $\boldsymbol{\Sigma}_i = \left(\boldsymbol{\Sigma}^{-1} - (\boldsymbol{\Sigma}^{\setminus i})^{-1}\right)^{-1}$
      $\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - (\boldsymbol{\Sigma}^{\setminus i})^{-1}\boldsymbol{\mu}^{\setminus i}\right)$
  end
end

Algorithm 4: Gaussian Density Filtering

Initialise the prior distribution $q_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
for $i = 1 \ldots T$ **do**
  1. The cavity distribution is simply the approximation made at the previous iteration:
    $\boldsymbol{\mu}^{\setminus i} = \boldsymbol{\mu}_{i-1}$
    $\boldsymbol{\Sigma}^{\setminus i} = \boldsymbol{\Sigma}_{i-1}$
  2. Project the moments of the tilted distribution onto the global approximation:
    $\boldsymbol{\mu} = \boldsymbol{\mu}^{\setminus i} + \boldsymbol{\Sigma}^{\setminus i}\boldsymbol{\alpha}_i$
    $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\setminus i} - \boldsymbol{\Sigma}^{\setminus i}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top - 2\mathbf{B}_i\right)\boldsymbol{\Sigma}^{\setminus i}$
end

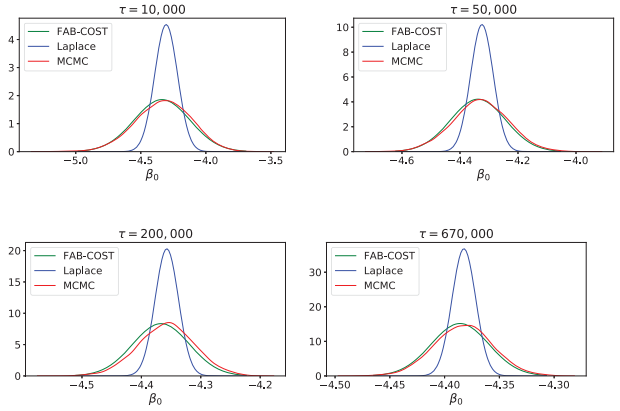

Fig. 2 FAB-COST vs Laplace. The Laplace approximation fails to capture the true variance, meaning that in the bandit setting a failure to balance the exploration-exploitation tradeoff is expected

E. Combining Methods in FAB-COST

An outline is now given of how the considerations above lead us to the FAB-COST approach to recommendation systems, which will turn out to provide improved performance on real data for well-controlled computational effort.

1) Data: Auto Trader PLC is the UK's largest digital automotive marketplace, and the 'AT' dataset was constructed from the website autotrader.co.uk. This consists of a day's 'featured listing' user click data, totalling at $T = 678,446$ impressions. After a user makes a search for a car they are

presented with a single 'featured listing' which appears at the top of the search result, meaning that there is no positional bias. The user then proceeds to either click or not click on the presented advert. Many dozens of covariates for each advert are available, which was reduced to the 15 most important using a random forest algorithm for feature selection, although other possibilities for feature selection can be used [11].

2) Computational Cost: The structure of Algorithms 3 and 4 makes clear that one iteration ADF is expected to be computationally cheaper than one iteration of EP. This is because, at each iteration, EP requires the expensive matrix inversion required to map between the natural and moment parameters. Although both matrix multiplication and matrix inversion come at cubic computational complexity $O(D^3)$, the pre-multiplication constant for ADF is much smaller. Calculations show that per site, EP's flop count of $\frac{38}{3}D^3 + O(D^2)$ is over three times greater than that of ADF's flop count of $4D^3 + O(D^2)$. Due to matrix multiplication being easily parallelised (which Python's Numpy library exploits), as well as EP requiring multiple sweeps over the data set, we recorded a 75-fold speed-up when using ADF in batch over EP.

