Big Learning with Little RAM

D. Sculley
Google, Inc.
dsculley@google.com

Daniel Golovin
Google, Inc.
dgg@google.com

Michael Young
Google, Inc.
mwyoung@google.com

Abstract

In large-scale machine learning, available memory (RAM) is often a key constraint, both during model training and when making new predictions. In this paper, we reduce memory cost by projecting our weight vector $\beta \in \mathbb{R}^d$ onto a coarse discrete set using randomized rounding. Because the values of the discrete set can be stored more compactly than standard 32-bit float encodings, this reduces RAM usage by 50% during training and by up 90% at prediction time. Theoretical analysis provides safety guarantees that bound the regret added by this projection. Empirical evaluation confirms excellent results in practice, adding only an additional 0.01% to logistic loss in testing.

1 Introduction

As the size of large-scale learning continues to accelerate, available machine memory (RAM) is an increasingly important constraint. Bigger data enables performance gains from bigger models, which in turn increases the memory required for model training. Also, in real-world prediction systems trained models are often replicated to multiple machines, making memory cost a key consideration at prediction time.

Scenarios like learning large-scale linear models for predicting ad click through rates (CTR) for sponsored search [12, 6, 1, 13] or filtering email spam at scale [8] are motivating examples of this setting. Learning in such domains may involve peta-scale data sets and result in models with a very large number of coefficient values. The number of bits used to store each value is thus a critical factor the overall memory cost of the system.

Efficient learning at peta-scale is commonly achieved by stochastic gradient descent (SGD) [3] in which millions or billions of tiny steps are accumulated in a weight vector $\beta \in \mathbb{R}^d$. Standard implementations use double-precision or single-precision floating point representations to store coefficient values, requiring 64 or 32 bits per value respectively. These provide numerical precision levels fine-grained enough to accumulate SGD steps without significant roundoff error, but have a dynamic range that far exceeds the needs of practical machine learning (see Figure 1).

In this paper, we limit the number of bits used to represent coefficient values by projecting our weight vector $\beta \in \mathbb{R}^d$ onto a discrete set $(\epsilon\mathbb{Z})^d$, which is essentially a coarse uniformly-spaced grid. This coarse grid spacing does *not* provide enough resolution to accumulate infinitesimal SGD steps without error; however, we are able to work with this space by adopting a randomized rounding scheme as part of this projection. With this strategy, we can store coefficient values with as little as 16 bits in training. At prediction time we can use as few as 9 bits, and these values can be further compressed down to a theoretical limit below 2 bits per value.

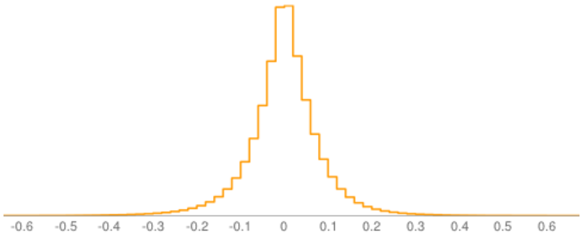

Figure 1: Histogram of coefficient values in a typical large-scale model. Values are tightly grouped near zero, so a large dynamic range is superfluous.

Using analysis similar to Zinkevich's analysis of projected online gradient descent [17], we prove bounds on the changes in model accuracy of these schemes as compared to a hypothetical infinite-precision scheme. This provides a useful safety guarantee. Empirical results show that these methods give excellent performance both during training and testing, with large RAM savings and negligible added error.

2 Related Work: Smaller Models

In this paper, we are primarily concerned with generalized linear models, which may be learned at large-scale using SGD [3] and the closely related projected-gradient descent [17]. These online learning methods have proven remarkably effective at peta-scale learning in recent years, and are available even in open-source packages.

A classic approach to reducing memory usage for linear models has been to encourage sparsity, through popular methods such as the Lasso [14] and L1-regularization [7, 9, 10]. A more recent trend has been to reduce memory usage via the use of feature hashing to control model size [16]. Both families of approaches are effective. The coarse encoding schemes reported here may be used in conjunction with either (or both) of these prior methods to give further reductions in memory usage. An alternate strategy for model compression was explored by Buciluǎ *et al.* who effectively compressed a large classifier ensemble by training a more compact neural-net on unlabeled data that was labeled by the ensemble [4].

