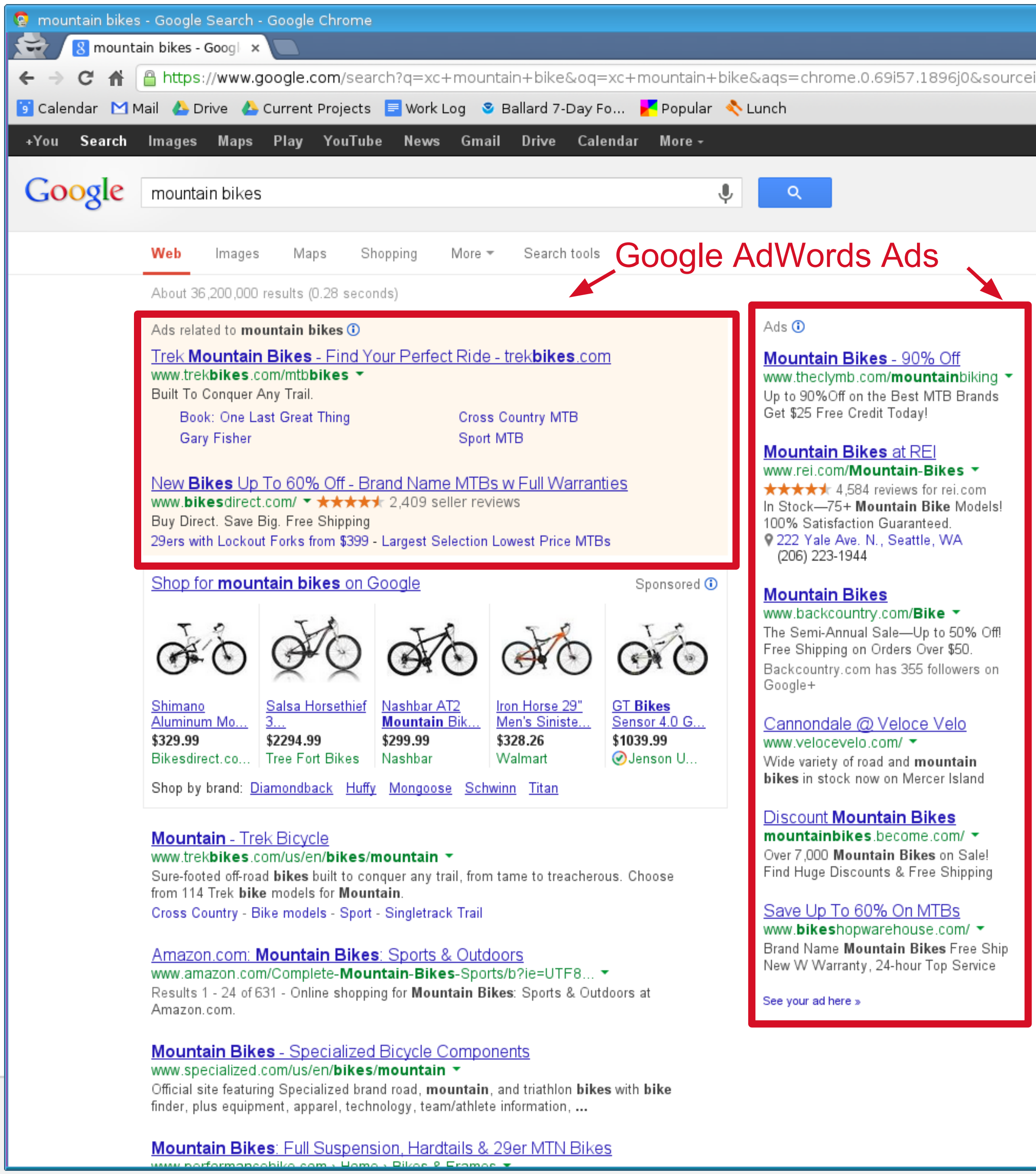
Ad Click Prediction: a View from the Trenches

Presented by Brendan McMahan
H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, Jeremy Kubica

Web Search Advertising

Targeted ads shown in response to a specific user query on a search engine.