Now consider the number of iterations involved in an online learning context. Let τ be the number of data points used to make the last posterior approximation and m is the number of data points arriving since the last posterior approximation. EP requires the entire data set up to the current iteration to make a posterior approximation leading to a computational cost of $O((\tau+m)D^3)$ compared to ADFs $O(mD^3)$. For $m \ll \tau$ this is going to result in a dramatic increase in computational cost choosing EP over ADF.

The computational cost vs accuracy trade-off is shown in the right column of Figure 3. The requirement for EP to be ran in batch is what causes the significant increase in the FLOPs use over the learning process. In these plots, an EP update is performed every $5,000$ impressions ($m = 5,000$) rather than in a true sequential manner. This batch size was chosen purely for illustrative purposes; an EP update would have to be done more frequently early on in a bandit setting. If these batch sizes decreased then the computational cost would increase as a result.

3) Accuracy: As discussed above, MCMC can be used to provide an arbitrarily accurate representation of the posterior, although the computational effort involved in doing this is prohibitive in an online context. To assess accuracy for the purposes of this study, however, the No-U-Turn sampler (NUTS) was implemented using PyMC3 [30], and takes the output of this as the ground truth of the posterior distributions we are trying to approximate. By measuring the absolute error between the moments of the ground truth and each inference procedure, an empirical estimate for the asymptotic convergence rate of each can be computed by measuring the gradient of a log-log plot (see the left column of Fig. 3).

The results show that EP gives better asymptotic accuracy to the posterior than other methods, as would be expected from the theoretical results discussed above, although for the real dataset considered, a simple power law in T is not observed. In general it is clear that the main improvements for EP and ADF

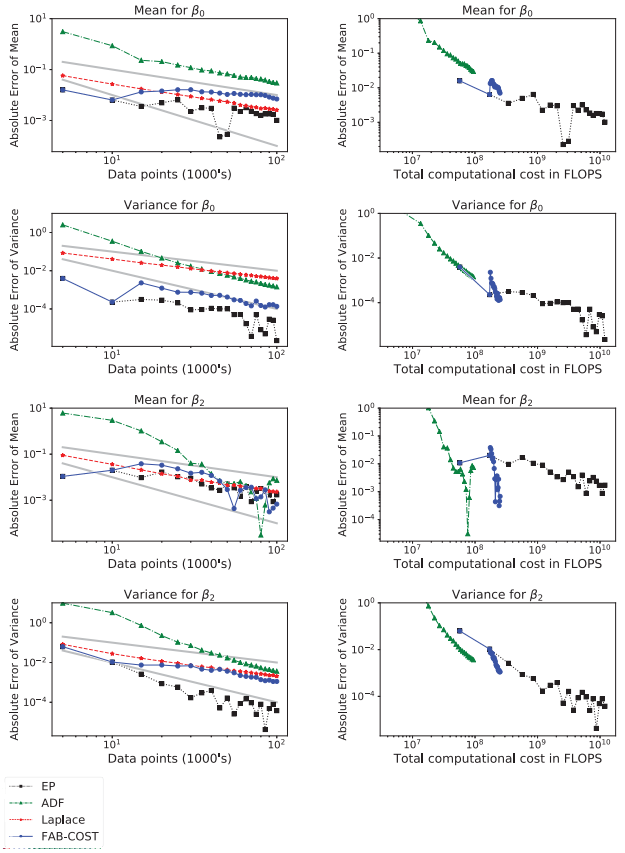

Fig. 3 The left column shows log-log plots of the mean and variance error for the Laplace approximation, ADF, EP and FAB-COST where the No U-Turn sampler was used to establish the ground truth. The grey lines show asymptotic error of $O(T^{-1})$ and $O(T^{-2})$ for reference. The right column shows the asymptotic accuracy computational cost trade-off between the methods as discussed in section IV-E2

over Laplace come from estimation of the variance, as Fig. 2 shows more explicitly. This to be particularly important for recommendation systems, where balancing the explore-exploit tradeoff is crucial.

