Towards an Efficient Algorithmic Framework for Pricing Cellular Data Service

MohammadHossein Bateni* MohammadTaghi Hajiaghayi† Sina Jafarpour* Dan Pei‡

* Princeton University † University of Maryland, College Park ‡ IEEE Member

***Abstract*—As wireless service providers move from flat-fee unlimited data plans to tiered usage-based ones, there has been little published research on how such tiered plans should be designed. In this paper, we tackle this problem from an algorithmic perspective: formulating the problem of tiered data pricing plans for a wireless provider, and proposing an efficient algorithmic framework to compute the plans. Our algorithmic framework can be applied to the usage and cost data of any provider to obtain the pricing functions specific to that provider.**

I. INTRODUCTION

As bandwidth-heavy mobile phones and applications become more and more popular, cellular providers are investing heavily on bandwidth increase and technology upgrade, and encouraging WiFi and Femtocell traffic offloading. Another important method to deal with this trend is to price the plans to better reflect the cost to carry individual user's traffic, just like for other commodities such as electricity and water where users are charged with higher rate for their heavy usage. Some major U.S. wireless providers already started offering usage-based tiered plans. Each of these plans has a monthly allowance for a flat-fee and then a linear charge rate for the extra usage beyond the allowance.

While this trend seems inevitable (with the assumption of continuing wireless data usage explosion and the fundamental limitation of wireless spectrum capacity), there is little published research regarding how tiered data plans should be designed for wireless providers, The challenges include (1) a trade-off between revenue increase and plan complexity as previous studies such as [1] show that typical customers prefer simpler plan structure, and are willing to pay more for flat-rate unlimited pricing in general; (2) a trade-off between charging heavy users more and offering them cheaper price per unit traffic (economies of scale), reflected in convex vs. concave pricing functions; (3) a trade-off between revenue increase and customer's tendency to switch to other providers; and (4) a model framework and corresponding algorithms that output the pricing plan based on empirical usage and cost data from any given provider.

We tackle this problem by studying it algorithmically: we formulate the problem of tiered data pricing for a wireless provider, and propose an efficient algorithmic framework to compute the plans. It consists of (1) a nonparametric Gaussian mixture model for user usage; (2) a service cost function for carrying a certain amount of usage of a customer; (3) a general pricing function for improving the provider's revenue with the consideration of potential cutomer churn, and an approach to simplify the general pricing function; and (4) an efficient dynamic programming algorithm to compute the pricing policies.

Based on a few months of anonymized actual user usage from a major US wireless provider, our nonparametric Gaussian mixture model decomposes the actual data usage into four lognormal distributions. We observe that the data download usage strongly correlates with phone models. When applied to the empirical usage data and synthetic cost data, our proposed dynamic-programming algorithm finds the solutions in a couple of minutes, fast enough to allow the providers to experiment with different parameters.

To the best of our knowledge, this is the first paper proposing an *efficient* algorithmic framework for studying *wireless data* pricing policies. For *wired* Internet data pricing, there are only a few explanatory and survey papers by Odlyzko [2], [3], [4], [5], Reichl et al. [6], Wang and Li [7], and others [8]. We are not aware of any computationally efficient algorithmic study to maximize the revenue of a provider; a recent work of Iyengar et al. [9] proposes a probabilistic method for inferring the cost function, though their algorithms are not efficient. Furthermore, voice and data usage distributions are quite different, and our framework considers heavy *data* user's impact on the network and other users, since data traffic typically has much higher bit rate than voice and is more likely to cause network congestion due to the fundamentally limited wireless spectrum capacity.

II. USAGE MODEL

We start by analyzing the current customer usage, which is an important factor for pricing. In addition to data usage, we also study voice usage for comparison purposes. Let $\mathcal{U}$ denote the set of different usage values. In general it may be simply $\mathbb{R}_+$, but in most practical applications one can assume $\mathcal{U}$ to be the set of nonnegative integers less than a large limit $\mathcal{U}^{\max}$. We define a *usage function* $\sigma : \mathcal{U} \mapsto \mathbb{N}$ that specifies the number of customers having a particular usage. The anonymized usage data were collected for a few months during which customers had unlimited flat-rate data plan.