Many billions of dollars a year are spent on web advertising.



mountain bikes - Google Search - Google Chrome

https://www.google.com/search?q=xc+mountain+bike&oq=xc+mountain+bike&aqs=chrome.0.69i57.1896j0&source

Google AdWords Ads

About 36,200,000 results (0.28 seconds)

Ads related to **mountain bikes**

Trek **Mountain Bikes** - Find Your Perfect Ride - trek**bikes**.com
www.trek**bikes**.com/mtb**bikes**
Built To Conquer Any Trail.
Book: One Last Great Thing — Cross Country MTB
Gary Fisher — Sport MTB

New **Bikes** Up To 60% Off - Brand Name MTBs w Full Warranties
www.**bikes**direct.com/ ★★★★★ 2,409 seller reviews
Buy Direct. Save Big. Free Shipping
29ers with Lockout Forks from $399 - Largest Selection Lowest Price MTBs

Shop for **mountain bikes** on Google — Sponsored

Shimano Aluminum Mo...	Salsa Horsethief 3...	Nashbar AT2 **Mountain** Bik...	Iron Horse 29" Men's Siniste...	GT **Bikes** Sensor 4.0 G...
$329.99	$2294.99	$299.99	$328.26	$1039.99
Bikesdirect.co...	Tree Fort Bikes	Nashbar	Walmart	Jenson U...

Shop by brand: Diamondback Huffy Mongoose Schwinn Titan

Mountain - Trek Bicycle
www.trek**bikes**.com/us/en/**bikes**/**mountain**
Sure-footed off-road **bikes** built to conquer any trail, from tame to treacherous. Choose from 114 Trek **bike** models for **Mountain**.
Cross Country - Bike models - Sport - Singletrack Trail

Amazon.com: **Mountain Bikes**: Sports & Outdoors
www.amazon.com/Complete-**Mountain**-**Bikes**-Sports/b?ie=UTF8...
Results 1 - 24 of 631 - Online shopping for **Mountain Bikes**: Sports & Outdoors at Amazon.com.

Mountain Bikes - Specialized Bicycle Components
www.specialized.com/us/en/**bikes**/**mountain**
Official site featuring Specialized brand road, **mountain**, and triathlon **bikes** with **bike** finder, plus equipment, apparel, technology, team/athlete information, ...

Mountain Bikes: Full Suspension, Hardtails & 29er MTN Bikes

Ads

Mountain Bikes - 90% Off
www.theclymb.com/**mountain**biking
Up to 90%Off on the Best MTB Brands
Get $25 Free Credit Today!

Mountain Bikes at REI
www.rei.com/**Mountain**-**Bikes**
★★★★★ 4,584 reviews for rei.com
In Stock—75+ **Mountain Bike** Models!
100% Satisfaction Guaranteed.
222 Yale Ave. N., Seattle, WA
(206) 223-1944

Mountain Bikes
www.backcountry.com/**Bike**
The Semi-Annual Sale—Up to 50% Off!
Free Shipping on Orders Over $50.
Backcountry.com has 355 followers on Google+

Cannondale @ Veloce Velo
www.velocevelo.com/
Wide variety of road and **mountain bikes** in stock now on Mercer Island

Discount **Mountain Bikes**
mountainbikes.become.com/
Over 7,000 **Mountain Bikes** on Sale!
Find Huge Discounts & Free Shipping

Save Up To 60% On MTBs
www.**bikes**hopwarehouse.com/
Brand Name **Mountain Bikes** Free Ship
New W Warranty, 24-hour Top Service

