Query Clustering based on Bid Landscape for Sponsored Search Auction Optimization

Ye Chen
Microsoft Corporation
1020 Enterprise Way
Sunnyvale, CA 94089
yec@microsoft.com

Weiguo Liu
Microsoft Corporation
1020 Enterprise Way
Sunnyvale, CA 94089
weigliu@microsoft.com

Jeonghee Yi
Microsoft Corporation
1020 Enterprise Way
Sunnyvale, CA 94089
jeyi@microsoft.com

Anton Schwaighofer
Microsoft Corporation
7 J J Thomson Ave
Cambridge CB3 0FB, UK
antonsc@microsoft.com

Tak W. Yan
Microsoft Corporation
1020 Enterprise Way
Sunnyvale, CA 94089
takyan@microsoft.com

ABSTRACT

In sponsored search auctions, the auctioneer operates the marketplace by setting a number of auction parameters such as reserve prices for the task of auction optimization. The auction parameters may be set for each individual keyword, but the optimization problem becomes intractable since the number of keywords is in the millions. To reduce the dimensionality and generalize well, one wishes to cluster keywords or queries into meaningful groups, and set parameters at the keyword-cluster level. For auction optimization, keywords shall be deemed as interchangeable commodities with respect to their valuations from advertisers, represented as bid distributions or landscapes. Clustering keywords for auction optimization shall thus be based on their bid distributions. In this paper we present a formalism of clustering probability distributions, and its application to query clustering where each query is represented as a probability density of click-through rate (CTR) weighted bid and distortion is measured by KL divergence. We first derive a k-means variant for clustering Gaussian densities, which have a closed-form KL divergence. We then develop an algorithm for clustering Gaussian mixture densities, which generalize a single Gaussian and are typically a more realistic parametric assumption for real-world data. The KL divergence between Gaussian mixture densities is no longer analytically tractable; hence we derive a variational EM algorithm that minimizes an upper bound of the total within-cluster KL divergence. The clustering algorithm has been deployed successfully into production, yielding significant improvement in revenue and clicks over the existing production system. While motivated by the specific setting of query clustering, the proposed clustering method is generally applicable to many real-world applications where an example is better characterized by a distribution than a finite-dimensional feature vector in Euclidean space as in the classical k-means.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering.

General Terms

Algorithms, Experimentation, Performance

Keywords

Clustering; Bayesian methods; sponsored search; auction; optimization

1. INTRODUCTION

In search advertising, advertisers bid on keywords for advertising opportunities alongside algorithmic search results, through a generalized second-price auction (GSP) [6]. The bidder with the highest estimated click-through rate (CTR) weighted cost-per-click (CPC) bid (also known as rank score) wins the auction (impression opportunity). If the served ads are clicked, the advertisers pay the search engine (e.g., Google or Bing) the CTR adjusted next highest CPC bid.

The auctioneer or the search engine operates the marketplace by setting a number of auction parameters, which play an important part in determining the outcome of the auction. An example of an auction parameter is reserve prices; only ads that clear the reserve price participate in the auction [12, 13]. Another example is the exponent to which the CTR estimate is raised in the rank score function [9, 11]. Auction optimization is the task of finding parameters to optimize an objective such as maximizing click volume or revenue, while satisfying constraints such as the average number of ad impressions shown. One may seek to set the auction parameters for each individual keyword, but the optimization problem becomes intractable since the number of keywords is in the millions. To reduce the dimensionality for a parsimonious model that generalizes well, one wishes to cluster keywords or queries into meaningful groups, and set parameters at the keyword-cluster level.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

For the purpose of auction optimization such as reserve setting, keywords shall be regarded as interchangeable commodities with respect to their valuations from advertisers, more precisely the estimated CTR weighted CPC bids. The valuation of a keyword is the underlying parameter of a probability distribution referred to as bid landscape, and each advertiser's bid is a sample drawn from the distribution. Clustering keywords for auction optimization shall thus be based on their bid distributions. To see intuitively why representing a keyword as a bid distribution is more advantageous than point estimates for this specific task of auction optimization, let us consider reserve price setting as an example. It is clear that both mean and variance are relevant to finding optimal reserve. The mean reflects the overall valuation level, while the variance captures the important aspects of competitive landscape including bid density and spread.

The main contribution of this paper is to present a formalism of clustering probability distributions. We describe a query clustering algorithm where each query is represented as a probability density of CTR-weighted bid and distortion is measured by Kullback-Leibler (KL) divergence. We first derive a k-means variant for clustering Gaussian densities, which have a closed-form KL divergence, and show that the iterative algorithm monotonically decreases the total KL divergence. We then develop an algorithm for clustering Gaussian mixture densities, which generalize a single Gaussian and are typically a more realistic parametric assumption for real-world data. The KL divergence between Gaussian mixture densities is no longer analytically tractable; hence we derive a variational EM algorithm that minimizes an upper bound of the total within-cluster KL divergence.

The clustering algorithm has been deployed successfully into production at Bing Ads, and has produced keyword clusters for auction optimization. The method yielded a 22% gain in CTR over k-means in offline simulation, and a 5% improvement in revenue and clicks over the existing production system, which is a very significant improvement for a multi-billion dollar marketplace. As a consequence, the reported query clustering method is now serving 100% Bing and Yahoo search advertising traffic.

The paper is organized as follows. In Section 2, we formulate the problem of auction optimization. We then examine some empirical bid distributions in sponsored search auctions in Section 3, to support the parametric assumption that each keyword is represented as a Gaussian mixture density. In Section 4, we discuss a Bayesian perspective of clustering, and in Section 5, we formalize the problem of clustering probability distributions and derive a k-means variant, using Gaussian density as an illustrative example, while the approach remains general. We then in Section 6 generalize the single Gaussian model to Gaussian mixture model (GMM) and derive a variational EM algorithm with the otherwise analytically intractable KL divergence. Empirical results with the application of keyword clustering for auction optimization is presented in Section 7. Finally, we conclude the work in Section 8.