Randomized rounding schemes have been widely used in numerical computing and related proofs [11]. Recently, the related technique of randomized counting has been used for developing compact language models [15]. To our knowledge, this paper gives the first formal regret-based analysis of randomized rounding with coarse discrete sets, and provides the first application of these methods for large-scale prediction problems.

3 Preliminaries

First we set up some notation. For concreteness, we focus on linear logistic regression with binary features and labels. The linear model has coefficients $\beta \in \mathbb{R}^d$, so that the prediction given (binary) feature vector $x \in \{0,1\}^d$ is $\sigma(\beta \cdot x)$, where $\sigma(z) := 1/(1+e^{-z})$ is the familiar logistic function. Let $p_\beta(x) := \sigma(\beta \cdot x)$. Unless otherwise stated, the loss function we consider is logistic–loss, *i.e.*, given a labeled example $(x, y) \in \{0,1\}^d \times \{0,1\}$, the logistic–loss is

$$\mathcal{L}(x, y, \beta) := -y \log(p_\beta(x)) - (1-y)\log(1 - p_\beta(x))$$

where we interpret $0 \log 0$ as zero. Here, we take $\log$ to be the natural logarithm for concreteness, though of course any other base just multiplies the loss by a constant factor. We define $||x||_p$ as the ℓ_p norm of a vector x; when the subscript p is omitted, the ℓ_2 norm is implied. For a set $F \subseteq \mathbb{R}^d$, we denote its diameter by $||F|| = \sup_{x,y \in F} ||x - y||$.

4 Projection to Discrete Set During Training

Similar to Zinkevich's paper on projected gradient descent [17], we modify SGD by include a projection of β to a discrete set $(\epsilon\mathbb{Z})^d$ after every update; randomized rounding is part of this projection. Essentially, we compute the result of the SGD update using double-precision values that use 64 bits per value, and then store the result in a much coarser representation using many fewer bits.

4.1 Fixed-Point Encoding as a Discrete Set

One convenient representation for this discrete set is a fixed-point representation using the popular `Qn.m` encoding. In this encoding, we use n bits for the integral part of the value, and m bits for the fractional part. Adding in one bit for the sign of the value results in a total of $n + m + 1$ bits per value.

To encode a coefficient value v into a `Qn.m` format value, we round it to the nearest multiple of $\epsilon = 2^{-m}$ using an unbiased randomized rounding scheme. Let $r = \lfloor v/\epsilon \rfloor$, and $s = v - r\epsilon$. Then round v to $r\epsilon$ with probability $1 - s/\epsilon$ and to $(r + 1)\epsilon$ with probability s/ϵ. The values are then stored as integers by multiplying the rounded value by $1/\epsilon$. Values too large or too small to encode in `Qn.m` are replaced with the maximum or minimum representable value, respectively. Decoding is simple; merely multiply by ϵ.

In practice, we only need to encode and decode those coefficients that are impacted by an update; when input vectors are sparse this is extremely efficient.

4.2 Theoretical Guarantees

We use some terminology from Zinkevich [17], as our analyses build on his. Let T be the number of rounds, $\{\eta_t\}_{t=0}^{T} \geq 0$ be the learning rate schedule, c^t be the convex cost function in round t, k be an upper bound on $||x||_0$, $||\nabla c||$ be the maximum gradient norm over the cost functions, and F be the feasible region. We fix a small constant $\epsilon > 0$. Unlike Zinkevich's paper, we use $\hat{\beta}^t$ for our choice in round t (Zinkevich uses x^t). Also, here $\hat{\beta}$ is the randomized rounding of β. We assume the algorithm proceeds by computing $\beta^{t+1} = \mathrm{Proj}\left(\hat{\beta}^t - \eta_t g^t\right)$ and then (probabilistically) rounding β^{t+1} to a grid point $\hat{\beta}^{t+1} \in (\epsilon\mathbb{Z})^d$, where $\mathrm{Proj}\,(\cdot)$ projects its input onto the feasible region F. We further assume that the randomized rounding procedure does not change the value of any coordinate by more than ϵ in either direction.

We prove the following result (full proof given in the Appendix).