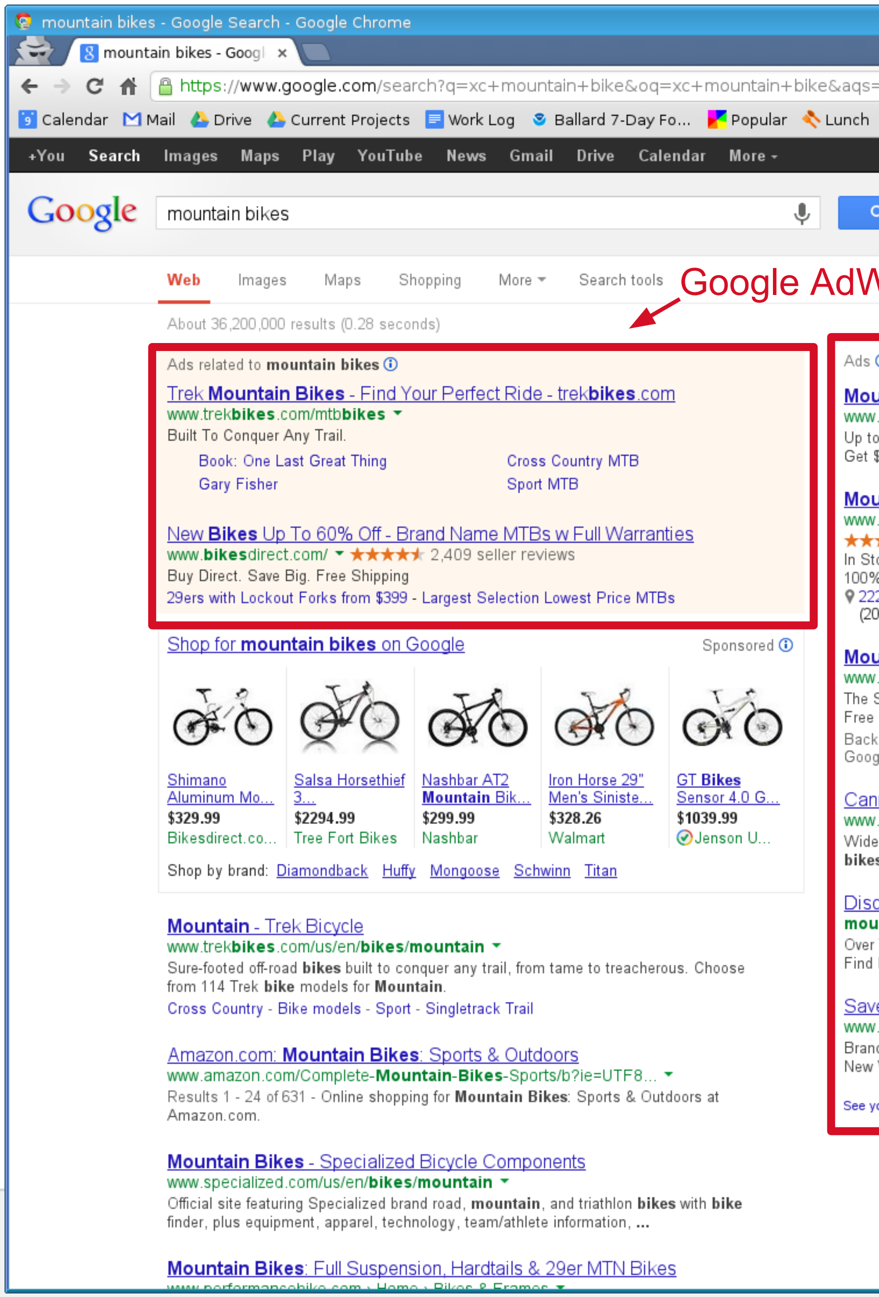
See your ad here »

Ad Click Prediction

Web search ads align incentives:

- The search engine only gets paid if the user clicks the ad
- So, the search engine only wants to show relevant ads
- and user only wants to see relevant ads.

Predicting the probability of a click for a specific ad in response to a specific query is the key ingredient!



mountain bikes - Google Search - Google Chrome

https://www.google.com/search?q=xc+mountain+bike&oq=xc+mountain+bike&aqs=

+You Search Images Maps Play YouTube News Gmail Drive Calendar More

mountain bikes

Web Images Maps Shopping More Search tools

Google AdW

About 36,200,000 results (0.28 seconds)

Ads related to **mountain bikes**

Trek Mountain Bikes - Find Your Perfect Ride - trek**bikes**.com
www.trek**bikes**.com/mtb**bikes**
Built To Conquer Any Trail.
Book: One Last Great Thing — Cross Country MTB
Gary Fisher — Sport MTB

New **Bikes** Up To 60% Off - Brand Name MTBs w Full Warranties
www.**bikes**direct.com/ ★★★★★ 2,409 seller reviews
Buy Direct. Save Big. Free Shipping
29ers with Lockout Forks from $399 - Largest Selection Lowest Price MTBs

Shop for **mountain bikes** on Google — Sponsored

Shimano Aluminum Mo...	Salsa Horsethief 3...	Nashbar AT2 **Mountain** Bik...	Iron Horse 29" Men's Siniste...	GT **Bikes** Sensor 4.0 G...
$329.99	$2294.99	$299.99	$328.26	$1039.99
Bikesdirect.co...	Tree Fort Bikes	Nashbar	Walmart	Jenson U...

Shop by brand: Diamondback Huffy Mongoose Schwinn Titan

Mountain - Trek Bicycle
www.trek**bikes**.com/us/en/**bikes**/**mountain**
Sure-footed off-road **bikes** built to conquer any trail, from tame to treacherous. Choose from 114 Trek **bike** models for **Mountain**.
Cross Country - Bike models - Sport - Singletrack Trail