2. AUCTION OPTIMIZATION

The state of the art of auction optimization is to formulate an integer programming (IP) problem using counterfactual auction simulation as input. Before we formulate the auction optimization problem, let us first introduce the following concepts and notations for sponsored search.

1. **Ranking.** Given a keyword-ad pair, let the estimated position-unbiased CTR be ρ [3], the CPC bid be b, the rank score is defined as $s = b\rho^{\alpha}$, where α is called click investment power.[1] If $\alpha > 1$, ranking favors ads with higher estimated CTRs; otherwise, ranking favors ads with higher bids.

2. **Pricing.** In a GSP auction, if an ad from bidder (advertiser) i is clicked, her payment or price per click depends on the value per impression (rank score) of the next highest bidder $i+1$, i.e., $c_i = b_{i+1}\rho_{i+1}^{\alpha}/\rho_i^{\alpha}$. This design is to avoid dynamic bidding behavior [9] (the price c_i does not depend on her own bid b_i), while motivating high quality ads (the higher the CTR ρ_i, the lower the price c_i).

3. **Allocation.** Ads are allocated, in the descending order of their rank scores, from top to bottom display positions, i.e., the ad with the highest rank score is displayed in the first (top) slot, and so forth. Ad positions are primarily from two page sections: mainline (ML) refers to the positions above the algorithmic search results, and sidebar (SB) refers to the positions right to the algorithmic results [9, 13]. There are two reserve prices in the unit of rank score controlling the sectional allocation: mainline reserve R and sidebar reserve r. Given an ad i with rank score s_i, if $s_i \geq r$, the ad will be shown; and further if $s_i \geq R$, it will be shown in ML, with other constraints satisfied[2]. The reserves R and r also affect pricing, e.g., $c_i = \max(b_{i+1}\rho_{i+1}^{\alpha}, R)/\rho_i^{\alpha}$, if ad i is shown at the last position in ML.

4. **Optimization.** For auction optimization, one typically maximizes either revenue y or clicks w, constrained on the average number of ads shown in mainline per search result pageview, called ML impression yield or MLIY and denoted as a. Another constraint typically used in practice is the average number of ads shown overall both in mainline and sidebar per search result pageview, called impression yield or IY. Since the contribution to the objective function of either revenue or clicks is dominated by MLIY, we focus our discussion on the primary constraint on MLIY for simplicity.

Auction optimization consists of two steps as follows.

1. A counterfactual auction simulation is performed by applying a discrete set of candidate values of auction parameters $\mathcal{H} = \{h_j\}_{j=1}^{m}$ to a historical auction log, to compute a number of metrics $\mathcal{U} = \{u_j\}_{j=1}^{m}$. For example, one can simulate how many more clicks would have

[1] In practice, CTR estimates are typically at more granular levels including, e.g., user signals. We focus our discussion on the keyword-ad level to abstract from the nuances irrelevant to this work, without loss of generality. The keyword-ad pair is arguably the most predictive feature anyway. The rank score function has been evolving as well, and may contains other terms such as relevance of the ad landing page.

[2] In search advertising auction, there are other constraints on ad relevance (minimum relevance), CTR (minimum estimated CTR), and page layout (maximum numbers of ads in ML and SB), and so on.

been yielded if mainline reserve is reduced by half, by replaying the auction with lowered mainline reserve to compute impressions and predicting clicks using estimated CTR. Without loss of generality, we only consider the major parameters $h = (\alpha, R)$, efficiency and budget metrics $u = (y, w, a)$.

2. An integer programming (IP) problem is formulated based on the objective and constraint coefficients computed from the auction simulation, and is solved to find the optimal parameter settings, one for each keyword cluster.

Formally, let j index m parameter settings, p index k keyword clusters, and x_{pj} be the binary decision variable indicating whether to choose parameter setting j for keyword cluster p or not. A representative IP problem to maximize clicks is formulated as follows.

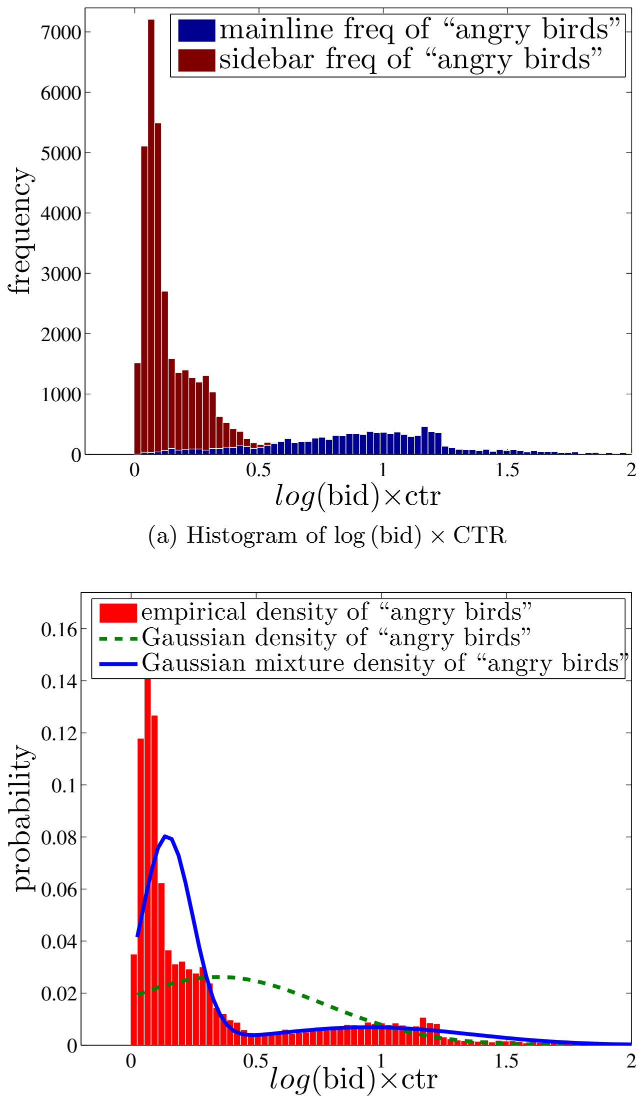
$$\begin{aligned} &\max_x \sum_{p,j} w_{pj} x_{pj} \\ &\text{s.t.} \sum_{p,j} y_{pj} x_{pj} \geq g_1; \\ &\sum_{p,j} (a_{pj} v_p x_{pj}) / \sum_p v_p \leq g_2; \\ &\sum_j x_{pj} = 1, \forall p; \\ &x_{pj} \in \{0, 1\}, \forall p, j. \end{aligned} \tag{1}$$