Theorem 4.1. *For online convex optimization, projected online gradient descent with randomized rounding onto a grid of precision ϵ has a regret bound of*

$$\mathbf{E}[\mathrm{regret}\,(T)] \leq \frac{||F||^2}{2\eta_T} + \frac{||\nabla c||^2}{2}\sum_t \eta_t + 2\,\epsilon\,||\nabla c||_1\,T$$

where the expectation is over the internal randomness of the algorithm, under the following assumptions: the randomized rounding is unbiased, and the feasible region contains all grid points within an $[-\epsilon, \epsilon]^d$ box around every β^t. Using $\eta_t = 1/\sqrt{t}$ yields a regret bound of $\mathbf{E}[\mathrm{regret}(T)] = O\left(\sqrt{T}\left(||F||^2 + ||\nabla c||^2\right) + \epsilon\,||\nabla c||_1\,T\right)$.

Thus, constraining ourselves to use grid points increases the regret of projected online gradient descent by an additive $2\,\epsilon\,||\nabla c||_1\,T$ factor. Note the linear dependence on T is unavoidable, and is optimal up to constant factors: Consider a one dimensional instance with $F = [0, \epsilon]$, and $c^t(\beta) = |\beta - \epsilon/2|$ for all t. Then every algorithm constrained to play grid points (in this case, 0 or ϵ), incurs regret $\epsilon T/2 = \epsilon\,||\nabla c||_1 T/2$. Nevertheless, when ϵ is small and input vectors are sparse, the additional regret is relatively small.

4.3 Experimental Results

We simulated the effect of training with projections to a discrete set with `Qn.m` encoding, using data drawn from real-world data sets for ad CTR prediction. For these experiments, we used a

Encoding	AucLoss	LogisticLoss	Bits / Value
q2.10	+0.56%	+0.21%	13
q2.11	+0.23%	+0.09%	14
q2.12	+0.09%	+0.03%	15
q2.13	+0.04%	+0.01%	16
q2.14	+0.02%	+0.01%	17

Table 1: **Effect on Predictive Performance Using Rounding in Training.** Results for the given fixed-point encodings all use randomized rounding, and are reported relative to a control model that uses 32-bit single-precision `float`'s. Values of 0.0 are ideal and higher values show more loss. Changes in loss metrics for q2.13 and q2.14 are negligible, and use only half the memory. The q2.13 encoding is preferred since it can be stored in exactly two bytes.

linear logistic regression trained with stochastic gradient descent, similar to the experimental setup described in [13].

We report changes in LogisticLoss and AucLoss (or, 1 - AUC) relative to a control model that used 32-bit single-precision floats. (We also tested 64-bit double-precision values and found no difference.) In this case, a value of 0.0 would be ideal, showing no detriment to predictive accuracy despite the memory savings, and values greater than zero are progressively less desirable. Metrics are computed using progressive validation [2] that is standard for online learning scenarios: on each round a prediction is made for a given example, loss metrics are computed, and only after that is the model allowed to train on that example. The data set contains tens of millions of examples.

The results, given in Table 1, show that the metrics for LogisticLoss and AucLoss do not substantially degrade with q2.14 or q2.13 encoding. (We also tested these encodings without randomized rounding, and found that removing this step indeed worsens performance significantly.)

The q2.13 encoding is preferred because it uses exactly two bytes and is cleanly implementable with primitive types. This saves a full 50% of RAM for storage of values compared to naively using 32-bit floats at minimal cost to predictive performance.

5 Encoding During Prediction Time

Many real-world problems require large-scale *prediction* as well as large-scale training. This may require that a trained model replicated to several machines [4]. In such a setting, it may make sense to train the model using high precision coefficients, and then compress the model by rounding coefficients before replication.

Unlike in training, we do not need to be concerned with the effects of accumulated roundoff error. This allows us to adopt even more aggressive parameters for rounding at prediction time. This intuition is first demonstrated theoretically. Then we go on to show how we can take advantage of the distribution over values to create even more efficient encodings, with only a few bits per value.

5.1 Theoretical Guarantees

Consider a trained model β and round it in some manner to some $\hat{\beta}$. How can we bound the resulting degradation in loss? First, we start to bound the effect on logistic–loss $\mathcal{L}(\cdot)$ in terms of the quantity $|\beta \cdot x - \hat{\beta} \cdot x|$. Any proofs omitted from the main text can be found in the Appendix.