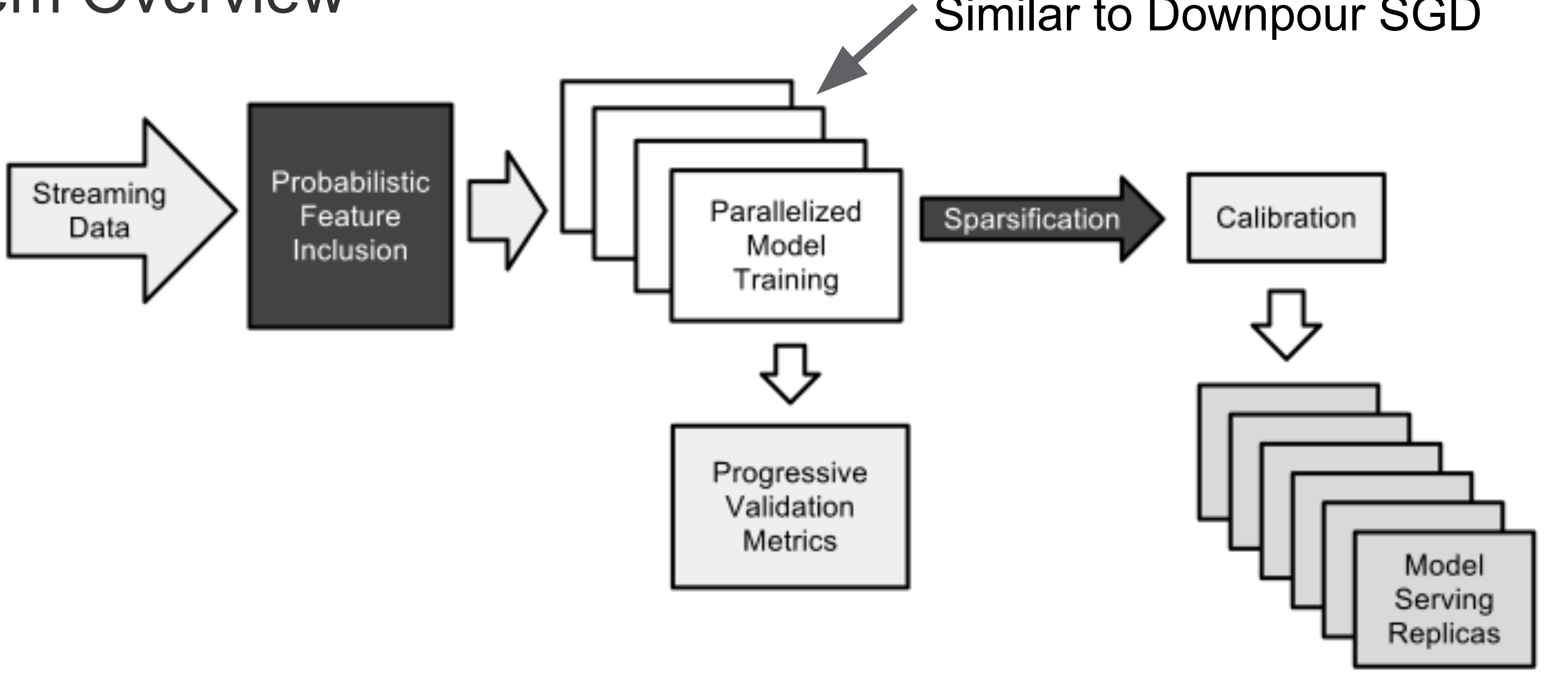
Amazon.com: **Mountain Bikes**: Sports & Outdoors
www.amazon.com/Complete-**Mountain**-**Bikes**-Sports/b?ie=UTF8...
Results 1 - 24 of 631 - Online shopping for **Mountain Bikes**: Sports & Outdoors at Amazon.com.

Mountain Bikes - Specialized Bicycle Components
www.specialized.com/us/en/**bikes**/**mountain**
Official site featuring Specialized brand road, **mountain**, and triathlon **bikes** with **bike** finder, plus equipment, apparel, technology, team/athlete information, ...

Mountain Bikes: Full Suspension, Hardtails & 29er MTN Bikes

Ads
Mou
www.
Up to
Get $
Mou
www.
In Sto
100%
222
(20
Mou
www.
The S
Free
Back
Goog
Can
www.
Wide
bikes
Disc
mou
Over
Find
Save
www.
Brand
New
See y

System Overview


Similar to Downpour SGD
Streaming Data
Probabilistic Feature Inclusion
Parallelized Model Training
Sparsification
Calibration
Progressive Validation Metrics
Model Serving Replicas

Training massive models on massive data with minimum resources:

- *billions of unique features (model coefficients)*
- *billions of predictions per day serving live traffic*
- *billions of training examples*

Google

The FTRL-Proximal Online Learning Algorithm

"Like Stochastic Gradient Descent, but much sparser models."

- Online algorithms: simple, scalable, rich theory
- FTRL-Proximal is equivalent to Online (Stochastic) Gradient Descent when no regularization is used [1]
- Implements regularization in a way similar to RDA [2], but gives better accuracy in our experiments. The key is re-expressing gradient descent implicitly as:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left(\mathbf{g}_{1:t} \cdot \mathbf{w} + \frac{1}{2} \sum_{s=1}^{t} \sigma_s \|\mathbf{w} - \mathbf{w}_s\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 \right)$$

- With a few tricks, about as easy to implement as gradient descent (a few lines of code, pseudo-code in the paper)

[1] *Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization.* McMahan, AISTATS 2011.
[2] *Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization*. Lin Xiao, JMLR 2010.