Here one maximizes clicks while lower bounding revenue by g_1. There is an empirical trade-off between these two objectives; whereas a sustaining auction shall yield revenue though clicks rather than CPC price. v_p is the number of search result pageviews for keyword cluster p, and g_2 is the global upper bound on MLIY. The coefficient matrix $[y_{pj}, w_{pj}, a_{pj}]_{k \times m}$ is obtained from auction simulation. The number of variables is $k \times m$, and in practice the number of settings m can easily be several hundred to cover a large number of parameters. It is clear that the optimization is intractable at the individual keyword level, where the number of keywords k is typically in the order of several million.

3. BID DISTRIBUTIONS

We wish to cluster keywords to reduce the dimensionality of the IP problem for a parsimonious optimal solution that generalizes well. For the purpose of clustering queries for setting parameters such as reserve prices, queries shall be treated as interchangeable commodities given same valuation distributions. To make a sound parametric assumption for the bid distribution in an appropriately chosen metric space, we examine the empirical distributions of rank scores for most frequently searched keywords, with several plausible metric transformations, as shown in Figure 1 for the example of "angry birds", and Figure 2 for the example of "flights san francisco".

First of all, we choose the coordinate space of log (bid) × CTR, instead of the original or logarithmic space of (bid × CTR). The original space of rank score does not exhibit any pattern of a smoothed parametric distribution, typically with a sharp spike focused on a very low rank score range. Bids (in cents) are observed from advertisers, hence a log transformation effectively squashes out large variances likely due to noises, e.g., from outlier bidders. On the other hand, a log transformation of CTR estimates is not necessary, since CTRs are already normalized into the range $[0, 1]$, typically below 10%. Empirically, we find that using the metric log (bid) × CTR works best for maximizing objectives, particularly clicks.



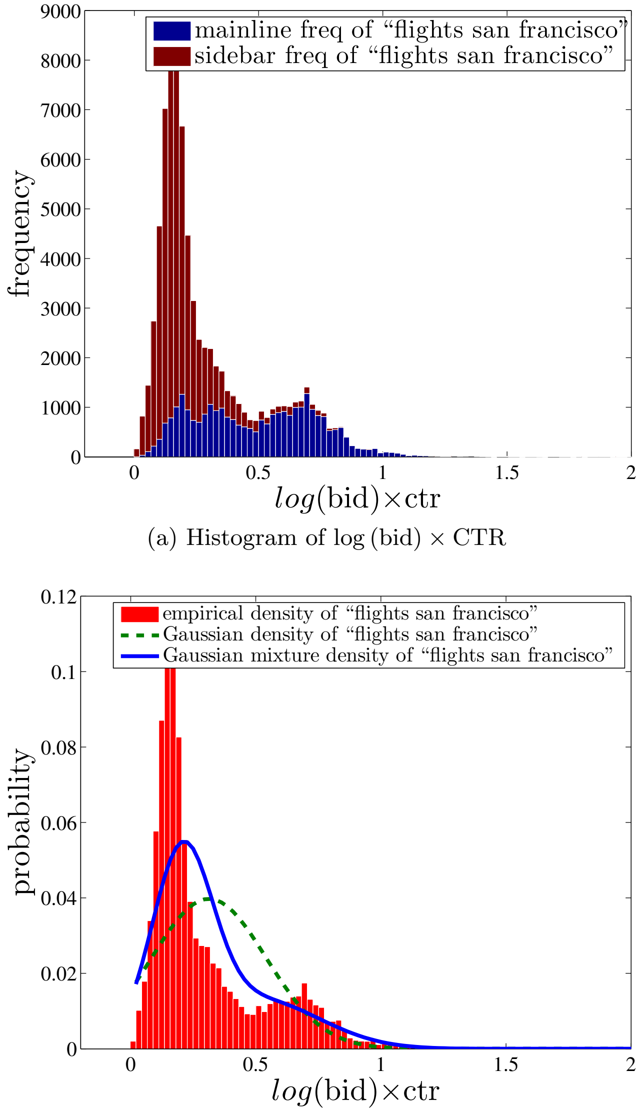
(a) Histogram of log (bid) × CTR

(b) Empirical and fitted densities of log (bid) × CTR

Figure 1: Empirical and fitted distributions of log (bid) × CTR for the keyword "angry birds".

Second and more appealingly, almost all empirical curves show a good fit to the two-component Gaussian mixture signature, as illustrated in Figures 1(a) and 2(a). One hypothesis is that there are two reserves, for mainline and sidebar respectively; bidders are aware and place bids reacting to one reserve at a time. This behavior is intuitive since advertisers usually have campaign goals and budgets. It turns out that our hypothesis is strongly supported by the data, as shown in Figures 1(a) and 2(a), the two peaks are dominated by mainline (blue) and sidebar (brown) ads, respectively.

Now that we have made the parametric assumption that each keyword is represented as a two-component Gaussian



(a) Histogram of log (bid) × CTR

(b) Empirical and fitted densities of log (bid) × CTR

Figure 2: Empirical and fitted distributions of log (**bid**) × **CTR for the keyword “flights san francisco”.**

mixture density of log (bid) × CTR, this is not only a more realistic assumption than, e.g., a single Gaussian, but also may expose better opportunities to the IP optimization, e.g., the saddle area between two peaks presents more feasible region for seeking optimal mainline reserve.