Lemma 5.1 (Additive Error)**.** *Fix* $\beta, \hat{\beta}$ *and* (x, y)*. Let* $\delta = |\beta \cdot x - \hat{\beta} \cdot x|$*. Then the logistic–loss satisfies*

$$\mathcal{L}\left(x, y, \hat{\beta}\right) - \mathcal{L}(x, y, \beta) \leq \delta.$$

Proof. It is well known that $\left|\frac{\partial \mathcal{L}(x,y,\beta)}{\partial \beta_i}\right| \leq 1$ for all x, y, β and i, which implies the result. □

Interestingly, the relative error in logistic–loss is also bounded.

ENCODING	AucLoss	LogisticLoss	Optimal Bits/Value
Q2.3	+5.72%	+2.55%	0.1
Q2.5	+0.44%	+0.18%	0.5
Q2.7	+0.03%	+0.01%	1.5
Q2.9	+0.00%	+0.00%	3.3
Q2.11	+0.00%	+0.00%	5.6

Table 2: **Effect on Predictive Performance Using Rounding at Prediction Time.** Results for the given fixed-point encodings all use randomized rounding, and are reported relative to a control model that uses 32-bit single-precision `float`'s. Values of 0.0 are ideal and higher values show more loss. Changes in loss metrics are negligible even for q2.7 encoding, and allow values to be encoded with less than two bits per value with theoretically optimal encoding.

Lemma 5.2 (Relative Error). *Fix $\beta, \hat{\beta}$ and $(x, y) \in \{0,1\}^d \times \{0,1\}$. Let $\delta = |\beta \cdot x - \hat{\beta} \cdot x|$ and suppose $\delta < 1$. Then*

$$\frac{\mathcal{L}\left(x, y, \hat{\beta}\right) - \mathcal{L}\left(x, y, \beta\right)}{\mathcal{L}\left(x, y, \beta\right)} \leq \frac{\delta}{1-\delta}.$$

Now, suppose we are using fixed precision numbers to store our model coefficients. such as the Qn.m encoding described earlier, with a precision of ϵ. This induces a grid of feasible model coefficient vectors. If we randomly round each coefficient β_i independently up or down to the nearest feasible value $\hat{\beta}_i$, such that $\mathbf{E}\left[\hat{\beta}_i\right] = \beta_i$, then for any $x \in \{0,1\}^d$ our predicted log-odds ratio, $\hat{\beta} \cdot x$ is distributed as a sum of independent random variables $\{\hat{\beta}_i : x_i = 1\}$. Using standard large-deviation bounds, such as Theorem A.1, we can obtain the following bound:

Lemma 5.3. *Let $\hat{\beta}$ be a model obtained from β using unbiased randomized rounding to a precision ϵ grid. Fix x, and let $Z = \hat{\beta} \cdot x$ be the random predicted log-odds ratio. Then*

$$\mathbf{Pr}[|Z - \beta \cdot x| > t] \leq 2 \exp\left(\frac{-t^2}{2\epsilon^2 ||x||_0}\right)$$

Combining this result with Lemma 5.2 and integrating over t, we can obtain the following result:

Theorem 5.4. *Let $\hat{\beta}$ be a model obtained from β using unbiased randomized rounding to a precision ϵ grid as described above. Then the expected logistic–loss additive error of $\hat{\beta}$ is at most $\epsilon \cdot ||x||_1$ on input x. Furthermore, if $||x||_0 \leq k$ for all input vectors x and $\epsilon \leq 1/2k$, then the expected logistic–loss relative error of $\hat{\beta}$ is at most $\epsilon \cdot 4\sqrt{2\pi k}$.*

5.2 Compressing Values Further

Figure 1 reveals that coefficient values are not uniformly distributed. Storing these values in a fixed-point representation means that individual values will occur many times. Thus, basic information theory tells us that we can compress the overall model by encoding more common values with fewer bits. The theoretical bound for a model with d coefficients is $\frac{-\sum_{i=1}^{d} \log p(\beta_i)}{d}$ bits per value, where $p(v)$ is the probability of occurrence of v in β across all dimensions d. At prediction time, we can approach this limit using variable length encoding schemes and achieve further RAM savings.

5.3 Experimental Results

Here, we present results similar to those in Section 4, but which only use randomized rounding at prediction time rather than in training. We compare loss metrics against a baseline of un-rounded coefficients. We also show the number of bits per value that need to be used for this encoding, if we use an information-theoretically optimal encoding.

The results, given in Table 2, show that surprisingly coarse levels of precision give excellent results, with little or no loss in predictive performance. Furthermore, the memory savings achievable in this scheme are considerable, down to *less than two bits per value* for q2.7 with theoretically optimal encoding of the discrete values.