Per-Coordinate Learning Rates: Intuition

Consider predicting for:

- English queries in the US (1,000,000 training examples)
- Icelandic queries in Iceland (100 training examples)

Imagine the features are disjoint.

If we trained **separate SGD models** for each, we would use a very different learning rate for each: generally ~ 1/sqrt(# examples)

Train a single model:
concatenate the feature vectors, interleave training examples
There is no good learning rate choice! SGD will either:

- never converge for Iceland (with a low learning rate)
- or oscillate wildly for the US (with a large learning rate)

There is a middle ground, but it is still suboptimal.

Google

Per-Coordinate Learning Rates: Implementation

- Theory indicates the learning rate $\eta_{t,i} = \dfrac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t} g_{s,i}^2}}$
- Requires storing one extra statistic per coordinate
  - But we have a trick for reducing this if you are training multiple similar models
- Huge accuracy improvement:
  - Improved AUC by 11.2% versus a global learning rate baseline
  - (In our setting a 1% improvement is large)

References:
Adaptive Bound Optimization for Online Convex Optimization. McMahan and Streeter, COLT 2010.
Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Duchi, Hazan, and Singer, JMLR 2011.

Techniques for Saving Memory

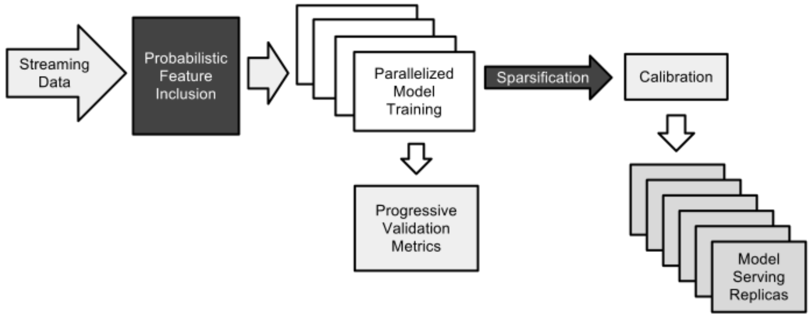
During training, we track some statistics for features not included in the final model (zero coefficients due to L1 regularization)

Saving ***Training Memory***

- Probabilistic feature inclusion
- Grouping similar models
- Randomized rounding