Further, given the two-reserve bidding mechanism evident from the data, the hidden variable or the membership of Gaussian component for each bid is known (think of the generative process of GMM). In other words, the mixture weights of GMM are known. This observation considerably reduces the complexity of learning GMMs for keywords, that is, the standard iterative EM is no longer needed. Recall that we need to learn one GMM for each of the possibly several million keywords. Specifically, fitting a GMM for each keyword has a closed-form solution as follow,

$$\mu_z = \mu(\rho_i \log b_i | i \in z), \forall z, \tag{2}$$

$$\sigma_z^2 = \sigma^2(\rho_i \log b_i | i \in z), \forall z, \tag{3}$$

$$\omega_z = \sum_{i \in z} 1 / \sum_i 1, \forall z, \tag{4}$$

where i indexes bids and z indexes mainline or sidebar. As shown in Figures 1(b) and 2(b), the fitted GMM (the blue solid line) captures the nature of the bid distribution much better than a single Gaussian (the green dotted line), particularly the sharpnesses of the two peaks.

4. CLUSTERING: A BAYESIAN PERSPECTIVE

Clustering is an unsupervised learning method widely used for dimensionality reduction. Given a set of examples $D = \{x_i\}_{i=1}^n$, the goal is to partition them into reasonable clusters. In k-means clustering, each example is represented as a feature vector $x_i \in \mathbb{R}^d$, and an iterative algorithm finds a locally optimal set of k clusters $\theta = \{\mu_j\}_{j=1}^k$ so as to minimize the distortion measured as the squared Euclidean distance $\theta^* = \text{argmin}_\theta L(D|\theta) = \text{argmin}_\theta \sum_j \sum_{i \in j} \|x_i - \mu_j\|_2^2$. The k-means algorithm is the limiting case of EM algorithm for Gaussian mixture models with infinitely small covariances.

The classical clustering methods hold a frequentist perspective in that data examples are a repeatable random sample from an underlying process with fixed parameters. The clustering task is then essentially inferencing the cluster centers θ as point estimates of means from repeatable observations, which are represented as point estimates as well. Recall that in the k-means case, minimizing quadratic loss $L(D|\theta)$ is equivalent to maximizing log likelihood $\ell(D|\theta)$ under Gaussian mixture models with infinitely small covariances, that is, the inference problem of computing the mode of log likelihood $\text{argmax}_\theta \, \ell(D|\theta)$ or the maximum likelihood estimate (MLE) of cluster centers θ. With the frequentist view, the classical clustering methods are more natural for clustering statically measurable objects such as documents and images using bag-of-words representation.

In many real-world applications, however, data examples to be clustered are better described probabilistically, and so are the cluster centers or representatives. In predictive modeling such as logistic regression, one wants to estimate the probability of an outcome given an unseen input feature vector $p(y|x)$. Clustering is typically applied to input features to reduce dimensionality for a better model generality. While clusters θ are learned from historical data, one is most interested in inferencing the cluster membership of a future x, which is bound to change. One motivational application is predicting CTR of search results or ads given a user among other input features [2]. To reduce the high dimensionality of user features (e.g., user ID or IP address), one may cluster users based on their click propensities. A frequentist approach would simply cluster point estimates of the Bernoulli success probability p, whose MLE is the sample mean, and hence unable to capture higher-order moments of the distribution of p such as variance and skewness. One important aspect of the distribution is the variance, e.g., for the purpose of exploring users with few Bernoulli trials formulated as a multi-armed bandit problem [7, 14]. The frequentist in this regard would characterize the distribution with a finite number of quantities, whereas clustering feature vectors of moments in Euclidean space has no clear interpretation.

A Bayesian perspective is more natural in such predictive settings where underlying parameters are unknown and their uncertainty is of interest. For the CTR prediction application, a user with respect to click propensity is represented as a probability distribution of the Bernoulli success probability p, that is, the posterior beta distribution. For clustering distributions, a natural and sound choice of distance function is the KL divergence, for its probabilistic underpinning and additivity. While KL divergence is asymmetric, centroid-based parametric clustering approaches [1] only require a directed distance function, i.e., from a cluster center to an example.

The significance of introducing a Bayesian perspective to the classical clustering methods is beyond the predictive setting as illustrated above. In many scenarios, an object shall be clustered with respect to some underlying parameters of a probability distribution, from which individual observations about the object are drawn. The motivational application in this paper is clustering keywords based on their underlying valuations for sponsored search auction optimization.

5. CLUSTERING GAUSSIAN DENSITIES

The Gaussian distribution is considered the most prominent probability distribution given the central limit theorem and its analytical tractability. We use Gaussian density as a representative example to derive a formal clustering algorithm. This choice is mathematically convenient and helps to guide intuition, yet it is sufficient to illustrate the underlying principle. It is important to emphasize, however, that the formulation of the clustering problem generalizes to other distributions.

Let us assume that the observations x for an object p follow a Gaussian distribution. Each object can be represented as a Gaussian density:

$$p(x) \sim \mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{5}$$

The goal is to cluster objects with similar distributions together, for which we need a pairwise distance function. A natural choice of distance measure between two distributions denoted by p and q is the KL divergence:

$$D_{\mathrm{KL}}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx. \tag{6}$$

KL divergence measures in bits how close a model distribution q is to the true underlying distribution p. Although KL divergence is not symmetric nor does it satisfy the triangle inequality, for centroid-based parametric clustering methods [1], such as the one we present, one only needs a directed distance measure from a cluster center p to an object q. KL divergence is additive for independent variable distributions, that is, $D_{\mathrm{KL}}(p\|q) = D_{\mathrm{KL}}(p_1\|q_1) + D_{\mathrm{KL}}(p_2\|q_2)$ if $p(x,y) = p_1(x)p_2(y)$ and $q(x,y) = q_1(x)q_2(y)$. This is particularly useful for clustering distributions based on KL divergence, since it allows for a clear interpretation of minimizing total KL divergence.