A. Mixture models

We gather the usage data of several million users over several months in 2010. For purposes of analyzing large datasets, it is useful and intuitive to understand the mathematical model behind them. Here we use a generative probabilistic approach.

In the Gaussian mixture model, each data point d_i is sampled i.i.d. from a collection (also called mixture) of Gaussian distributions: $\forall i \in \{1, \ldots, N\} : d_i \sim \sum_{j=1}^{n} \omega_j \mathcal{N}(\nu_j, \sigma_j)$, where $\sum_{j=1}^{n} \omega_j = 1$ and $\omega_j \geq 0$ for $1 \leq j \leq n$. In case n is not known a priori, this parameter can also be learned from the data set. The nonparametric distribution models [10] are as follows: $\forall i \in \{1, \ldots, N\} : d_i \sim \sum_{j \geq 1} \omega_j \mathcal{N}(\nu_j, \sigma_j)$, where $\sum_{j \geq 1} \omega_j = 1$ and $\omega_j \geq 0$ for $j \geq 1$.

Nonparametric mixture models have the advantage that the number of mixture components n does not need to be known a priori. However, unless some prior restriction is imposed on the component weights $(\omega_j)_{j \geq 1}$, the learned weight vector can be dense, causing the overfitting problem[1]. Latent Dirichlet Allocation (LDA) [11] is the simplest nonparametric approach that learns the most probable weight vector $(\omega_j)_{j \geq 1}$ under the *prior* assumption that the sequence is sampled from a Dirichlet distribution. A Dirichlet distribution is a distribution over distributions such that sparser distributions have higher probability density.

After gathering the data for the usage of the customers, we learn the parameters n and $\omega_j, \nu_j, \sigma_j$ for $j = 1, \ldots, n$. This can be done efficiently via the Expectation Maximization Algorithm (EM) [12], empowered by variational inference methods [13]. We use the "lda" variational inference package [14] to infer the mixture model. Moreover, we only consider the Gaussian components with $\omega_j \geq 0.05$.

B. Experimental data

Our experimental analysis illustrates that the usage data, when considered in logarithmic scale, can be modeled with a good precision by a few Gaussian distributions. Different Gaussian distributions suggest that the users belong to different categories: e.g., those who watch movies on their cell-phones, those who browse the web, and those who only read emails. We analyze and compare the distributions of the download, upload, and voice usages separately:

Download usage: Here the provided download usage of the users (in logarithmic scale) is used to learn a nonparametric mixture model. Moreover, motivated by the empirical frequency statistics of the data, we also include an explicit constraint, requiring that the learning algorithm has to learn at least one component in each interval $[0, 4)$, $[4, 7)$, and $[7, 11)$.

Fig. 1 illustrates the learned model. The decomposition of the mixture model into its four components provides the following clustering of the users:

- Light users: The Gaussian components *concentrated* at 2.63 represents light users, e.g., those who only read emails.
- Medium users: The overlapping Gaussian components centered at 4.53 and 6.28 correspond to medium users, e.g., those who browse the web. These two components have heavy tails, and form the majority of the users.

[1]In the extreme case, every data point d_i can be learned as an individual Gaussian cluster.

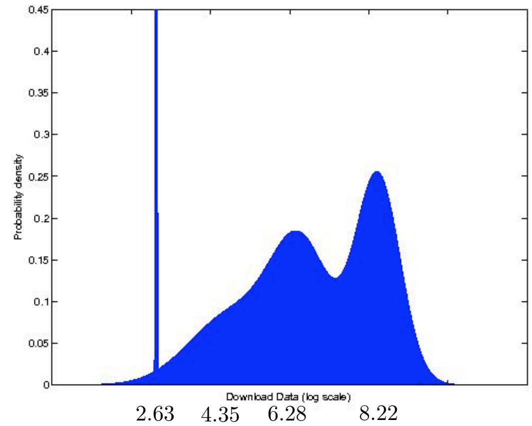

Fig. 1: Distribution of the logarithm of download usage data is modeled as a mixture of four Gaussian distributions. The numbers on the x-axis do not correspond to any natural usage value.

