Low Complexity Approximate Bayesian Logistic Regression for Sparse Online Learning
Gil I. Shamir
Google Deepmind
Email: gshamir@google.com

Wojciech Szpankowski
Purdue University
Email: szpan@purdue.edu

Abstract—Theoretical results show that Bayesian methods can achieve lower bounds on regret for online logistic regression. In practice, however, such techniques may not be feasible especially for very large feature sets. Various approximations that, for huge sparse feature sets, diminish the theoretical advantages, must be used. Often, stochastic gradient methods is used with hyper-parameters that must be tuned on some surrogate loss, defeating theoretical advantages of Bayesian methods. The surrogate loss, defined to approximate the mixture, requires techniques as Monte Carlo sampling, increasing computations per example. We propose low complexity algorithm for sparse online logistic and probit regressions that runs linearly in time horizon. Unlike variational inference and other methods, our methods use analytical closed forms, substantially lowering computations. Unlike dense solutions, as Gaussian Mixtures, our methods allow for sparse problems with huge feature sets without increasing complexity. With the analytical closed forms, there is also no need for applying stochastic gradient methods on surrogate losses, and for tuning and balancing learning and regularization hyper-parameters. Empirical results top the performance of the more computationally involved methods.

I. INTRODUCTION

We consider online [8], [44] binary logistic regression over a series of rounds \( t \in \{1, 2, \ldots, T\} \). At round \( t \), a sparse feature vector \( x_t \in [-1, 1]^d \) with \( d_t \ll d \) nonzero values, is revealed, and a prediction for the label \( y_t \in \{-1, 1\} \) must be generated. The dimension \( d \) can be huge (billions), but \( d_t \) is usually tens or hundreds. Logistic regression is used in a huge portion of existing learning problems.

A prediction algorithm attempts to maximize probabilities of the observed labels. Online methods sequentially learn parameters for the \( d \) features. With stochastic gradient methods [7], [14], these are weights \( w_{i,t} \) associated with feature \( i \in \{1, \ldots, d\} \) at round \( t \). Bayesian methods keep track of some distribution over the parameters, and assign an expected mixture probability to the generated prediction [19], [36]. The overall objective is to maximize a sequence likelihood probability, or to minimize its negative logarithm. A benchmark measure of an algorithm’s performance is its regret, the excess loss it attains over an algorithm that uses some fixed comparator values of \( w^* \triangleq (w_1, w_2, \ldots, w_d)^T \) (\( T \) denoting transpose). A comparator \( w^* \) that minimizes the cumulative loss can be picked to measure the regret relative to the best possible comparator in some space of parameter values.

The authors of [22], [15], [47] demonstrated that, in theory, Bayesian methods are capable to achieve regret, logarithmic with the horizon \( T \) and linear with \( d \), that even matches regret lower bounds for \( d = o(T) \). Classical stochastic gradient methods are usually implemented as proper learning algorithms, that determine \( w_t \) prior to observing \( x_t \), and are inferior in the worst-case [17], although, in many cases depending on the data, they can still achieve logarithmic regret [1], [2], [3]. Recent work [21] demonstrated non-Bayesian improper gradient based algorithms with better regret. Unfortunately, superiority of Bayesian methods diminishes by their intractability.

Most Bayesian literature addressed the dense problem, where \( x_t \) consists of mostly nonzero entries for every \( t \), and the dimension \( d \) of the feature space is relatively small. Techniques, like Gaussian Mixtures [18], [34], [33], [37], [38], [48], that may use VB, usually apply matrix computations quadratic in \( d \) on the covariance matrix. In many practical problems, however, a very small feature subset is present in each example. For categorical features, only one of the features in the vector is present at any example. Techniques, useful for the low dimensional dense problem, may thus not be practical.

Paper Contributions: We provide a simple analytical Bayesian method for online sparse logistic and probit regressions with closed form updates. Our results are first to study regret for Bayesian methods that are simple enough to be applied in practice. They provide an example that connects uncertainty and regret, and more broadly the Minimum Description Length (MDL) principle [40], [39], [41], [42], [46], [47]. Empirical results presented in Section IV demonstrate the advantages of our method over computationally involved methods and over other simpler approximations, both by achieving better regret on synthetic data and better loss on real data. As part of the algorithm, uncertainty measures are provided with no added complexity. We specifically demonstrate that it is sufficient to have an approximation focus on the location of the peak of the posterior and its curvature or value, which are most likely to dominate regret, instead of approximating the full posterior, which brings unnecessary complexity missing the real goal of preserving the effects of a good prior. Approximating the posterior may eventually lead to poor generalization and overfitting by focusing on the tails of the posterior. Our approach directly approximates the posterior.