6 Acknowlegements

We would like to thank Matthew Streeter, Gary Holt, Todd Phillips, Brendan McMahan, and Mark Rose for their thoughts and contributions to this work.

References

[1] Mikhail Bilenko and Matthew Richardson. Predictive client-side profiles for personalized advertising. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 413–421, 2011.

[2] Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: bounds for k-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 203–208, 1999.

[3] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. 2008.

[4] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[5] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, January 2006.

[6] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, pages 87–94, 2008.

[7] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[8] Joshua Goodman, Gordon V. Cormack, and David Heckerman. Spam and the ongoing battle for the inbox. *Commun. ACM*, 50(2):24–33, 2 2007.

[9] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, June 2009.

[10] H. Brendan McMahan. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[11] Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 12 1987.

[12] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530, 2007.

[13] Matthew J. Streeter and H. Brendan McMahan. Less regret via online conditioning. *CoRR*, abs/1002.4862, 2010.

[14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), 1996.

[15] Benjamin Van Durme and Ashwin Lall. Probabilistic counting with randomized storage. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1574–1579, 2009.

[16] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120, 2009.

[17] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

A Appendix: Proofs

We use the following well–known inequality, which is a direct corollary of the Azuma–Hoeffding inequality. For a proof, see [5].

Theorem A.1. *Let $X_1, \ldots, X_d$ be independent random variables such that for each i, there is a constant c_i such that $|X_i - \mathbf{E}[X_i]| \leq c_i$, always. Let $X = \sum_{i=1}^{d} X_i$. Then $\mathbf{Pr}[|X - \mathbf{E}[X]| \geq t] \leq 2\exp\{-t^2/2\sum_i c_i^2\}$.*

A.1 Encoding During Training Time

If we use projected gradient descent with randomized rounding onto a grid of precision ϵ, how does the cumulative regret bound degrade? Theorem 4.1 provides a quantitative bound, which we will now prove.

Proof of Theorem 4.1

We modify Zinkevich's analysis as follows: Let $g^t = \nabla c^t(\hat{\beta}^t)$ be the gradient at time t at the played point, where c^t is the round t convex cost function. Let β^* be any feasible point (with possibly infinite precision coefficients). By the convexity of c^t, we have

$$c^t(\hat{\beta}^t) - c^t(\beta^*) \leq g^t \cdot (\hat{\beta}^t - \beta^*). \tag{1}$$

and by the definition of β^{t+1}, we can conclude

$$||\beta^{t+1} - \beta^*||^2 = ||\hat{\beta}^t - \beta^*||^2 - 2\eta_t g^t \cdot (\hat{\beta}^t - \beta^*) + \eta_t^2 ||g^t||^2. \tag{2}$$

Rearranging the above yields

$$g^t \cdot (\hat{\beta}^t - \beta^*) \leq \frac{1}{2\eta_t} \left(||\hat{\beta}^t - \beta^*||^2 - ||\beta^{t+1} - \beta^*||^2 \right) + \frac{\eta_t}{2} ||g^t||^2. \tag{3}$$

At this point we could carry through Zinkevich's analysis verbatim, except for the fact that we play $\hat{\beta}^{t+1}$ instead of β^{t+1}. Hence we add a (zero) factor of $\frac{1}{2\eta_t}\left(||\hat{\beta}^{t+1} - \beta^*||^2 - ||\hat{\beta}^{t+1} - \beta^*||^2 \right)$ to equation 3, and define an *excess-regret* factor of $\rho\,(t) = \frac{1}{2\eta_t}\left(||\hat{\beta}^{t+1} - \beta^*||^2 - ||\beta^{t+1} - \beta^*||^2 \right)$. This yields

$$g^t \cdot (\hat{\beta}^t - \beta^*) \leq \frac{1}{2\eta_t} \left(||\hat{\beta}^t - \beta^*||^2 - ||\hat{\beta}^{t+1} - \beta^*||^2 \right) + \frac{\eta_t}{2} ||g^t||^2 + \rho\,(t)\,. \tag{4}$$

We can then combine this with equation 1, sum over t, and repeat Zinkevich's analysis to obtain a bound of

$$\text{regret}\,(T) \leq \frac{||F||^2}{2\eta_T} + \frac{||\nabla c||^2}{2} \sum_t \eta_t + \sum_t \rho\,(t) \tag{5}$$

To complete the proof, it suffices to plug in the bound on the excess regret, $\rho\,(t)$, from Lemma A.2.