- Heavy users: The largest Gaussian component, centered at 8.22, represents heavy users, e.g., those who watch movies on their smart phones.

We also found that the data download usage, not surprisingly, strongly correlates with phone models.

Upload and voice usage: A similar analysis for upload usage leads to four lognormal components, although these components are closer to each other. Voice usage is a completely different story: it has only one lognormal distribution since typical customers do not belong to different behavioral groups according to their voice usage. Furthermore, we observe no correlation between voice usage and phone models.

III. Cost model

Our basic cost model is that the wireless service provider spends $\phi(x)$ dollars to serve x amount of data usage to a customer. We assume that the *service cost* function, ϕ, is an increasing function. In this section, we first present our conjecture on how the cost function might look purely based on the public plan information and some networking intuition. Obtaining an accurate cost function is often impossible due to its proprietary and sensitive nature, and is beyond the scope of this paper. In addition, the cost fucntion is affected in reality by so many parameters that the existence of a closed mathematical form for it is unlikely.

We then show that the cost function can be approximated and be specified by a small collection of threshold functions with very small error.

A. Model for Cost Function

This section presents our conjectured service cost function, which is an input to our pricing algorithm. The output of our algorithm changes as the cost function changes, but the algorithm itself remains the same. A wireless provider, equipped with its own actual cost function, can apply our algorithm to compute its pricing plans.

Our conjectured cost function model has two basic underlying assumptions: (1) the more total monthly usage a

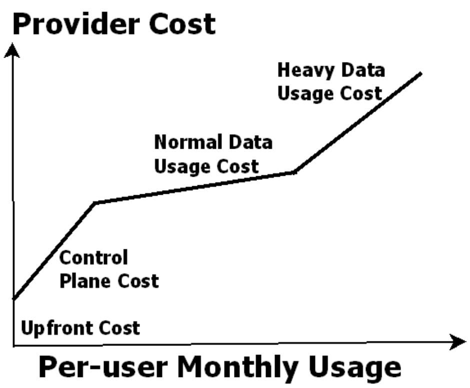

Fig. 2: Conjectured Service Cost Function Model

user has, the more likely the user can cause the network congestion since its average traffic rate is higher; and (2) when the network is congested, the cost to carry an additional byte is higher than that for lightly-loaded condition. When the network load is light (e.g., only a few light users connected to a wireless base station), a small number of additional bytes will not cause other user performance to degrade. In a congested network, additional bytes (especially a large number of bytes) are more likely to cause the network to reach its capacity limit (fundametally limited by wireless spectrum), and thus degrade other users' performance. This may lead to the provider's revenue loss both in the long term (e.g. reputation) and in the short term (e.g., more operational cost to deal with congestion). Therefore, in the context of avoiding an extremely heavy user to degrade the network performance, maximizing a provider's revenue is to some extent aligned with improving the performance of the majority of the users.

With the above assumptions, we thus postulate that cost function has a concave part followed by a convex portion with four linear segments, illustrated in Figure 2. (1) Each user has an upfront cost regardless of his usage, to pay for maintaining his account. (2) There is a control plane cost for setting up and maintaining the data or voice sessions regardless of traffic rate and session duration. (3) Therefore, the cost per byte for low total usage is relatively high. The data plane cost per byte is lower than that of the control plane when the network is not congested. (4) Some extremely heavy users whose traffic is beyond a large threshold tend to constantly have their data sessions on to upload and download data and cause the network and other user's experience to significantly degrade, and with the two assumptions we have above, the cost to carry their additional traffic beyond the threshold is higher than the middle part.