Related Work: The simplest single dimensional online logistic regression problem \((d = 1)\) and \( x_{1,t} = 1, \forall t \) was widely studied. Jefferys’ prior, \( \rho(\theta) \triangleq 1/\left(\pi \sqrt{\theta(1-\theta)}\right) \), is asymptotically optimal [11], [50], [51], [13]. It can be expressed...
in terms of log-odds weights \( w \) as \( \rho(w) = e^{w/2}/[\pi (1 + e^w)] \). Applying a mixture leads to the [24] (KT) add-1/2 estimator \( Q(y_t|y_{t-1}) = \frac{1}{n_{t-1}(y_t) + 0.5}/t \), where \( n_{t-1}(y_t) \) counts occurrences of \( y_t \). We use \( y^t \) to express a sequence from 1 to \( t \). Applying this prior in a Follow The Regularized Leader (FTRL) setting [28] also leads to the KT estimator. This raised the question whether regret optimality generalizes to large dimensions [29], [17] showed that this was not the case for proper methods. Theoretically achievable bounds of Bayesian methods, however, do generalize [22], [15], [47], with large variance Gaussian or uniform prior with diagonal covariance. Peaked priors fail, as for each feature in an example, other features provide a self excluding log-odds prior, that shifts the relation between the overall distribution and the feature weight. While wide priors are good theoretically, because of the intractability of the Bayesian mixture integrals, diagonal approximations that are used unfortunately degrade their effect.

Bayesian methods have been studied extensively for estimating posteriors [4], [26], [43]. There is ample literature researching such techniques in deep networks (see, e.g., [6], [20], [23], [25], [27], [49]). Most of the work focuses on the ultimate posterior after the full training dataset has been visited. One attempts to leverage the uncertainty measurements to aid in inference on unseen examples. Techniques like expectation propagation (EP) [32], [31] (see also [4], [10], [12], [16]) and VB are used to generate estimates of the posterior. In a dense setup, where there is a relatively small number features (or units in a deep network), Gaussian Mixture models can also be learned, where a jointly Gaussian posterior is learned, usually with some kernel that is used to reduce the dimensionality of the parameters. Such methods, however, do not fit the sparse online setup.

II. PRELIMINARIES

For binary logistic regression, the probability of the label given the example and weights, is given by the Sigmoid of the label weighted dot product of the example and weights, that is,

\[
p(y_t|x_t, w) \triangleq \frac{1}{1 + \exp(-y_t x_T^T w)} \triangleq \text{Sigma} \left( y_t x_T^T w \right). \tag{1}
\]

Let \( \rho_1(w) \) be the prior on the weights at round \( t \), where we start by initializing with some \( \rho_1(w) \). We will assume that \( \rho(\cdot) \) is approximated by a diagonal covariance Gaussian, with means \( \mu_{t,i} \) and variances \( \sigma^2_{t,i} \) for component \( i \) at time \( t \). Leveraging results in [22], [15], [47], if we restrict \( w_i \in [-B, B] \), a uniform prior over this interval or a normal prior with standard deviation proportional to \( B \) can be picked. Observing sparse \( x_t \), the prediction for \( y_t \) is given by

\[p_t \triangleq P(y_t|x_t) = \int_w p(y_t|x_t, w)p(w)dw \triangleq \int p_t(y_t, w|x_t)dw. \tag{2}\]

The expected prediction \( p_t \) in (2) marginalizes out the weights \( w \) according to the prior \( \rho_1(\cdot) \) from the joint probability \( w \) and \( y_t \). The prediction \( p_t \) is a function also of all prior pairs sequence \( \{x_{t-1}, y_{t-1}\} \) through the prior \( \rho_1(\cdot) \). After observing \( y_t \), we try to match a (diagonal) posterior \( Q(\cdot) \) to the weights that will equal the next round’s prior

\[p_{t+1}(w) \triangleq Q_t(w) \approx p(w|x^t, y^t) = \frac{p(y_t|x_t, w)p_t(w)}{p_t}. \tag{3}\]

Using \( S_T \triangleq \{x^T, y^T\} \), the logarithmic loss incurred by approximation \( Q(\cdot) \) on the sequence of predictions is

\[L(S_T, Q) \triangleq -\sum_{t=1}^{T} \log p_t. \]

Let \( w^* \) be some fixed comparator in the parameter values’ space. Then, the regret of approximation \( Q(\cdot) \) relative to comparator \( w^* \) is given by

\[R(S_T, Q, w^*) \triangleq L(S_T, Q) - L(S_T, w^*) \]

\[= -\sum_{t=1}^{T} \left( \log p_t + \log(1 + \exp(-y_t x_T^T w^*)) \right). \]

The regret, studied in this paper, measures the excess loss relative to the best possible \( w^* \) comparator.