We now formulate the optimization problem underlying the task of clustering Gaussian densities, following a similar approach as k-means. Let us begin with the KL divergence from a cluster center $p \sim \mathcal{N}\left(\mu_p, \sigma_p^2\right)$ to an example $q \sim \mathcal{N}\left(\mu_q, \sigma_q^2\right)$. The KL divergence between two Gaussian densities has a closed form.

$$D_{\mathrm{KL}}^{\mathrm{Gauss}}(p\|q) = \frac{1}{2}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{(\mu_p - \mu_q)^2}{\sigma_q^2} - \log\left(\frac{\sigma_p^2}{\sigma_q^2}\right) - 1\right). \tag{7}$$

Given a set of training examples $D = \{q \sim \mathcal{N}\left(x; \mu_q, \sigma_q^2\right)\}$ indexed by q and observed as x, we wish to learn a set of cluster centers $\theta = \{p \sim \mathcal{N}\left(x; \mu_p, \sigma_p^2\right)\}$ indexed by p, by minimizing the loss in KL divergence.

$$\min_{\mu_p, \sigma_p^2; \forall p} L(D|\theta) = \sum_p \sum_{q \in p} D_{\mathrm{KL}}^{\mathrm{Gauss}}(p\|q), \tag{8}$$

where $q \in p$ denotes that the example q belongs to the cluster p.

If the assignment $q \in p, \forall p, q$ is fixed, the objective function L is convex in μ_p and σ_p^2. This is an unconstrained optimization problem, thus we set the partial derivatives of L w.r.t. μ_p and σ_p^2 to zero to derive the update rule for cluster centers.

$$\begin{aligned} \frac{\partial L}{\partial \mu_p} &= \sum_{q \in p} \frac{1}{2}\left(\frac{2(\mu_p - \mu_q)}{\sigma_q^2}\right) \to 0 \\ \mu_p &= \left(\sum_{q \in p} \frac{1}{\sigma_q^2}\mu_q\right) / \sum_{q \in p} \frac{1}{\sigma_q^2}; \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\partial L}{\partial \sigma_p^2} &= \sum_{q \in p} \frac{1}{2}\left(\frac{1}{\sigma_q^2} - \frac{\sigma_q^2}{\sigma_p^2}\frac{1}{\sigma_q^2}\right) \to 0 \\ \sigma_p^2 &= \left(\sum_{q \in p} 1\right) / \sum_{q \in p} \frac{1}{\sigma_q^2}. \end{aligned} \tag{10}$$

For fixed cluster centers μ_p and $\sigma_p^2, \forall p$, we simply assign q to its closest cluster center $p(q)$ in terms of KL divergence. We now arrive at an iterative algorithm that monotonically decreases the loss in KL divergence as follows.

1. Randomly choose k examples as cluster centers.
2. Repeat until convergence.
   (a) Assignment step:
   $$p(q) = \operatorname*{argmin}_p D_{\mathrm{KL}}^{\mathrm{Gauss}}(p\|q), \forall q. \tag{11}$$
   (b) Update step:
   $$\mu_p = \left(\sum_{q \in p} \frac{1}{\sigma_q^2}\mu_q\right) / \sum_{q \in p} \frac{1}{\sigma_q^2}, \forall p; \tag{12}$$
   $$\sigma_p^2 = \left(\sum_{q \in p} 1\right) / \sum_{q \in p} \frac{1}{\sigma_q^2}, \forall p. \tag{13}$$

The optimal solution to the cluster centers μ_p and σ_p^2, as in Eqs. (12) and (13), reveals an appealing yet somewhat nontrivial intuition. First, the optimizing center mean μ_p is an inverse-variance weighted average of example means μ_q. Inverse-variance weighted averaging is known to minimize the variance of the sum and is typically used in statistical meta-analysis to combine evidences from independent studies [8]. Intuitively, we weight studies to give preference to the more precise ones with larger samples. This is appropriate for clustering distributions, since the measurements or sampling of x for each example distribution q shall be treated as individual studies, instead of from a single large study, to allow for other differences such as in variance. The latter views examples q as parts of a single sampling, whose mean would be a sample-size weighted average of example means. Second, the optimizing center variance σ_p^2 is the harmonic mean of example variances σ_q^2. Harmonic mean is typically used for averaging rate variables, and variance can be interpreted as the rate of imprecision of a study. A more well-known metric is the F_1-score, defined as the harmonic mean of precision and recall, from information retrieval.

The proposed algorithm for clustering Gaussian densities can be viewed as a special case of the Bregman clustering problem [1], in that KL divergence is the Bregman divergence realized from the convex function $\phi(p) = \sum_j p_j \log p_j$ in a d-simplex. In [1], Banerjee et al. unify centroid-based clustering approaches into a meta hard clustering algorithm that is applicable to all Bregman divergences including squared Euclidean distance and KL divergence. They also show that there is a bijection between regular exponential families and regular Bregman divergences. These findings establish a general theoretical foundation for our work. More specifically, the update step for the centroid mean in the Bregman hard clustering (Algorithm 1 in [1]) has the form: $\mu_j \leftarrow \frac{1}{\pi_j} \sum_{i \in j} \nu_i x_i$, where i indexes examples, j indexes cluster centers, $\pi_j = \sum_{i \in j} \nu_i$, and $\nu = \{\nu_i\}_{i=1}^n$ is a probability measure over $\mathcal{X} = \{x_i\}_{i=1}^n$. We have shown, in Eqs. (12) and (13), that for Gaussian densities with unknown μ and σ^2, the general probability measure ν_i in Bregman hard clustering is realized as the inverse variance $1/\sigma_i^2$.

6. CLUSTERING GAUSSIAN MIXTURE DENSITIES

We have formalized the problem of clustering probability distributions with KL divergence, and derived a simple k-means type iterative algorithm for Gaussian densities. In this section, we generalize a single Gaussian to Gaussian mixture model (GMM), which often times appears to be a better parametric assumption for many real-world applications such as speech and image recognition [10].