B. Simplifying the cost function

We say that a function $f : \mathcal{U} \mapsto \mathbb{R}$ is approximated by $g : \mathcal{U} \mapsto \mathbb{R}$ to a factor $1+\epsilon$ if $f(x) \leq g(x) \leq (1+\epsilon) f(x)$ for all $x \in \mathcal{U}$. For the following two reasons, we try to approximate the usage cost, as well as the price function (to be introduced later), using a collection of simple functions.

1) As a pricing function, for instance, such functions are easier to describe. As shown in previous studies such as [1], customers look for plans that they can easily understand. Giving them a continuous function for determining the price is neither feasible nor reliable.
2) It is simpler to work on discrete functions. Later in the paper, we use the dynamic programming technique to find the pricing function. Although most of the discussion applies equally well to continuous functions, it is more convenient to approximate the usage function with small error, and use the simpler function in the algorithm.

We consider *threshold functions* as the primitives for this approximation. Let us define $\tau_{(u,p)} : \mathcal{U} \mapsto \mathbb{R}$ such that $\tau_{(u,p)}(x) = p$ if $x \leq u$ and it takes value ∞ otherwise. In terms of pricing functions, threshold functions correspond to simple plans in which a customer pays a fixed price after which he may use up to a fixed amount from the service.

The following theorem shows how to approximate any increasing function via a small number of threshold functions.

Theorem 1. *Any increasing function $f : \mathcal{U} \mapsto [L, R]$ (where $0 < L < R$) can be approximated to a factor $1+\epsilon$ by $f^*(x) = \min_{l=1}^{k} \tau_{(u_i,p_i)}(x)$, where $k = \lceil \log \frac{R}{L} / \log(1+\epsilon) \rceil$.*

In the rest of the paper, we assume that the service cost function ϕ is approximated by a small collection of threshold functions. The error parameter ϵ is less than 0.01.

IV. Pricing Algorithms

In this section, we first formulate the pricing problem as a revenue optimization problem with the consideration of potential customer churn. Note that our problem formulation is based on a variety of assumptions, which may not reflect the actual experience of service providers. As a result, the theoretical solutions provided in this paper may not be the right choices in practice. Thus, we do not claim any optimality of our algorithms. We then consider user behavior changes in the face of new pricing plans. Finally, we present the dynamic programming algorithm to compute the threshold functions for pricing and further approximate the pricing function with a few quasi-linear pricing policies.

A. Formulating Pricing Problem

The service provider announces a *price function* $\pi : \mathcal{U} \mapsto \mathbb{R}$ to its customers. A customer with usage x then pays $\pi(x)$ to the service provider. In general, the price function can be any increasing nonnegative function. However, we look for functions that

- are increasing and nonnegative;
- can be easily explained to the customers; and
- are within a tolerance factor of the service cost function ϕ.

The tolerance parameter $\alpha \geq 1$ is a hypothetical factor to model potential customer churn. We assume that a customer switches to another provider if and only if his charge is more than α times the service cost in the current provider. This is merely a conjecture of customer behavior, and is not based on any real data.

Theorem 1 ensures that any price function can be approximated by a series of threshold functions. Thus we assume the solution is described in this form. In fact, we first propose an algorithm that finds the best such pricing that consists of a relatively small number of threshold functions. Later, we show how to make the pricing even more compact without losing much in the revenue. Each of these threshold functions can be thought of as one available voice/data plan. A fixed usage may be charged differently in different plans since some plans are suitable for heavy users, and some plans are good for light users. Each customer selects a plan that suits his needs best. If it turns out that his need was more than his estimate, as discussed above, it is usually possible to go to the higher plans without paying a penalty. A customer does not have complete flexibility to change his plan every month, but he will move to the right plan pretty soon, hence we assume that each customer is paying the minimum possible price in the long term.