III. MARGINALIZED BAYESIAN GAUSSIAN

In this section, we describe the proposed method. First, the Sigmoid is approximated by a normal Cumulative Distribution Function (CDF). A prediction for the label of the current example is generated shrinking the cumulative mean score as function of the cumulative variance over all features. The main idea for updating feature distributions is marginalizing away all other covariates for each feature in an example at a given round, such that the mean and variance of the feature can be updated to match the location of the peak and either its curvature or value to the true marginalized posterior. We observe that Probit Regression follows the same steps, except that it does not require the initial approximation so we do not discuss it in details here. It turns out that similar approximation methodology can be used to apply simple multi-dimensional updates instead of a marginalized one, the can be performed when sparsity is limited.

Gaussian Approximation of a Sigmoid: The relation between the logistic distribution and the Normal one was well studied in the statistics literature (see, e.g., [4], [35]). The Sigmoid function in (1) can be viewed as a CDF, which can be approximated by a normal CDF \( \Phi(z) \) (The inverse of \( \Phi(\cdot) \) is the Probit function.) The derivative of the Sigmoid function is the 0-mean Logistic Probability Density Function (PDF). Matching the PDFs, we have \( e^w/(1+e^w)^2 \approx 1/\sqrt{2\pi\sigma^2} \exp \left \{ -w^2/2\sigma^2 \right \} \). This yields that the Sigmoid function can be approximated by a 0-mean Gaussian CDF with variance \( 8/\pi \). Using the standard 0-mean normal \( \Phi(\cdot) \) function, the argument is scaled by the inverse of the standard deviation \( \sqrt{8/\pi} \), giving

\[\text{Sigma}(w) \triangleq \frac{1}{1 + e^{-w}} \approx \Phi \left( \sqrt{\frac{8}{\pi}} \cdot w \right). \tag{5}\]

Approximation approach and some notation: With the diagonal and Gaussian assumptions, for each sparse example (with only \( d_t \ll d \) nonzero entries in \( x_t \)), we can assume that we have a single normal random variable, whose mean is the \( x_t \) weighted mean of covariate weights, and whose variance...
is the quadratically weighted sum of variances. Denote the example total weight, mean, and variance by
\[ w_t \triangleq \sum_{i=1}^{d} x_{i,t} w_{i,t}, \quad \mu_t \triangleq \sum_{i=1}^{d} x_{i,t} \cdot \mu_{i,t}, \quad \sigma^2_t \triangleq \sum_{i=1}^{d} x_{i,t}^2 \cdot \sigma^2_{i,t}. \]

Since we consider a sparse problem, there is benefit to breaking the dependencies between features present in a given example and updating each independently. We can achieve that by marginalizing the prior at \( t \) over all other features. Because we assume all features are jointly independent Gaussians, we can break the joint prior into a product of two components; one, the marginal of the feature, and the other the marginal of all other features together, i.e., the self excluding prior. To match the posterior, we then marginalize on the latter, and match a single dimensional posterior for each feature. The self excluding prior for feature \( i \) at time \( t \), its mean and variance are
\[ w_{-i,t} = \sum_{j \neq i}^{d} x_{j,t} w_{j,t} = \sum_{j \neq i}^{d} x_{j,t} w_{j,t}. \]
\[ \mu_{-i,t} \triangleq \mu_t - x_{i,t} \cdot \mu_{i,t}; \quad \sigma^2_{-i,t} \triangleq \sigma^2_t - x_{i,t}^2 \cdot \sigma^2_{i,t}. \]

**Prediction:** With the probit approximation in (5) and the single dimensional variable \( w_t \), we can compute \( p_t \) in (2), replacing \( p(y_t | x_t, \textbf{w}) \) in (1) by a normal CDF. Approximating this integral (see, e.g., [35], Section 8.4.4.2, and [4]) gives
\[ p_t \approx \text{Sigma} \left( \frac{y_t \cdot \mu_t}{\sqrt{1 + \frac{\pi}{8} \sigma^2_t}} \right). \]

This result demonstrates how the variance shrinks the prediction towards probability 0.5.