Let us begin with a simplified yet practically representative case of GMMs with equal-number non-exchangeable components. The Gaussian mixture densities of a cluster center p and an example q are:

$$p(x) = \sum_z \pi_z p_z(x), \tag{14}$$
$$\text{where } \sum_z \pi_z = 1 \text{ and } p_z(x) \sim \mathcal{N}(x; \mu_{pz}, \sigma_{pz}^2).$$
$$q(x) = \sum_z \omega_z q_z(x), \tag{15}$$
$$\text{where } \sum_z \omega_z = 1 \text{ and } q_z(x) \sim \mathcal{N}(x; \mu_{qz}, \sigma_{qz}^2).$$

Here z indexes matched components, π_z and ω_z are mixture weights, p_z and q_z are component Gaussian, for the cluster center and the example, respectively.

The KL divergence between two GMMs is no longer analytically tractable, which renders exact EM impossible. One solution is to use variational inference that decreases but not necessarily minimizes the loss in KL divergence, so as to find approximate estimates of cluster center parameters $\theta = \{\mu_{pz}, \sigma_{pz}^2, \pi_{pz}; \forall p, z\}$. Let us treat the assignment $q \in p, \forall p, q$ as variational parameters (indicator variables) and cluster centers θ as model parameters, which is invariant for finding a local minimum. We first minimize a tractable upper bound w.r.t. the variational parameters $q \in p, \forall p, q$, and then for fixed variational parameters, minimize the upper bound w.r.t. the model parameters θ, alternating until convergence. This procedure is known as variational EM algorithm.

Let us first give an upper bound on the otherwise intractable KL divergence.

$$\begin{aligned} D_{\mathrm{KL}}^{\mathrm{GMM}} &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \int \left(\sum_z \pi_z p_z \right) \log \left(\frac{\sum_z \pi_z p_z}{\sum_z \omega_z q_z} \right) dx \\ &\leq \int \sum_z \left(\pi_z p_z \log \left(\frac{\pi_z p_z}{\omega_z q_z} \right) \right) dx \qquad (16) \\ &= \sum_z \pi_z \log \left(\frac{\pi_z}{\omega_z} \right) + \sum_z \pi_z \int p_z \log \left(\frac{p_z}{q_z} \right) dx \\ &= D_{\mathrm{KL}}(\pi \| \omega) + \sum_z \pi_z D_{\mathrm{KL}}^{\mathrm{Gauss}}(p_z \| q_z). \end{aligned}$$

The inequality in Eq. (16) follows from the log-sum inequality [4, 5], that is, $x \log \left(\frac{x}{y}\right) \leq \sum_i x_i \log \left(\frac{x_i}{y_i}\right)$, where $x = \sum_i x_i, y = \sum_i y_i$ and $x_i, y_i \geq 0$, with equality iff $\frac{x_i}{y_i}$ is a constant. When p and q are aligned well in terms of both mixture weights and component Gaussian, $\frac{\pi_z p_z}{\omega_z q_z}$ approaches one $\forall z$, $D_{\mathrm{KL}}^{\mathrm{GMM}}$ approaches zero, so does its upper bound, which tends to be tight. This is particularly useful for hard clustering since, in the assignment step, one only seeks the minimizing p. At the other extreme, when all components except one vanish, The KL divergence between GMMs $D_{\mathrm{KL}}^{\mathrm{GMM}}$ degrades to the one for Gaussian $D_{\mathrm{KL}}^{\mathrm{Gauss}}$. This is why the clustering algorithm for GMMs is a generalization of clustering Gaussian densities, hence can be used directly for the latter.

The optimization problem defined on the upper bound is

$$\begin{aligned} &\min_{\mu_{pz}, \sigma_{pz}^2, \pi_{pz}; \forall p, z} L(D|\theta) \\ &= \sum_p \sum_{q \in p} \left(D_{\mathrm{KL}}(\pi_p \| \omega_q) + \sum_z \pi_{pz} D_{\mathrm{KL}}(p_z \| q_z) \right) \\ &= \sum_p \sum_{q \in p} \left(\sum_z \pi_{pz} \log \left(\frac{\pi_{pz}}{\omega_{qz}}\right) + \right. \qquad (17) \\ &\left. \sum_z \pi_{pz} \frac{1}{2} \left(\frac{\sigma_{pz}^2}{\sigma_{qz}^2} + \frac{(\mu_{pz} - \mu_{qz})^2}{\sigma_{qz}^2} - \log \left(\frac{\sigma_{pz}^2}{\sigma_{qz}^2}\right) - 1 \right) \right). \\ &\text{s.t. } \sum_z \pi_{pz} = 1, \forall p. \end{aligned}$$

For a fixed assignment $q \in p, \forall p, q$, this is a constrained optimization problem. We form the Lagrangian

$$\mathcal{L} = L(D|\theta) - \sum_p \lambda_p \left(\sum_z \pi_{pz} - 1 \right), \tag{18}$$

and set its partial derivatives w.r.t. μ_{pz}, σ_{pz}^2 and π_{pz} to zero.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_{pz}} &= \sum_{q\in p}\left(\pi_{pz}\frac{1}{2}\frac{1}{\sigma_{qz}^2}2(\mu_{pz}-\mu_{qz})\right) \to 0\\ \mu_{pz} &= \left(\sum_{q\in p}\frac{1}{\sigma_{qz}^2}\mu_{qz}\right)/\sum_{q\in p}\frac{1}{\sigma_{qz}^2};\end{aligned} \tag{19}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma_{pz}^2} &= \sum_{q\in p}\left(\pi_{pz}\frac{1}{2}\left(\frac{1}{\sigma_{qz}^2}-\frac{\sigma_{qz}^2}{\sigma_{pz}^2}\frac{1}{\sigma_{qz}^2}\right)\right) \to 0\\ \sigma_{pz}^2 &= \left(\sum_{q\in p}1\right)/\sum_{q\in p}\frac{1}{\sigma_{qz}^2};\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \pi_{pz}} &= \sum_{q\in p}\left(\log \pi_{pz} - \log \omega_{qz} + 1 + D(p_z\|q_z)\right) - \lambda_p \to 0\\ \pi_{pz} &\propto \exp\left(\sum_{q\in p}(\log \omega_{qz} - D(p_z\|q_z) - 1)/\sum_{q\in p}1\right).\end{aligned} \tag{21}$$

For cluster assignment with fixed centers μ_{pz}, σ_{pz}^2 and $\pi_{pz}, \forall p, z$, we assign q to its closest cluster $p(q)$ in terms of the upper bound. We now arrive at the variational EM algorithm as follows.