Also as expected, once data availability becomes large, one can switch to ADF as proposed in the FAB-COST algorithm without significant loss of accuracy. Although EP can be used periodically to update the posterior approximation in the FAB-COST algorithm, only two EP updates were performed in the simulations above (at 100 and $10,000$ iterations) although the posterior accuracy would be improved with more frequent EP updates. Theses updates at 100 and $10,000$ iterations out of the $T = 670,000$ sized data set shows that if given a reasonable prior, ADF achieves good accuracy.

4) Recommendation System and Performance: Algorithm 5 shows the pseudocode for FAB-COST. This takes the logistic bandit (shown in Algorithm 1) and adds moment updating using both EP and ADF. In the experiments, two EP updates were chosen at $E = 100$ and $E = 10,000$ (although this can be performed as often as is computationally feasible) with ADF sequentially updating the posterior approximation at each iteration for the remainder of the simulation. These EP updates

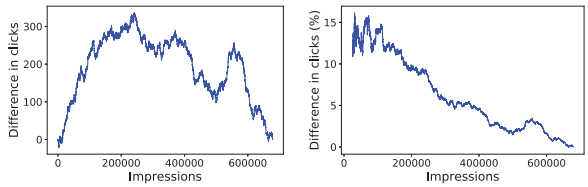

Fig. 4 The difference in cumulative clicks received from FAB-COST and the Laplace bandit. FAB-COST generates over 16.1% more clicks after around $31,000$ impressions

were chosen to show that even with few updates, FAB-COST is superior to the Laplace approximation for both posterior approximation accuracy and reward that is achieved in the bandit setting.

So that the bandit algorithms could be tested in an offline setting, the following was decided: It was assumed that the action space of available adverts at the beginning of the simulation $\mathbf{A}^0$ was the entire set of adverts shown on the day. After iteratively showing adverts and observing if they did or did not receive a click, they are removed from $\mathbf{A}$ meaning that at each iteration $\mathbf{A}$ reduces by a row in size. Because there are a finite number of clicks in the day's worth of data, both bandit algorithms will finish the simulation with an equal cumulative reward (see Fig. 4); in a real-time online environment this will not be the case and the difference in the number clicks will be expected to increase over time. In the experiments performed it is assumed that clicks are generated independently with respect to time meaning that a user would choose to click or not click on an advert irrespective of when it was shown to them. It is important to reiterate that decrease in the improvement of using FAB-COST over the Laplace bandit over the simulation is due to the finite number of clicks available in the offline dataset. In a true online setting it is likely that FAB-COST will continue to outperform.

By measuring the difference in clicks received – as shown in Fig. 4 – it is clear that FAB-COST is the better algorithm, generating over 16% more clicks after around $31,000$ impressions, and given reasonable assumptions about continued new adverts we might expect such an improvement in performance to persist and yield significantly improved results over time.

V. Discussion

In this paper the Fast Approximate Bayesian Cold Start algorithm FAB-COST has been introduced; a fully Bayesian algorithm which combines both Expectation Propagation and Assumed Density Filtering to improve on the inference procedure for the logistic bandit proposed by [4]. Not only would it be beneficial to use FAB-COST's inference procedure in a bandit setting, but any online learning scenario for Bayesian logistic regression or with data sets prohibitively large for EP to be used on. This is the first time to the authors knowledge that either EP or ADF have been used to improve bandit performance. FAB-COST addresses two problems with the classic Laplace approximation: firstly it is an online scheme which can deal with large cumulative amounts of data and fast throughout; secondly it achieves better variance accuracy which will result in better balance between exploration and exploitation and hence improved website performance, which one would expect to see in a variety of contexts.