**Marginalization:** Given the diagonalization assumption, the prior at \( t \) can be expressed as \( \rho_t(w) = \rho_t(w_i) \cdot \rho_{-i,t}(w_{-i}) \), where \( \rho_{-i,t}(\cdot) \) is the prior on the self excluding prior of \( w_i \). Hence, \( p(y_t, \textbf{w} | x_t) = p(y_t | x_t, \textbf{w}) \rho_{-i,t}(w_i) \rho_{-i,t}(w_{-i}) \).

Marginalizing on \( w_{-i} \) gives
\[ p_t \triangleq \rho_t(w_i) \int_{-\infty}^{\infty} p(y_t | x_t, \textbf{w}) \rho_{-i,t}(w_{-i}) \text{d}w_{-i}. \]

The inner integral \( I_{w_{-i,t}} \), which marginalizes over \( w_{-i} \) with its prior \( \rho_{-i,t}(w_{-i}) \), can be approximated by
\[ I_{w_{-i,t}} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2 \pi \sigma^2_{-i,t}}} \exp \left( \frac{- (w_{-i,t} - \mu_{-i,t})^2}{2 \sigma^2_{-i,t}} \right) \text{Sigma} \left[ y_t \cdot (x_{i,t} w_t + w_{-i}) \right]. \]
\[ = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2 \pi \sigma^2_{-i,t}}} \exp \left( \frac{- (w_{-i,t} - \mu_{-i,t})^2}{2 \sigma^2_{-i,t}} \right) \Phi \left( \frac{\sqrt{\pi}}{8} y_t \cdot (x_{i,t} w_t + w_{-i}) \right). \]
\[ = \Phi \left( \frac{\sqrt{\pi}}{8} y_t \cdot (x_{i,t} w_t + w_{-i}) \right) \approx \text{Sigma} \left( \frac{y_t \cdot (x_{i,t} w_t + w_{-i})}{\sqrt{1 + \frac{\pi}{8} \sigma^2_{-i,t}}} \right). \]

Step (a) follows, again, from the approximation in (5). For (b), we apply the change of variables \( z = (w_{-i} - \mu_{-i,t})/\sigma_{-i,t} \), where \( \phi(\cdot) \) is the standard Gaussian PDF. The integral in (b) gives \( \Phi \left( \frac{x_{i,t} w_t}{\sqrt{\pi} \sigma_{-i,t}} \right) \), with \( a = \sqrt{\pi} y_t \cdot (x_{i,t} w_t) + x_{i,t}^2 \sigma_{i,t} \) and \( b^2 = \frac{\pi}{8} \sigma^2_{-i,t} \) to lead to (c). Finally, the approximation in (5) is used to go back from a Normal CDF to a Sigmoid in (d).

**Posterior:** The posterior on \( w_i \) is given by plugging (10) into (9) normalizing by \( p_t \) in (2) leading to:
\[ \rho_{i,t+1}(w_i) = Q_{i,t}(w_i) \approx p(w_i | x^i, y^i) = \frac{1}{p_t} \cdot \rho_i(w_i). \]

The approximation on the right implies matching the current true posterior with the \( i \)th component of the approximate posterior \( Q(\cdot) \). It can be simplified to
\[ \frac{1}{p_t \cdot \sigma_{i,t}} \exp \left( -\frac{(w_i - \mu_{i,t+1})^2}{2 \sigma^2_{i,t+1}} \right) \approx \frac{1}{p_t \cdot \sigma_{i,t}} \exp \left( -\frac{(w_i - \mu_{i,t})^2}{2 \sigma^2_{i,t}} \right) \cdot \text{Sigma} \left( \frac{y_t (x_{i,t} w_t + x_{i,t} \mu_{i,t})}{\sqrt{1 + \frac{\pi}{8} \sigma^2_{i,t}}} \right). \]

**Algorithm GAUSS:** Because the functional form of the posterior is not Gaussian, there are multiple ways to fit a Gaussian. However, we want to ensure that the regions of the true posterior are not scaled down too much, as this will incur additional loss. It is thus desirable to match the peak of the true posterior with the peak of the approximation. One method is to match both the location and height of the peak. The other, Laplace approximation [4], matches the location and curvature at the peak. Both methods give the same approximate for \( \mu_{i,t+1} \), but a somewhat different one for \( \sigma^2_{i,t+1} \).