The provider knows the usage function $\sigma : \mathcal{U} \mapsto \mathbb{N}$, the service cost function $\phi : \mathcal{U} \mapsto \mathbb{R}$, and the tolerance parameter $\alpha \geq 1$. It also fixes a parameter M of how many threshold price functions is desirable. The goal is to find such a pricing that maximizes its revenue. In the simplest case (called "passive customers" case), after the price function is fixed, a customer with usage x picks the best plan for herself, and thus pays $\pi(x)$. The revenue is then $\sum_{x\in\mathcal{U}} \sigma(x)[\pi(x) - \phi^*(x)]$, where ϕ^* denotes the actual service cost for the provider. This function is hard to compute although we will provide a conjecture in the next section. For the *passive customers* setting, ϕ^* is irrelevant in the maximization objective since its total contribution is constant if we require not to lose any customers. Then, we can only try to optimize the first portion of the objective, i.e., $\sum_{x\in\mathcal{U}} \sigma(x)\pi(x)$. However, in the more realistic model described below, we need to consider a more general revenue function, and we cannot just focus on the first term.

B. Considering Customers' Response

We have an estimate of the customers' usages based on historical data, denoted by the frequency function $\sigma : \mathcal{U} \mapsto \mathbb{N}$. However, common sense has it that customers' usage behavior changes in the face of new pricing plans. In our formulation, σ and the price function π result in a modified usage function, denoted by $\sigma_\pi : \mathcal{U} \mapsto \mathbb{N}$. In what follows we devise a model for this behavior based on our conjecture (not based on actual data), in the *responding customers* setting. Suppose a customer originally has usage x, and the two closest threshold functions to x in price function π are $\tau_{(u_1,p_1)}$ and $\tau_{(u_2,p_2)}$ such that $u_1 < x \leq u_2$. We say that $\tau_{(u_2,p_2)}$ is the "normal" plan the customer should join. Yet, either of the following situations may occur in practice.

- **Frugality:** If $x \ll u_2$, the customer may feel that his "normal" allowance is more than his needs, he may opt to join the lower plan, $\tau_{(u_1,p_1)}$, and reduce his usage to u_1.
- **Expansiveness:** In case the customer does not switch to the lower plan, though, he may feel that he is paying for a usage u_2 rather than x. As a result he may have a tendency to use more service, say $x' \leq u_2$.

We postulate that a customer with usage x, regardless of her identity, when given two pricing plans with allowances u_1, u_2, behaves like a probability distribution $\delta_{(x,u_1,u_2)}$ over $\{u_1\} \cup [x, u_2]$. We conjecture that

- the distribution has a spike at u_1 and is smooth on the rest of the support;
- $\delta_{(x,u_1,u_2)}(u_1)$ decreases as x increases;
- $\delta_{(x,u_1,u_2)}(u_1)$ increases as u_2 increases; and
- $\delta_{(x,u_1,u_2)}(x')$ decreases as $x' \geq x$ increases.

The general revenue function is thus $\sum_{x\in\mathcal{U}} \sigma_\pi(x)[\pi(x) - \phi^*(x)]$, for which the second term depends on π, and cannot be removed from the optimization problem. The algorithms in the next section can be extended to work for any polynomially computable class of probability distributions $\delta_{(x,u_1,u_2)}$, however, for simplicity in the rest of the paper we focus on one in which each customer joins the lower plan with a fixed probability q. A more extensive analysis of the customers' behavior is deferred to future work.

C. Pricing via dynamic programming

We use a dynamic-programming (DP) technique to find the pricing with only M threshold function components. For $x, x' \in \mathcal{U}$ and $0 \leq \mu \leq M$, define $A[x, x', \mu]$ as the maximum revenue we can obtain with μ price threshold functions from customers with usage less than x such that the last threshold is $p' = \alpha\phi(x')$; notice that this function has the potential of attracting customers with usage from $[x', x)$. We give a recursive formula to compute this quantity for all possible values. Then, the final solution to the instance can be found from $\max_{x'} A[\infty, x', M]$. We assume for simplicity that $\infty \in \mathcal{U}$, and $\sigma(\infty) = 0$.