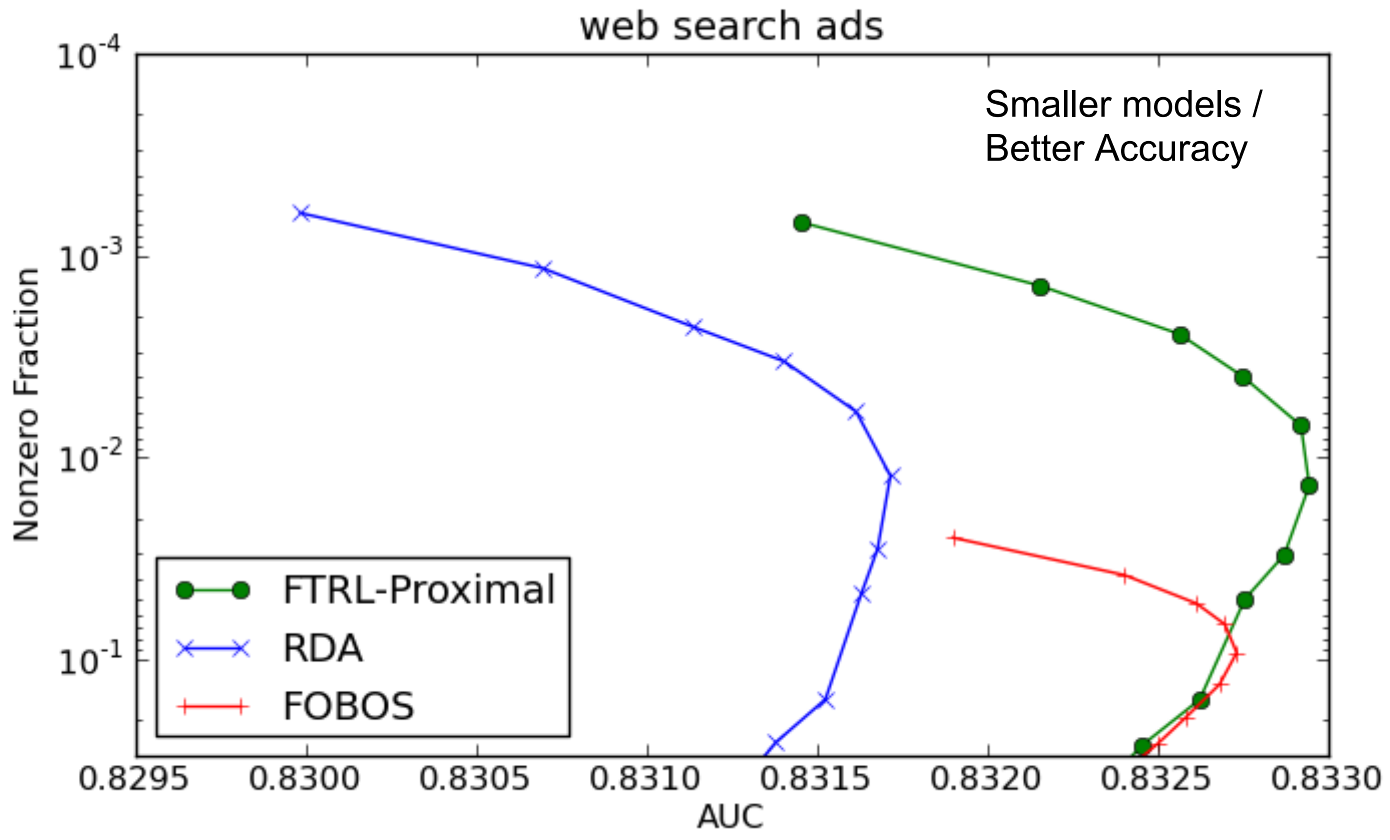
Saving ***Memory at Serving***

- L1 regularization
- Randomized rounding


Streaming Data
Probabilistic Feature Inclusion
Parallelized Model Training
Progressive Validation Metrics
Sparsification
Calibration
Model Serving Replicas

FTRL-Proximal with L1 Regularization

Tradeoff frontier for a small (10^6 examples) dataset



Each line varies the L1 parameter for different size/accuracy tradeoffs

FTRL-Proximal with L1 Regularization

Full-scale experiment

- Baseline algorithm: Only include features that occur at least K times.
    - Tuned the L1 parameter for FTRL-Proximal to provide a good sparsity/accuracy tradeoff.
    - Then tuned K to get the same accuracy with the baseline.

- Baseline model has about 3x as many non-zero entries compared to FTRL-Proximal with L1 regularization

Probabilistic Feature Inclusion

In some of our models, ***half*** *of the unique features occur only* ***once*** *in a training set of billions of examples.*

- Long-tailed distributions produce many features that only occur a handful of times. These generally aren't useful for prediction.
- So, how do you tell these rare features from common ones without tracking any state? Two probabilistic techniques:
  - Poisson inclusion: When we see a feature not in the model, add it with probability *p.*
  - Bloom-filter inclusion: use a rolling set of counting Bloom filters to count occurrences (include after *K*, but sometimes includes a feature that has occurred fewer than *K* times).

Method	RAM Saved	AucLoss Detriment
Bloom ($n = 2$)	66%	0.008%
Bloom ($n = 1$)	55%	0.003%
Poisson ($p = 0.03$)	60%	0.020%
Poisson ($p = 0.1$)	40%	0.006%

Storing Coefficients with Fewer Bits

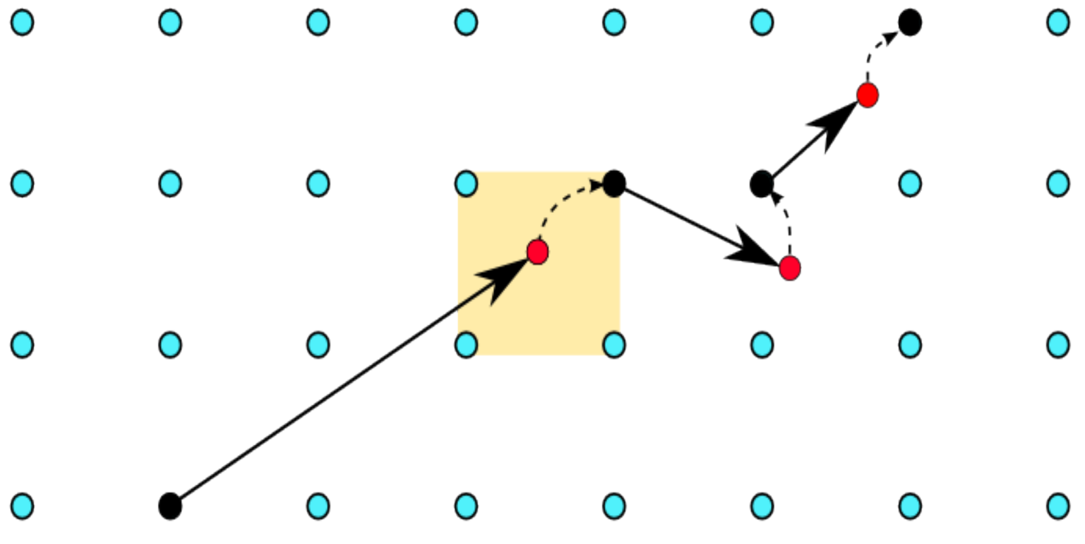
The learning rate tells us how accurately we might know the coefficient.