To give \( \mu_{i,t+1} \), we find \( w_i \) that maximizes the r.h.s. of (12), or minimizes its negative logarithm. Let
\[ p_{i,t+1} \triangleq \text{Sigma} \left( \frac{y_t (x_{i,t} w_t + x_{i,t} \mu_{i,t+1})}{\sqrt{1 + \frac{\pi}{8} \sigma^2_{i,t}}} \right) \]
\[ = \left[ 1 + \exp \left( -\frac{y_t (x_{i,t} w_t + x_{i,t} \mu_{i,t+1})}{\sqrt{1 + \frac{\pi}{8} \sigma^2_{i,t}}} \right) \right]^{-1}. \]

Thus \( p_{i,t+1} \) is the probability predicted for \( y_t \) if we update \( \mu_{i,t} \) and shrink as function of \( \sigma^2_{i,t} \). The minimization of (12) gives
\[ \mu_{i,t+1} = \arg \max_{w_i} \left\{ \frac{1}{p_t \cdot \sigma_{i,t}} \exp \left[ -\frac{(w_i - \mu_{i,t})^2}{2 \sigma^2_{i,t}} \right] \right\}. \]

Authorized license use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
which leads, after some algebra, to
\[ \mu_{i,t+1} = \mu_{i,t} + \frac{y_t x_i, t \sigma_{i,t}^2}{1 + \frac{n}{8} \sigma_{i,t}^2} \cdot (1 - p_{i,t+1}). \]

Eq. (14) can be solved iteratively, where Newton’s method can be used. The solution for \( \mu_{i,t+1} \) can be expressed in terms of the \( r \) generalized Lambert W function [30].

Alternatively, to avoid multiple iterations per update when using Newton’s method, we can use a Taylor series approximation of \( 1 - p_{i,t+1} \) around \( 1 - p_{i,t} \), where \( p_{i,t} \) is defined as \( p_t \) in (8) replacing \( \sigma_{i,t}^2 \) by \( \sigma_{i,t}^2 \). Notice that \( p_{i,t} \) is the probability of \( y_t \) as projected by the means of the weights at \( t \), shrunk as function of \( \sigma_{i,t}^2 \) instead of \( \sigma_{i,t}^2 \). More importantly, it depends only on parameters before the update at \( t + 1 \) is applied, giving the following closed form solution
\[ \mu_{i,t+1} = \mu_{i,t} + \frac{y_t x_i, t \sigma_i^2}{1 + \frac{n}{8} \sigma_i^2} \cdot (1 - p_{i,t}) \cdot \left[ 1 + \frac{1}{1 + \frac{n}{8} \sigma_i^2} y_t^2 x_i, t \sigma_i^2 (1 - p_{i,t}) \right]. \]

It may be simpler to store the precision \( 1/\sigma_{i,t}^2 \), in which case, (15) may be easier to compute by normalizing both numerator and denominator by \( \sigma_{i,t}^2 \), applying this normalization on the right term of the denominator. Second or higher orders approximations can also be applied, but may not be necessary, as the first order one already gives identical performance.

After updating \( \mu_{i,t+1} \), we can apply (12) to update \( \sigma_{i,t+1} \). Plugging (14) in (12), we solve for \( \sigma_{i,t+1} \),
\[ \sigma_{i,t+1} = \frac{p_t \sigma_{i,t}}{p_{i,t+1}} \exp \left\{ \frac{(\mu_{i,t+1} - \mu_{i,t})^2}{2 \sigma_i^2} \right\} \]
\[ = \frac{p_t \sigma_{i,t}}{p_{i,t+1}} \exp \left\{ \frac{y_t^2 x_i, t \sigma_i^2}{2 (1 + \frac{n}{8} \sigma_i^2)} \cdot (1 - p_{i,t+1}) \right\}. \]

Alternatively to (16), Laplace derivative of the negative logarithm of the posterior, leading to
\[ \sigma_{i,t+1}^2 = \left[ \frac{1}{\sigma_{i,t}^2} + \frac{y_t^2 x_i, t \sigma_i^2}{1 + \frac{n}{8} \sigma_i^2} \cdot p_{i,t+1} \cdot (1 - p_{i,t+1}) \right]^{-1}. \]

The procedures described are summarized in Algorithm 1 that we call GAUSS. Its running time is \( O(d_t T) \).