In order to compute the value of $A[x, x', \mu]$, we guess $x'' \in \mathcal{U}$ that is the low end of the next-to-last interval, i.e., the second-highest price function has value $p'' = \alpha\phi(x'')$. The last two threshold function are $\tau_{(x',p'')}, \tau_{(x,p')}$. The revenue has two components. One component, that accounts for customers with usage less than x', comes from $A[x', x'', \mu-1]$. The other component deals with customers with usage from $[x', x)$. A customer with usage y in this range ends up with usage x' with probability q, joins the lower plan, and pays $\alpha\phi(x'')$ and incurs a cost of $\phi(x')$ on the network. With probability $1-q$, though, this customer does not change her usage, pays $\alpha\phi(x')$, and incurs a cost of $\phi(y)$ on the network. Hence, we earn

$$\sum_{y\in[x',x)} \sigma(y) \left\{ q \left[p'' - \phi(x') \right] + (1-q) \left[p' - \phi(y) \right] \right\} \qquad (1)$$

from the last plan, and the problem reduces to $A[x', x'', \mu-1]$ as previously mentioned.

For simplicity, without loss of generality, we assume that the usage values are nonnegative. Then, we augment $\mathcal{U}$ and σ, so that $\{0\} \in \mathcal{U}$, and $\sigma(0) = 0$. Let $u_{\min} := \min_{x\in\mathcal{U}-\{0\}} x$. Then we do the initialization for the DP at $x = u_{\min}, x' = 0$. We set $A[u_{\min}, 0, \mu] = 0$ for all μ, and use (1) to fill the DP table. Standard techniques allow us to find the corresponding pricing function.

Theorem 2. *There is an algorithm that computes $A[x, x', \mu]$ for all $x, x' \in \mathcal{U}$ and $0 \leq \mu \leq M$; the algorithm runs in time $O(|\mathcal{U}|^3 M)$.*

D. Final pricing

So far we have shown that the price function can be closely approximated using threshold functions, and then the total revenue can be further improved if a sufficiently large number of thresholds are selected. However, in order to present the plans to customers, the pricing policy should be reduced to a few simple functions. Here we approximate the threshold price functions by a piecewise linear function, composed of four linear pieces in the following way.

Define $(x)_+ := \max\{x, 0\}$. Then any piecewise smooth function can be written as $f(x) = a + bx + \sum_{i \geq 0} c_i (x - \theta_i)_+$, where $\forall i : \theta_i \leq \theta_{i+1}$. Note that the starting point of the i^{th} component is θ_i, and its slope is $b + \sum_{j \leq i} c_j$. In order to make the function appropriate for pricing, we impose the following constraints to the family of piecewise linear functions.

- Without loss of generality, we assume $b = 0$ by allowing θ_1 to be zero.
- We restrict ourselves to four-wise linear functions:
$$f(x) = a + c_1(x - \theta_1)_+ + c_2(x - \theta_2)_+ + c_3(x - \theta_3)_+ + c_4(x - \theta_4)_+.$$
- The price function should be positive and increasing, i.e., $a \geq 0$ and $d_1, \ldots, d_4 \geq 0$ where $\forall j : d_j = \sum_{j'=1}^{j} c_j$.
- In order to assure that the price function combines concave and convex pricing functions, we explicitly impose the requirement $c_2 \leq 0, c_3 \leq 0$, and $c_4 \geq 0$.[2]

Therefore, we define the family of *valid* piecewise-smooth functions as

$$\mathcal{F} := \left\{ f(x) = a + \sum_{i=1}^{4} c_i (x - \theta_i)_+ \ : \begin{array}{c} a, c_1, c_4 \geq 0 \\ c_2, c_3 \leq 0 \\ \forall j : d_j \geq 0 \end{array} \right\}.$$