Algorithm 5: FAB-COST

Set E as the iteration(s) at which you want to make an EP update to the posterior.
Initialise the prior distribution $q_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
for $i = 1 \dots T$ **do**
  1. Generate a sample from the approximated posterior:
    $\tilde{\boldsymbol{\theta}}_i \sim \mathcal{N}(\boldsymbol{\mu}_{i-1}, \boldsymbol{\Sigma}_{i-1})$
  2. Select an advert:
    $a_i = \underset{\mathrm{j} \in \mathcal{A}}{\operatorname{argmax}}(\mathbf{A}^i \tilde{\boldsymbol{\theta}}_i)$
  3. Update moments via ADF:
    $\boldsymbol{\mu}_i = \boldsymbol{\mu}_{i-1} + \boldsymbol{\Sigma}_{i-1}\boldsymbol{\alpha}_i$
    $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i-1} - \boldsymbol{\Sigma}_{i-1}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top - 2\mathbf{B}_i\right)\boldsymbol{\Sigma}_{i-1}$
  4. IF i = E , perform an EP approximation as shown in algorithm 3 using the last E data points.
end

Acknowledgment

We thank David Hoyle and Robert Trickey from Auto Trader for their meets and discussions at the beginning of my PhD. We thank Edward Pyzer-Knapp from IBM research for his advice on how to take this paper forward. Acknowledgements also go to Simon Barthelmé for his emails and discussion on EP.

References

[1] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International Conference on Machine Learning*, 2013, pp. 151–159.
[2] T. Lattimore and C. Szepesvári, "Bandit algorithms," *preprint*, 2018.
[3] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.
[4] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, 2011, pp. 2249–2257.
[5] J. L. Doob, "Application of the theory of martingales," *Le calcul des probabilites et ses applications*, pp. 23–27, 1949.
[6] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.
[7] M. Opper and O. Winther, "Gaussian processes for classification: Mean-field algorithms," *Neural computation*, vol. 12, no. 11, pp. 2655–2684, 2000.
[8] ——, "A bayesian approach to on-line learning," *On-line learning in neural networks*, pp. 363–378, 1998.
[9] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, Eds. New York: Chapman and Hall/CRC.
[10] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, ser. Applied Logistic Regression. John Wiley & Sons, 2004.
[11] T. Hastie, R. Tibshirani, and J. Friedman., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, 2nd ed. New York: Springer-Verlag, 2009.

[12] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
[13] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 2019, pp. 101–109.
[14] A. DasGupta, *Asymptotic theory of statistics and probability*. Springer Science & Business Media, 2008.
[15] J. D. Murray, *Asymptotic Analysis*. New York: Springer, 2012.
[16] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[17] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
[18] J. M. Bernardo, "Approximations in statistics from a decision-theoretical viewpoint," in *Probability and Bayesian statistics*. Springer, 1987, pp. 53–60.
[19] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial intelligence*, vol. 29, no. 3, pp. 241–288, 1986.
[20] M. Seeger, "Expectation propagation for exponential families," Tech. Rep., 2005.
[21] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014.
[22] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013, pp. 127–135.
[23] M. Mézard, G. Parisi, and M. Virasoro, "Random free energies in spin glasses," *Journal de Physique Lettres*, vol. 46, no. 6, pp. 217–222, 1985.
[24] M. W. Seeger, "Bayesian inference and optimal design for the sparse linear model," *Journal of Machine Learning Research*, vol. 9, no. Apr, pp. 759–813, 2008.
[25] A. Gelman, A. Vehtari, P. Jylänki, C. Robert, N. Chopin, and J. P. Cunningham, "Expectation propagation as a way of life," *arXiv preprint arXiv:1412.4869*, vol. 157, 2014.
[26] G. Dehaene and S. Barthelmé, "Expectation propagation in the large data limit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 80, no. 1, pp. 199–217, 2018.
[27] G. P. Dehaene and S. Barthelmé, "Bounding errors of expectation-propagation," in *Advances in Neural Information Processing Systems*, 2015, pp. 244–252.
[28] R. Herbrich, "Minimising the kullback–leibler divergence," *Microsoft, Tech. Rep.*, 2005.
[29] D. J. MacKay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.
[30] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in python using PyMC3," *PeerJ Computer Science*, vol. 2, p. e55, 2016.