- If training on one example might change the coefficient by Δ, you don't know the value with more precision than Δ.
- So, don't waste memory storing a full-precision floating-point value! (32-64 bits)
- Adaptive schemes are possible
- But in practice, storing a 16-bit fixed point representation is easy and works very well.

But, you need to be careful about accumulating errors. The trick: **randomized rounding**

Large-Scale Learning with Less RAM via Randomization. Golovin, Sculley, McMahan, Young. ICML 2013.

Training with Randomized Rounding



$$\beta_{t+1} = \text{Project}\left(\beta_t - \eta_t g_t\right)$$

Compute the usual update (Gradient Descent or FTRL-Proximal) at full precision, then randomly project to an adjacent value expressible in the fixed-point encoding.

Use unbiased rounding:
E[rounded coefficient]
= unrounded coefficient

No accuracy loss with a 16-bit fixed-point representation,
compared to 32- or 64 bit floating point values (and 50-75% less memory)

Training Many Similar Models: Individually

Training 4 similar models, each model trained separately with coefficients stored in a hash table.
Each model uses say 4+4+2 = **10 bytes per coefficient**

Feature Key (e.g., 32-64 bit hash)	Gradient-Squared Sum	Model 0 Coefficient (q2.13)

Feature Key (e.g., 32-64 bit hash)	Gradient-Squared Sum	Model 1 Coefficient (q2.13)

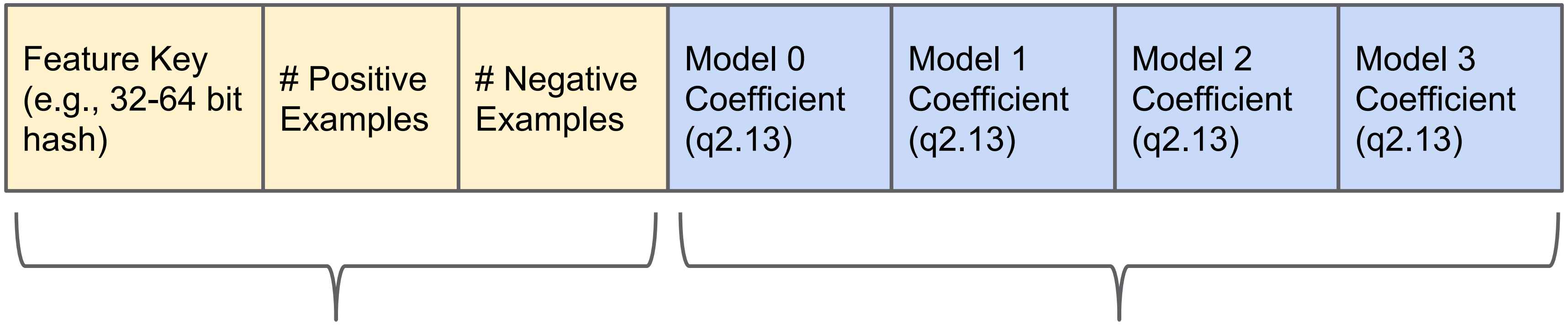
Feature Key (e.g., 32-64 bit hash)	Gradient-Squared Sum	Model 2 Coefficient (q2.13)

Feature Key (e.g., 32-64 bit hash)	Gradient-Squared Sum	Model 3 Coefficient (q2.13)

Why many similar models? We may want to try different feature variations, learning rates, regularization strengths, ...

Training Many Similar Models: Grouped

Data for multiple similar models stored in a single hash table
Only **2 bytes per coefficient** (plus 12 bytes of overhead) per model


Feature Key (e.g., 32-64 bit hash)
# Positive Examples
# Negative Examples
Model 0 Coefficient (q2.13)
Model 1 Coefficient (q2.13)
Model 2 Coefficient (q2.13)
Model 3 Coefficient (q2.13)

Common data is only stored once, so cost (memory, cpu, and network) is amortized.

The only cost per model variant is for storing the model's coefficient, which can be small (say 16 bit q2.13 fixed point).