1. E-step:

$$p(q) = \operatorname*{argmin}_p D(\pi_p\|\omega_q) + \sum_z \pi_{pz} D(p_z\|q_z), \forall q. \tag{22}$$

2. M-step:

$$\mu_{pz} = \left(\sum_{q\in p}\frac{1}{\sigma_{qz}^2}\mu_{qz}\right)/\sum_{q\in p}\frac{1}{\sigma_{qz}^2}, \forall p, z; \tag{23}$$

$$\sigma_{pz}^2 = \left(\sum_{q\in p}1\right)/\sum_{q\in p}\frac{1}{\sigma_{qz}^2}, \forall p, z; \tag{24}$$

$$\pi_{pz} \propto \exp\left(\frac{\sum_{q\in p}(\log \omega_{qz} - D(p_z\|q_z) - 1)}{\sum_{q\in p}1}\right), \forall p, z. \tag{25}$$

The EM recurrence substantiates simple intuitions. In the M-step update for the cluster center mixture weights π_{pz} (Eq. (25)), the belonging example mixture weights or priors ω_{qz} contribute multiplicatively as $\exp\sum_{q\in p}(\log \omega_{qz}\ldots)$, while penalized by their matched-component KL divergence $D(p_z\|q_z)$, or equivalently, the negative log likelihood [1]. The M-step update for the component Gaussian parameters μ_{pz} and σ_{pz}^2 (Eqs. (23) and (24)) is very similar to the single Gaussian case (Eqs. (12) and (13)), except at the component level.

As the clustering algorithm (the assignment E-step and the update M-step) iterates, the total KL divergence decreases and clusters become more homogeneous, hence the upper bound becomes tight. In practice, we find that the clustering algorithm converges sublinearly, and typically converges after 20 to 30 iterations.

It is important to note that, when some example variances σ_{qz}^2 approach zero, the optimization problem becomes ill-conditioned. Since example variances σ_{qz}^2 appear as denominators in the update formulae (Eqs. (23), (24) and (25)), the zero-variance examples will dominate hill climbing. One approach to coping with this numerical issue is to smooth the Gaussian parameters by adding an i.i.d. zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \varsigma^2)$ to each observation x, and the resulting Gaussian parameters of example q is $\mathcal{N}(\mu_{qz}, \sigma_{qz}^2 + \varsigma^2), \forall z$. The smoothing variance ς^2 can be chosen in a data-driven manner, e.g., the first percentile of nonzero variances. In fact, the smoothing variance ς^2 introduces a useful mechanism to control whether the clustering shall emphasize more on mean or variance, depending on different applications. A sufficiently large smoothing ς^2 effectively makes the clustering based upon the example means. On the other hand, smoothing cluster center variances σ_{pz}^2 is not necessary, since they never appear as denominators in the algorithm.

7. EXPERIMENTAL RESULTS

With the learned GMMs for keywords described in Section 3, we apply the variational EM algorithm described in Section 6 to cluster keywords into k partitions. We collected auction related data (e.g., bids, CTR estimates, and display positions) over a one-month period, learned k clusters from a smaller set of most frequent keywords (about $1M$), and performed a final assignment step to infer clusters for all keywords (about $8M$). The choice of k is made such that clusters will have a mild loss in entropy, while the dimensionality $k \times m$ fits well with the IP solver. A good empirical choice is $k = 2000 \sim 4000$. The clustering results are visualized in Figure 3. Figure 3(a) shows how clusters are spanning in the 3D space of $(\mu_{ml}, \mu_{sb}, \pi_{ml})$, where each ball denotes a cluster center, with a volume proportional to σ_{ml}^2. It is clear that the algorithm does not partition examples in the Euclidean sense, e.g., more clusters are derived in the low-variance area since those examples have greater impacts on the total loss in KL divergence (Eq. (16)). Figure 3(b) illustrates how keywords are clustered, where each ball represents a keyword GMM and each same-color cloud forms a cluster. The clustering exhibits a meaningful yet non-Euclidean pattern, e.g., low-variance clusters are denser in belonging keywords.

Finally, we evaluate the effectiveness of the proposed GMM clustering algorithm in the context of auction optimization, through both offline simulation and online A/B testing. The IP problem is formulated as:

$$\begin{aligned}&\max\{\text{clicks}\}\\ &\quad \text{s.t. revenue} \geq 1.0;\\ &\qquad \text{MLIY} \leq 1.05;\\ &\qquad \text{both constraints relative to actual log traffic.}\end{aligned} \tag{26}$$

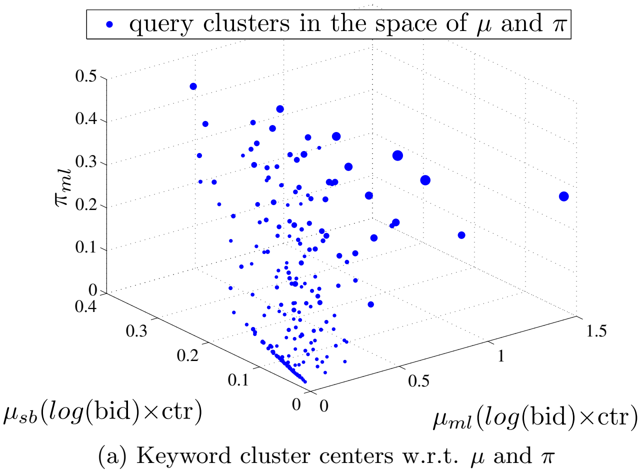
This optimization is to maximize clicks given a 5% more budget in mainline impression yield, while maintaining the same amount of revenue.