**Regret.** The expected logarithmic regret as in (4) can be normalized to give \( r_T \triangleq \mathbb{E}_T / \log T \). It can also be demonstrated as time series behavior using the subsequence normalized \( r_t \triangleq \mathbb{E}_t / \log t \). From [47], [45], we have small \( \epsilon > 0 \),
\[ r_T \triangleq \frac{\mathbb{E}_T}{\log T} \geq (1 - \epsilon) \frac{d}{2}. \]

As simulations in the next section demonstrate, with proper choice of prior, the GAUSS algorithm empirically matches the lower bound with \( r_T \sim d/2 \). However, we leave as an open to prove it theoretically.

### Algorithm 1 GAUSS: Marginalized Bayesian Gaussian Approximation

1. **procedure** MARGINALIZED BAYESIAN GAUSSIAN APPROXIMATION(Parameters: \( \mu_0, \sigma_0^2 \))
2. \[ \forall t \in 1, \ldots, d; \mu_{i,t} \leftarrow \mu_0, \sigma_{i,t}^2 \leftarrow \sigma_0^2. \]
3. **for** \( t = 1, 2, \ldots, T \) **do**
4. \[ \text{Get } x_t. \]
5. \[ \text{Compute } \mu_t, \sigma_t^2 \text{ with (6)}. \]
6. \[ \text{Generate } p_t \text{ for } y_t \in \{-1, 1\} \text{ with (8)}. \]
7. \[ \text{Observe } y_t. \]
8. **for** \( i : x_{i,t} \neq 0 \) **do**
9. \[ \text{Compute } p_{i,t} \text{ with modified (8), and } p_{i,t+1} \text{ with (13) using } \mu_{i,t+1} = \mu_{i,t} \text{ for (13)}. \]
10. **iterate** on (14) and (13) with Newton’s method, or use (15) to update \( \mu_{i,t+1} \).
11. **Update** \( \sigma_{i,t+1}^2 \) with either (16) or (17)
12. **end for**
13. **end for**
14. **end procedure**

### IV. Numerical Results

To benchmark regret of an algorithm, one needs the ground truth of real or loss minimizing parameters. On real data, the loss minimizing parameters are unknown. Furthermore, true benchmark datasets can consist of non-stationary data, and the feature sets selected by a model one trains may misspecify the “true” features of such a model. Thus real data may not give clean evaluation of the proposed methods. We present results on a benchmark dataset, but to measure regret performance of different algorithms, we present results on synthetic data. We emphasize that in all simulations we observed Algorithm 1:GAUSS consistently gives regret close to the lower bound \( d/2 \log T \). The other methods fail to do so consistently in all conditions.

---

**Fig. 1:** \( \mathbb{E}_t / \log t \) vs. round \( t \) for different methods for randomly drawn \( d = 200 \) binary features, with \( d_i = 20 \) and standard deviation 1 of true log-odds.
**Synthetic Data:** We simulated data by setting $d$ features with true log-odds weights that were drawn randomly with some prior (or with multiple priors, each governing a subset of the features). At round $t$, $d_t \ll d$ features were selected in random from the set of $d$ features, where different rules were used for different simulations to draw the $d_t$ features. In the fully random case, we set a random fraction $\alpha = d_t/d$ parameter, and each feature was activated with probability $\alpha$. We either used binary features $x_{t,i} \in \{0, 1\}$, or also drew $x_{t,i}$ randomly in $[0, 1]$. For categorical features, the $d$ features were partitioned into categories, and for every example, one or a preset number of features from each category were randomly selected, with different types of randomness, including fast decaying long tail distributions. Let $\theta$ be the vector of true parameters, then, $\Pr(Y_t = 1) = \Sigma(x_t^T \theta)$. We used Algorithm 1, and other algorithms, to sequentially predict $y_t$ and update posterior parameters. For gradient methods, we used *Stochastic Gradient Descent (SGD)* with AdaGrad scheduling [14].

We ran grids of algorithm hyper-parameters for all algorithms to find optimal ones, and we show results for these optimal hyper-parameters for all algorithms. Since we know the true weights $\theta$, we use them for a comparator baseline $w^*$. Curves show *progressive validation* [5] regret. At round $t$ we measure the cumulative regret up to $t$ given by $R_t \triangleq \sum_{t=1}^{t} \left[ \log p_r + \log(1 + \exp(-y_t x_t^T w^*)) \right]$ and plot $R_t/\log t$. If an algorithm has logarithmic regret, the normalized curve of the algorithm will converge to the constant. This methodology thus allows us to observe whether an algorithm has logarithmic regret or not. Results are shown for Algorithm 1 with its different variations (labeled by Gauss for updates with (14), and by Gauss Approx with (15)). Labels also designate the prior used ($\sigma^2_0$ and $\mu_0$), and which variance approximation was used (G for (16) and L for (17)). Reference results are shown for SGD, multi-dimensional Gaussian approximation update (DimGauss); using *Assumed Density Filtering (ADF)* [32], and marginalized VB (VBAprox).