Lemma A.2. *Using the notation above, under the assumptions in the second part of Theorem 4.1, the expected excess regret is bounded by*

$$\mathbf{E}[\rho\,(t)] \leq 2\,\epsilon\,||g^t||_1$$

where the expectation is computed over the randomized rounding.

Proof. Fix t. Recall $\rho\,(t) := \frac{1}{2\eta_t}\left(||\hat{\beta}^{t+1} - \beta^*||^2 - ||\beta^{t+1} - \beta^*||^2 \right)$. Note $\rho\,(t)$ is invariant under translations of the vectors $\hat{\beta}^{t+1}, \beta^{t+1}$, and β^*. For convenience, we bound $\rho\,(t)$ in the coordinate system obtained by applying the translation $\tau\,(\beta) = \beta - \beta^*$, so that in this new coordinate system β^* is at the origin. Let $G_1 \times G_2 \times \cdots \times G_d$ be our ϵ-precision grid in the new coordinate system; in general it need not include the origin. Formally, $G_i = \{\tau\,(j\epsilon) : j \in \mathbb{Z}\}$. For any $v \in \mathbb{R}$ and i, let $\underline{G}_i(v) = \max\{u \in G_i : u \leq v\}$ and $\bar{G}_i(v) = \min\{u \in G_i : u > v\}$. For the remainder of the proof, all references to points such as $\hat{\beta}^t, \beta^t$, and β^* are in the new coordinate system.

We divide the coordinates i into two sets, roughly corresponding to the set of coordinates whose values are so close to $\beta_i^* = 0$ that rounding β_i^{t+1} either up or down to the nearest grid point can increase its distance to β_i^*, and the set of coordinates for which rounding might either increase or decrease their distance to β_i^{t+1}. Formally, we define the sets $A = \{i : \beta_i^{t+1} \in [\underline{G}_i(0)\,, \bar{G}_i(0)]\}$ and $B = \{i : i \notin A\}$.

Then in the new coordinate system

$$\mathbf{E}[\rho\,(t)] = \frac{1}{2\eta_t}\mathbf{E}\left[\sum_{i \in A} \left((\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2 \right) + \sum_{i \in B} \left((\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2 \right) \right] \tag{6}$$

Applying linearity of expectation to look at the contributions coordinate–wise, we bound the contribution from each set of coordinates separately.

We start with the case that $i \in A$. Note that since β_i^{t+1} is in the same grid cell as 0, $|\beta_i^{t+1}| \leq \epsilon$ and $|\hat{\beta}_i^{t+1}| \leq \epsilon$. Hence

$$(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2 = \left(\hat{\beta}_i^{t+1} + \beta_i^{t+1} \right) \left(\hat{\beta}_i^{t+1} - \beta_i^{t+1} \right) \leq 2\epsilon \left(\hat{\beta}_i^{t+1} - \beta_i^{t+1} \right). \tag{7}$$

We can divide the expected contribution to $(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2$ from two scenarios: $\hat{\beta}_i^{t+1} = \hat{\beta}_i^t$, or $\hat{\beta}_i^{t+1} \neq \hat{\beta}_i^t$. In the first scenario, $\hat{\beta}_i^{t+1} - \beta_i^{t+1} = \hat{\beta}_i^t - \beta_i^{t+1} = \eta_t g_i^t$. Combining with equation 7 yields a bound of $(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2 \leq 2\epsilon\eta_t|g_i^t|$. In the second scenario, we use the facts that $|\hat{\beta}_i^{t+1} - \beta_i^{t+1}| \leq \epsilon$, and that $\mathbf{Pr}\big[2^{\text{nd}} \text{ scenario}\big] \leq |\hat{\beta}_i^t - \beta_i^{t+1}|/\epsilon$ to conclude that in this scenario contributes at most $2\epsilon\eta_t|g_i^t|$ to $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\Big]$. The total expected contribution from coordinates in A is thus bounded by

$$\mathbf{E}\left[\sum_{i\in A}\left((\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\right)\right] \leq 4\,\epsilon\,\eta_t \sum_{i\in A} |g_i^t| \tag{8}$$