Let $\langle (T_1, Y_1), \ldots, (T_M, Y_M) \rangle$ be the M threshold and price values output by the algorithm. For every $f \in \mathcal{F}$, define the square-loss of f as $\mathcal{L}(f) := \frac{1}{M} \sum_{i=1}^{M} |Y_i - f(T_i)|^2$. Here we use regression [12] to find a function that minimizes the square-loss:

$$\text{minimize}_{f \in \mathcal{F}} \mathcal{L}(f). \tag{2}$$

As the loss function of Eqn. (2) is nonconvex, we use the simulated annealing [15] method to approximate its optimal solution. Simulated annealing is an iterative local search method that repeatedly updates the current solution by a random solution which is sufficiently close to the current solution. The update probability is

$$\min \left\{ 1, \exp \left[-\frac{\mathcal{L}(f_n) - \mathcal{L}(f_0)}{T} \right] \right\},$$

[2]This condition is only for the concave-convex modeling motivated from the existing price functions. In case a new model is proposed, the condition needs to be modified accordingly.

where f_0 is the current solution, f_n is the next solution, and T is a global (temperature) parameter, that is gradually decreased during the process. It has been proved that the simulated annealing algorithm converges to the global optimum of the cost function [16]. In this paper we implemented the algorithm with 10^7 iterations.

V. Conclusions and further work

Finally we summarize our findings.

- The logarithm of (data and voice) usage of the customers can be modeled via a mixture of Gaussian model with a few components.
- The usage, especially for the data download, is highly correlated with the phone make and model since this specifies the type of service the customers demand: whether they only read emails, browse the web, or watch video clips, for instance.
- We seek price functions that are simple to describe, e.g., threshold functions, or piecewise linear functions.
- Customers may change their usage when faced with a new set of plans. We propose a model for this behavior.
- The threshold price function can be found via dynamic programming.
- From the thresholds, we can obtain a much more concise piecewise linear price function.

For future work, we plan to conduct systematic analysis on the impact of potential customer churn.

References

[1] A. Odlyzko, "Internet pricing and the history of communications," *Computer Networks*, pp. 493–517, 2001.

[2] ——, "The economics of the internet: Utility, utilization, pricing, and quality of service," AT&T Labs, Tech. Rep., 1999.

[3] ——, "Privacy, economics, and price discrimination on the internet," in *Int'l conf. Elec. Commerce*, 2003, pp. 355–366.

[4] A. M. Odlyzko, "The internet and 19th century railways," 2003.

[5] ——, "Internet traffic growth: Sources and implications," 2003.

[6] P. Reichl, B. Stiller, and S. Leinen, "Pricing and Cost Recovery for Internet Services: Practical Review, Classification, and Application of Relevant Models," *Netnomics*, vol. 13, no. 2, pp. 149–171, 2001.

[7] J. Wang and K. Li, "Metrics to Evaluate Internet Pricing Schemes," in *Int'l Symp. on Signal Proc. and Information Theory*, 2006, pp. 600–605.

[8] J. K. MacKie-Mason and H. R. Varian, *Pricing the Internet*, 1994. [Online]. Available: http://ideas.repec.org/p/wpa/wuwpco/9401002.html

[9] R. Iyengar, K. Jedidi, and R. Kohli, "A Conjoint Approach to Multi-Part Pricing," http://www.bus.brocku.ca/faculty/documents/twopart_feb5.pdf, 2007.

[10] P. Muller and F. Quintana, "Nonparametric Bayesian Data Analysis," *Stat. Science*, vol. 19, no. 1, pp. 95–110, 2004.

[11] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, pp. 993–1022, 2003.

[12] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009.

[13] D. Blei and M. Jordan, "Variational Inference for Dirichlet process mixtures," *J. Bayesian Statistics*, pp. 121–144, 2006.

[14] D. Mochihashi, "lda, a Latent Dirichlet Allocation Package," http://chasen.org/ daiti-m/dist/lda/, 2009.

[15] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, vol. 220, no. 4598, pp. 671–680, 1983.

[16] V. Granville, M. Krivanek, and J.-P. Rasson, "Simulated annealing: A proof of convergence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 652–656, 1994.