![Fig. 2: Loss relative to the best SGD on the Criteo benchmark dataset for multiple algorithms.](image)

**Criteo display advertising challenge benchmark Dataset:**

The graph in Fig. 2 gives relative percent aggregate loss performance of the Bayesian algorithms relative to the best configuration we found for an AdaGrad SGD on the Criteo dataset. We trained all algorithms on the over 45M examples in this dataset, which consists of 13 integer valued features, and 26 categorical features with different category counts. For each example, we generated a prediction, computed its log loss on the label, and applied update. The aggregate log loss is a sum of data uncertainty and regret. The first is for all algorithms and linear in the size of the data, where the second is sub-linear in $T$ for a good algorithm. Since in real data, there is no prior knowledge of the true parameters, the measured loss does not distinguish between the two terms. Because the regret, which differentiates between the different algorithms is sub-linear, while the data uncertainty component which is linear in $T$, even small noticeable percent improvements imply possible substantial improvements of regret. Algorithm 1 (using a linear model) achieved 0.465 progressive validation log loss, which is better, for example, than results reported using deep networks in [9]. We observe advantages to the Bayesian methods over SGD, where Algorithm 1 was superior to all methods.

**V. Conclusions**

We introduced a simple Bayesian mixture diagonal Gaussian approximation method based on marginalization for sparse online logistic regression, that attempts to retain the affects of a good prior around the optimal values of the weights. The method does not require the complexities of standard Bayesian methods, as VB, but was empirically shown to achieve regret rates as good and even better.

**Acknowledgement.** This work was partially supported by the NSF Center for Science of Information (CSoI) Grant CCF-0939370, and in addition by NSF Grants CCF-1524312, CCF-2006440, and CCF-2211423.