Next, consider the case that $i \in B$, which implies that both $\underline{G}_i\left(\beta_i^{t+1}\right)$ and $\bar{G}_i\left(\beta_i^{t+1}\right)$ are on the same side of the origin. Without loss of generality, suppose $\beta_i^{t+1} > 0$. Let $a = \underline{G}_i\left(\beta_i^{t+1}\right), b = \bar{G}_i\left(\beta_i^{t+1}\right) = a + \epsilon$. We consider two possibilities: $\hat{\beta}^t \leq a$ and $\hat{\beta}^t \geq b$. If $\hat{\beta}^t \leq a$, let $\Delta = \beta_i^{t+1} - a$. Then $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2\Big] = \frac{\Delta}{\epsilon}b^2 + \left(1 - \frac{\Delta}{\epsilon}\right)a^2$ and $\beta_i^{t+1} = a + \Delta$. After some algebra, we get

$$\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\Big] = \Delta(\epsilon - \Delta) \leq \Delta\epsilon \tag{9}$$

Since $\hat{\beta}^t \leq a$ by assumption, $\Delta \leq \eta_t|g_i^t|$, so $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\Big] \leq \epsilon\eta_t|g_i^t|$. Similarly, if $\hat{\beta}^t \geq b$, let $\Delta = b - \beta_i^{t+1}$. In this case, $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2\Big] = \frac{\Delta}{\epsilon}a^2 + \left(1 - \frac{\Delta}{\epsilon}\right)b^2$. After some more algebra, we again get $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\Big] = \Delta(\epsilon - \Delta)$, and again conclude $\mathbf{E}\Big[(\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\Big] \leq \epsilon\eta_t|g_i^t|$. Hence,

$$\mathbf{E}\left[\sum_{i\in B}\left((\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\right)\right] \leq \epsilon\,\eta_t \sum_{i\in B} |g_i^t| \tag{10}$$

Combining equation 8 and equation 10, we get an overall bound of

$$\mathbf{E}\left[\sum_{i}\left((\hat{\beta}_i^{t+1})^2 - (\beta_i^{t+1})^2\right)\right] \leq 4\epsilon\,\eta_t \sum_{i\in B} ||g^t||_1 \tag{11}$$

which, after dividing by $2\eta_t$, completes the proof. □

A.2 Encoding During Prediction Time

Consider a trained model β and round it in some manner to some $\hat{\beta} \in (\epsilon\mathbb{Z})^d$. How can we bound the resulting degradation in loss?

Lemmas 5.1 and 5.2 provide bounds in terms of the quantity $|\beta \cdot x - \hat{\beta} \cdot x|$. The former is proved in Section 5.1; we now provide a proof of the latter.

Proof of Lemma 5.2

As before, without loss of generality, we can suppose $y = 1$, and $p_{\hat{\beta}}\left(x\right) \leq p_{\beta}\left(x\right)$. First we bound the error relative to our rounded model in terms of $\delta = |\beta \cdot x - \hat{\beta} \cdot x|$:

$$\frac{\mathcal{L}\left(x, y, \hat{\beta}\right) - \mathcal{L}\left(x, y, \hat{\beta}\right)}{\mathcal{L}\left(x, y, \hat{\beta}\right)} \leq \delta \tag{12}$$

Define z so that $p_{\hat{\beta}}\left(x\right) = (1 + e^{-z})^{-1}$ so that $p_{\beta}\left(x\right) = (1 + e^{-(z+\delta)})^{-1}$.

The following formula for the derivative of the logistic–loss in logistic regression will be useful:

$$\frac{\partial\mathcal{L}\left(x, y, \beta\right)}{\partial\beta_i} = x_i\left(p_{\beta}\left(x\right) - y\right) \tag{13}$$

Let $\hat{\alpha}$ be the loss incurred by $\hat{\beta}$, i.e., $\hat{\alpha} = -\log p_{\hat{\beta}}\left(x\right)$. Then $p_{\hat{\beta}}\left(x\right) = e^{-\hat{\alpha}}$. Define $\mathcal{L}\left(w\right) := -\log(\sigma(w))$. Note that we can rewrite equation (13) in the context of a single parameter model with a single (always on) feature as

$$\frac{\partial\mathcal{L}\left(w\right)}{\partial w} = \sigma(w) - y.$$

Using this fact in our setting, with $y = 1$ and $p_{\hat{\beta}}(x) = e^{-\hat{\alpha}}$, together with the convexity of the loss function, we take the linear approximation to $\mathcal{L}(w)$ around $w = z$ to lowerbound $\mathcal{L}(z + \delta)$, and conclude