In offline experiments, we compare the proposed GMM clustering (k-GMM) with three benchmarks:

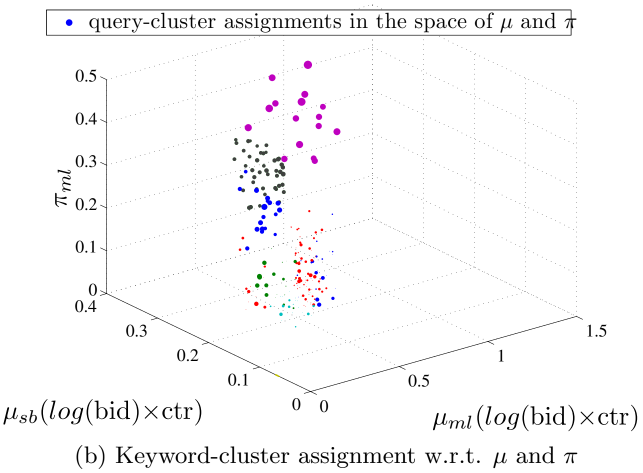
1. The Gaussian clustering (k-Gauss) that represents each keyword as a single Gaussian of $\rho \log b$, as described in Section 5.

2. The k-means clustering that represents each keyword as a vector of valuation-based features:

$$\begin{aligned}q_i = \Big(&\mu(n_i), \mu(\log(b_i)), \sigma(\log(b_i))/\mu(\log(b_i)),\\ &\mu_{ml}(\rho_i \log(b_i)), \mu_{sb}(\rho_i \log(b_i))\Big).\end{aligned} \tag{27}$$



(a) Keyword cluster centers w.r.t. μ and π



(b) Keyword-cluster assignment w.r.t. μ and π

Figure 3: Keyword clustering results.

Here n_i is the number of bids casted for query q_i in one auction, also called bid density or auction density, and $\mu(n_i)$ is the average bid density over all auctions for query q_i. The standard deviation of log bid is normalized by the mean $\sigma(\log(b_i))/\mu(\log(b_i))$. For a proper scaling, each feature dimension is quantized into univariate percentile. This feature vector is chosen as the empirical optimal based upon careful engineering and extensive experiments. In particular, the k-means feature vector encodes the same domain knowledge of two-reserve bidding as μ_{ml} and μ_{sb}, and uses the same metric space of $\rho_i \log(b_i)$.

3. A univariate binning approach (k-bins) that partitions keywords by their 95th rank score percentiles, which is the current method used in production and reflected in the actual log.

The offline simulation results are summarized in Table 1. The k-GMM clustering outperforms all other methods with respect to lift in clicks over actual log traffic. The ratio Δclicks/ΔMLIY measures the efficiency of converting ad impression to click by a particular method. With a 5% MLIY budget, k-GMM yields a 5.3-fold improvement in click-converting efficiency over the existing approach in production k-bins, a 22% improvement over k-means primarily due to the Bayesian treatment, and a 48% improvement over k-Gauss in consequence of a sound parametric assumption.

Table 1: Auction optimization results with different clustering methods

Model	Lift in clicks@5% MLIY
k-GMM	**13.01%**
k-Gauss	8.78%
k-means	10.66%
k-bins	2.46%

We have also conducted online A/B testing to compare the k-GMM clustering algorithm with the current k-bins approach in production. The online experiment ran for a two-week period and accounted for 16.2M search result pageviews. The results are shown in Table 2. The proposed GMM clustering has gained a 5.60% revenue lift over the existing k-bins approach, entirely from the gain in clicks 5.79% with a slight and favorable drop in CPC price −0.27%, at an approximately same MLIY level 0.80%. As a consequence, the novel query clustering method has been successfully deployed to the Bing search engine.

Table 2: Online A/B testing results

Metric	Lift of k-GMM over k-bins
Revenue	5.60%
Clicks	5.79%
CPC	−0.27%
MLIY	0.80%

8. CONCLUSIONS

We have presented a formalism of clustering probability distributions, motivated by real-world applications where observations are drawn from underlying distributions and the goal is to cluster the underlying concepts with uncertainty. An appealing Bayesian analog is that the cluster center or representative distribution is the prior $p(\theta)$ and the example distribution is the posterior $p(\theta|D)$. We have derived the algorithms for clustering Gaussian densities and GMMs, while the underlying principle generalizes to other distributions such as beta distribution for binomially distributed data, Dirichlet distribution for multinomial data, and gamma distribution for Poisson data. The algorithm has been applied to the important problem of sponsored search auction optimization, and yielded significant improvement in CTR over k-means in offline simulation, and as well as improvement in revenue and clicks over the existing production system.

9. REFERENCES

[1] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[2] Y. Chen, M. Kapralov, D. Pavlov, and J. F. Canny. Factor modeling for advertisement targeting. *Advances in Neural Information Processing Systems (NIPS 2009)*, 22:324–332, 2009.

[3] Y. Chen and T. W. Yan. Position-normalized click prediction in search advertising. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012)*, 2012.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*, page 26. Wiley-Interscience, 99th edition, 1991.

[5] M. N. Do. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Processing Letters*, 10(4):115–118, 2003.

[6] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

[7] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B (Methodological)*, 41(2):148–177, 1979.

[8] G. V. Glass. Primary, secondary, and meta-analysis of research. *Educational Researcher*, 5(10):3–8, 1976.

[9] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 13–20, 2010.

[10] J. R. Hershey and P. A. Olsen. Approximating the Kullback-Leibler divergence between Gaussian mixture models. *2007 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2007)*, 4:IV–317–IV–320, 2007.

[11] S. Lahaie and P. Mcafee. Efficient ranking in sponsored search. *WINE 2011, LNCS 7090*, pages 254–265, 2011.

[12] M. Ostrovsky and M. Schwarz. Reserve prices in internet advertising auctions: a field experiment. *Stanford University Graduate School of Business Research Paper No. 2054*, 2009.

[13] F. Pin and P. Key. Stochastic variability in sponsored search auctions: observations and models. *Proceedings of the 12th ACM Conference on Electronic Commerce (EC 2011)*, pages 61–70, 2011.

[14] A. Slivkins. Multi-armed bandits on implicit metric spaces. *Advances in Neural Information Processing Systems (NIPS 2011)*, 24:1602–1610, 2011.