[1] https://www.kaggle.com/c/criteo-display-ad-challenge
REFERENCES

[1] F. Bach. Self-concordant analysis for logistic regression. Electronic Journal of Statistics, 4:384–414, 2010.
[2] F. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. Journal of Machine Learning Research, 15(1):595–627, 2014.
[3] F. Bach and E. Moulines. Non-strongly-convex stochastic approximation with convergence rate $o(1/n)$. In Advances in neural information processing systems, pages 773–781, 2013.
[4] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, New York, 2006.
[5] A. Blum, A. Kalai, and J. Langford. Beating the hold-out: bounds for k-fold and progressive cross-validation. In COLT, 1999.
[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424, 2015.
[7] L. Bottou. Large-scale machine learning with stochastic gradient descent. In COMPSTAT, 2010.
[8] Léon Bottou. Online learning and stochastic approximations. On-line learning in neural networks, 17(9):142, 1998.
[9] Heng-Tze Cheng, Levent Koc, Jeremiah Hanna, Tal Shaked, Tushar Chandra, Harshi Aradhya, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Isip, et al. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems, pages 7–10, 2016.
[10] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 195–203, 2011.
[11] Bertrand S Clarke and Andrew R Barron. Jeffreys’ prior is asymptotically least favorable under entropy risk. Journal of Statistical planning and Inference, 41(1):37–60, 1994.
[12] John P Cunningham, Philipp Hennig, and Simon Lacoste-Julien. Path integrals for online Gaussian learning. In Advances in neural information processing systems, pages 195–203, 2016.
[13] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Linear regression with column-stable kernels. In Proceedings of Machine Learning Research, volume 44 of Proceedings of the 27th Conference on Learning Theory, COLT 2014, page 197–209. MIT press, 2014.
[14] Ralf Herbrich, Neil D Lawrence, and Matthias Seeger. Fast sparse gaussian process methods: The informative vector machine. In Advances in neural information processing systems, pages 625–632, 2003.
[15] Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent dirichlet allocation. In Advances in Neural Information Processing Systems 23, pages 856–864. Curran Associates, Inc., 2010.
[16] Seong Jae Hwang, Ronak R Mehta, Hyunwoo J Kim, Sterling C Johnson, and Vikas Singh. Sampling-free uncertainty estimation in gated recurrent units with applications to normative modeling in neuroimaging. In Uncertainty in Artificial Intelligence, pages 809–819. PMLR, 2020.
[17] Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. Efficient improper learning for online logistic regression. In COLT - Conference on Learning Theory, volume 125 of Proceedings of Machine Learning Research, pages 2085–2108. PMLR, 09–12 Jul 2020.
[18] S. M. Kakade and A. Y. Ng. Online bounds for bayesian algorithms. In NIPS, pages 641–648, 2005.
[19] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In Advances in neural information processing systems, pages 5574–5584, 2017.
[20] R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. IEEE Trans. Inf. Theory, IT-27(2):199–207, Mar. 1981.
[21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems, pages 6402–6413, 2017.
[22] David Makowski, Daniel Wallach, and Marie Tremblay. Using a Bayesian approach to parameter estimation; comparison of the glue and mcmc methods. Agranomie, 22(2):191–203, 2002.
[23] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In Advances in Neural Information Processing Systems, pages 7047–7058, 2018.
[24] H. B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In AISTATS, 2011.
[25] H. B. McMahan and M. J. Streeter. Open problem: Better bounds for online logistic regression. In Journal of Machine Learning Research-Proceedings Track, 23, 2012.
[26] I. Mezo and A. Baricz. On the generalization of the lambert w function. arXiv:1408.5999v2, 2015.
[27] Thomas P Minka. Expectation propagation for approximate Bayesian inference. arXiv preprint arXiv:1301.2294, 2013.
[28] Thomas Peter Minka. A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology, 2001.
[29] Lucie Montuelle, Erwan Le Penne, and Serge Cohen. Gaussian mixture regression model with logistic weights, a penalized maximum likelihood approach. Electronic Journal of Statistics, 8(1):1661–1695, 2014.
[30] Lucie Montuelle, Erwan Le Pennec, and Serge Cohen. Gaussian mixture regression model with logistic weights, a penalized maximum likelihood approach. arXiv preprint arXiv:1304.2696, 2013.
[31] K. P. Murphy. Machine Learning A Probabilistic Perspective. MIT Press, Cambridge, Massachusetts, 2012.
[32] Manfred Opper and Ole Winther. A Bayesian approach to on-line learning. On-line learning in neural networks, pages 363–378, 1998.
[33] Carl Edward Rasmussen. The infinite gaussian mixture model. In Advances in neural information processing systems, pages 554–560, 2000.
[34] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. Digital signal processing, 10(1-3):19–41, 2000.
[35] J. Rissanen. Minimax codes for finite alphabets. EEE Trans. Inform. Theory, IT-24(3):389–392, 1978.
[36] J. Rissanen. Modeling by the shortest data description. Automatica, 14:465–471, 1978.
[37] J. Rissanen. Universal coding, information, prediction, and estimation. IEEE Trans. Inform. Theory, IT-30(4):629–636, Jul. 1984.
[38] J. Rissanen. Stochastic complexity and modeling. The Annals of Statistics, 14:1080–1100, 1986.
[39] Minral K Sen and Paul I. Stoffa. Bayesian inference, gibbs’ sampler and uncertainty estimation in geophysical inversion 1. Geophysical Prospecting, 44(2):313–350, 1996.
[40] Shai Shalev-Shwartz et al. Online learning and online convex optimization. Foundations and trends in Machine Learning, 4(2):107–194, 2011.
[41] G. I. Shamir and W. Szpankowski. A lower bound for regret in logistic regression. In International Symposium on Information Theory, 2021.
[42] Gil I. Shamir. Minimum description length (MDL) regularization for online learning. In Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015, volume 44 of Proceedings of Machine Learning Research, pages 260–276, Montreal, Canada, 11 Dec 2015. PMLR.
[43] Gil I. Shamir. Logistic regression regret: What’s the catch? In COLT - Conference on Learning Theory, volume 125 of Proceedings of Machine Learning Research, pages 3296–3319, 2020.
[44] Hsi Guang Sung. Gaussian mixture regression and classification. PhD thesis, 2004.
[45] Andrew Gordon Wilson. The case for Bayesian deep learning. arXiv preprint arXiv:2001.10905, 2020.
[46] Q. Xie and A. Barron. Minimax redundancy for the class of memoryless sources. IEEE Trans. Information Theory, pages 647–657, 1997.
[47] Q. Xie and A. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. IEEE Trans. Information Theory, 46:431–445, 2000.