$$\mathcal{L}(x, y, \beta) \geq \hat{\alpha} - (1 - e^{-\hat{\alpha}})\delta.$$

The relative error (12) can then be bounded by

$$\begin{aligned}
\frac{\mathcal{L}\left(x, y, \hat{\beta}\right) - \mathcal{L}(x, y, \beta)}{\mathcal{L}\left(x, y, \hat{\beta}\right)} &\leq \frac{\hat{\alpha} - \left(\hat{\alpha} - (1 - e^{-\hat{\alpha}})\delta\right)}{\hat{\alpha}} && (14) \\
&= \delta \left(\frac{1 - e^{-\hat{\alpha}}}{\hat{\alpha}}\right) && (15) \\
&\leq \delta && (16)
\end{aligned}$$

where in the last line we have used $1 + x \leq e^x$ for all $x \in \mathbb{R}$ to establish $1 - e^{-\hat{\alpha}} \leq \hat{\alpha}$.

Now, let $\alpha := \mathcal{L}(x, y, \beta)$ be the loss of β. We have a bound of $(\hat{\alpha} - \alpha)/\hat{\alpha} \leq \delta$ which we want to convert to the form $(\hat{\alpha} - \alpha)/\alpha \leq f(\delta)$ for some function f. To do so, note the bound we have established already gives $\hat{\alpha} \leq \alpha/(1 - \delta)$, and hence $(\hat{\alpha} - \alpha)/\alpha \leq 1/(1 - \delta) - 1 = \delta/(1 - \delta)$, which completes the proof.

Proof of Theorem 5.4

Fix a binary feature vector x and label y. We first prove the bound on additive error in Theorem 5.4. Note that $|\beta \cdot x - \hat{\beta} \cdot x| \leq \epsilon ||x||_1$, since $|\beta_i - \hat{\beta}_i| \leq \epsilon$ for all β and i. This with Lemma 5.1 proves the claimed bound of $\mathcal{L}\left(x, y, \hat{\beta}\right) - \mathcal{L}(x, y, \beta) \leq \epsilon \cdot ||x||_1$. For the bound on relative error, we start with Lemma 5.2 and integrate over t, as follows. Let $R = \frac{\mathcal{L}\left(x,y,\hat{\beta}\right) - \mathcal{L}(x,y,\beta)}{\mathcal{L}(x,y,\beta)}$ denote the relative error due to rounding, and let $R(t)$ be the worst case expected relative error in cases where $t = |\hat{\beta} \cdot x - \beta \cdot x|$. Since $\epsilon \leq 1/2k \leq 1/2||x||_0$, it is clear that $|\hat{\beta} \cdot x - \beta \cdot x| \leq ||x||_0\, \epsilon \leq 1/2$. By Lemma 5.2, $R(t) \leq t/(1 - t)$ and since $t \leq 1/2$ we have $R(t) \leq 2t$. Hence, we have

$$\begin{aligned}
\mathbf{E}[R] &\leq \textstyle\int_{t=0}^{1/2} \mathbf{Pr}\left[|\hat{\beta} \cdot x - \beta \cdot x| = t\right] R(t)\, dt \\
&\leq \textstyle\int_{t=0}^{1/2} \mathbf{Pr}\left[|\hat{\beta} \cdot x - \beta \cdot x| = t\right] \cdot 2t\, dt \\
&= 2 \textstyle\int_{t=0}^{1/2} \mathbf{Pr}\left[|\hat{\beta} \cdot x - \beta \cdot x| \geq t\right] dt \\
&\leq 4 \textstyle\int_{t=0}^{1/2} \exp\left(\frac{-t^2}{2\epsilon^2 ||x||_0}\right) dt && \text{[Lemma 5.3]} \\
&\leq 4 \textstyle\int_{t=0}^{\infty} \exp\left(\frac{-t^2}{2\epsilon^2 ||x||_0}\right) dt \\
&= 2\sqrt{2\pi ||x||_0} \cdot \epsilon && \text{[Gaussian Integral]} \\
&\leq 2\sqrt{2\pi k} \cdot \epsilon && \text{[Definition of } k\text{]}
\end{aligned}$$

which concludes the proof.