Rather than using exact per-coordinate learning rates based on each model's gradients, we use a (good enough) approximation based on the # of observed positive and negative examples.

Computing Learning Rates with Counts

The previous trick requires using the same learning-rate statistic for all grouped models. Approximate the theory-recommended learning-rate schedule

$$\eta_{t,i} = \frac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t} g_{s,i}^2}}$$

using only counts:

$$\begin{aligned}\sum g_{t,i}^2 &= \sum_{\text{positive events}} (1-p_t)^2 + \sum_{\text{negative events}} p_t^2 \\ &\approx P\left(1 - \frac{P}{N+P}\right)^2 + N\left(\frac{P}{N+P}\right)^2 \\ &= \frac{PN}{N+P}.\end{aligned}$$

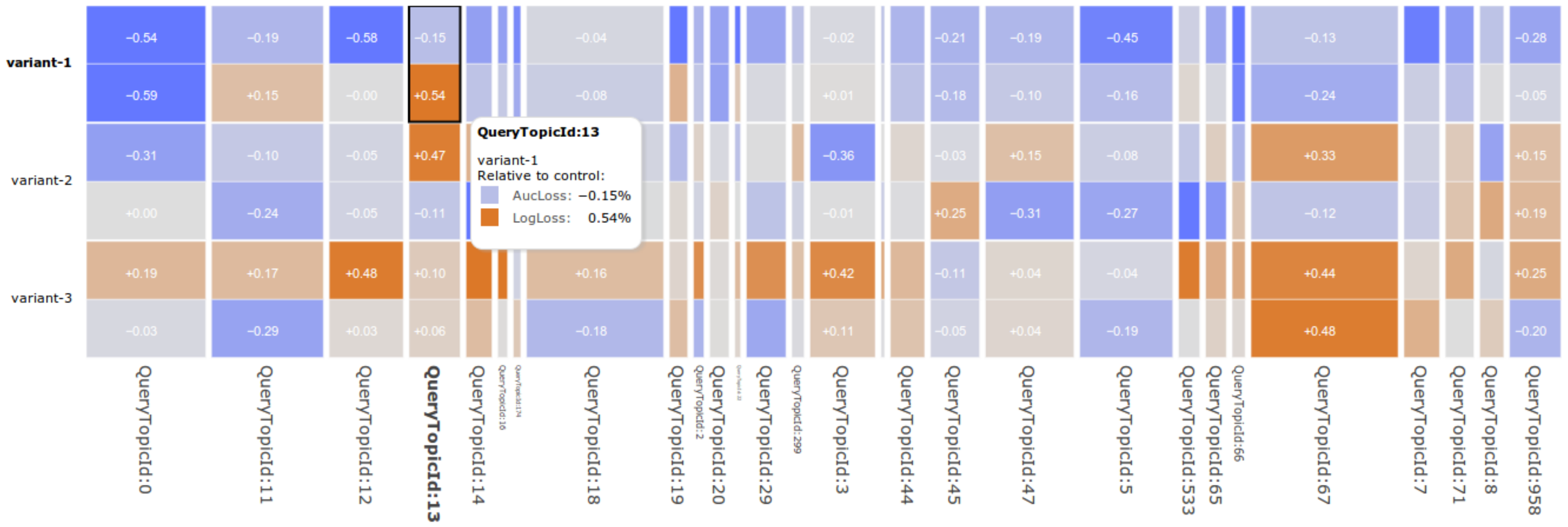
N is the number of Negative examples
P is the number of Positive examples

High-Dimensional Data Visualization for Model Accuracy

We don't just care about aggregate accuracy!

- Progressive validation: compute prediction on each example before training on it
- Look at relative changes in accuracy between models
- Data visualization to quickly spot outliers or problem areas

High-Dimensional Data Visualization for Model Accuracy



Automated Feature Management System

Many raw **signals**: words-in-user-query, country-of-origin, etc

Lots of engineers working on signals, multiple different learning platforms / teams consuming them.

A metadata index to manage all these signals:

- what systems are using what signals
- what signals are available to different systems
- status and version information / deprecation
- whitelists for production readiness / test status

Used for automatic testing, alerting, notification, cleanup of unused signals, etc.

Also in the paper …

- Fast approximate training with a single value structure
- Unbiased training on a biased subsample of training data
- Assessing model accuracy with progressive validation
- Cheap confidence estimates
- Approaches for calibrating predictions

Techniques that didn't work as well as expected

- Aggressive feature hashing
- Randomized feature dropout
- Averaging models trained on different subsets of the features
- Feature vector normalization

Thanks for listening!

And thanks to all my co-authors:

Gary Holt, **D. Sculley**, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, **Eugene Davydov**, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, Jeremy Kubica