Case Study 1: Estimating Click Probabilities

SGD cont'd
AdaGrad

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
April 2nd, 2015

Learning Problem for Click Prediction

Case Study 1

- Prediction task: $X \to \{0,1\}$ $\quad P(\text{click}=1 \mid X)$ ← features
- Features: $X = $ (feats of page, ad, user)
- Data: (x^i, y^i) $\quad$ (webpage1, ad7, user25, time12) ← x^i; click = 1 ← y^i
  - Batch: Fixed dataset $(x^1, y^1) \ldots (x^N, y^N)$
  - Online: data as a stream; user arrives at a page → x^t → predict $\hat{y}$ click?; observe y^t
- Many approaches (e.g., logistic regression, SVMs, naïve Bayes, decision trees, boosting,…)
  - Focus on logistic regression; captures main concepts, ideas generalize to other approaches

Standard v. Regularized Updates

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[\prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

- Regularized maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[\prod_j P(y^j | \mathbf{x}^j, \mathbf{w})) \right] - \frac{\lambda}{2} \sum_{i>0} w_i^2$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}$$

neg. derivative ← move towards 0

©Emily Fox 2015

Challenge 1: Complexity of computing gradients

d features

- What's the cost of a gradient update step for LR???

$$\underbrace{w_i^{(t+1)}}_{\text{for each } i} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \underbrace{\sum_{j=1}^{N} x_i^j [y^j - \underbrace{\hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})}_{O(d)\ \text{cache}}]}_{O(Nd)} \right\}$$

$\begin{bmatrix} w_1^{(t)} \\ \vdots \\ w_d^{(t)} \end{bmatrix}$

∀ features i, cost is $O(Nd^2)$ ... can cache $p(y^j=1|x^j, w^{(t)})$

$\boxed{O(Nd)}$

In "big data" - N is very large
$O(Nd)$ for only taking little η step

©Emily Fox 2015

Challenge 2: Data is streaming

- Assumption thus far: **Batch data** $\sum_{j=1}^{N} \cdots$
- But, click prediction is a streaming data task:
  - User enters query, and ad must be selected:
    - Observe $\mathbf{x}^j$, and must predict y^j

→ x^j → predict $\hat{y}^j$ → show ad
click?

  - User either clicks or doesn't click on ad:
    - Label y^j is revealed afterwards
      - Google gets a reward if user clicks on ad
  - Weights must be updated for next time:

$w^{(t+1)} \leftarrow w^{(t)} + \Delta$ ← depends just on recent example(s)

SGD: Stochastic Gradient Ascent (or Descent)

- "True" gradient: $\nabla \ell(\mathbf{w}) = E_{\mathbf{x}}\left[\nabla \ell(\mathbf{w}, \mathbf{x})\right]$
- Sample based approximation: $x^j \overset{iid}{\sim} p(x)$

$$\nabla \ell(w) = E_x[\nabla \ell(w,x)] \approx \hat{\nabla} \ell(w) = \frac{1}{N}\sum_{j=1}^{N} \nabla \ell(w, x^j)$$

Monte Carlo approx

the bigger N, the closer $\hat{\nabla}\ell$ to $\nabla \ell$

- What if we estimate gradient with just one sample??? $\boxed{N=1}$ $x^{(t)}$
  - Unbiased estimate of gradient $\nabla \ell(w) \approx \hat{\nabla} \ell(w) = \nabla \ell(w, x^{(t)})$
  - Very noisy! $E_x[\hat{\nabla} \ell(w)] = E_{x^{(t)}}[\nabla \ell(w, x^{(t)})]$
  - Called stochastic gradient ascent (or descent) $= \nabla \ell(w)$
    - Among many other names
  - VERY useful in practice!!!

Stochastic Gradient Ascent: General Case

- Given a stochastic function of parameters: $\ell(w) = E_x[\ell(w,x)]$ random
  - Want to find maximum

$$w^* \in \arg\max_w E_x[\ell(w,x)]$$

- Start from $\mathbf{w}^{(0)}$ e.g. $w^{(0)} = 0$
- Repeat until convergence:
  - Get a sample data point $\mathbf{x}^t$
  - predict $\hat{y}^t$ (click?) ... note: use running avg. of $w^{(t)}$ $\hat{w}_t = \frac{1}{t}\sum_{t=1}^{t} w^{(t)}$
  - sell ad, observe y^t ← actual click or not
  - Update parameters:

$$w^{(t+1)} \leftarrow w^{(t)} + \eta_t \nabla \ell(w^{(t)}, x^t)$$

actual w, not avg.

just current data pt

- Works in the online learning setting!
- Complexity of each gradient step is constant in number of examples!
- In general, step size changes with iterations

e.g. $\eta_t = K/t$ for $K > 0$

©Emily Fox 2015 7

Stochastic Gradient Ascent for Logistic Regression

- Logistic loss as a stochastic function: regularized setting

$$E_{\mathbf{x}}[\ell(\mathbf{w},\mathbf{x})] = E_{\mathbf{x}}\left[\ln P(y|\mathbf{x},\mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||_2^2\right]$$

- Batch gradient ascent updates:

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta\left\{-\lambda w_i^{(t)} + \underbrace{\frac{1}{N}\sum_{j=1}^{N} x_i^{(j)}[y^{(j)} - P(Y=1|\mathbf{x}^{(j)},\mathbf{w}^{(t)})]}_{\text{avg. of all data pts}}\right\}}$$

- Stochastic gradient ascent updates:
  - Online setting:

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta_t\left\{-\lambda w_i^{(t)} + \underbrace{x_i^{(t)}[y^{(t)} - P(Y=1|\mathbf{x}^{(t)},\mathbf{w}^{(t)})]}_{\text{one data point at a time}}\right\}}$$

©Emily Fox 2015 8

Convergence Rate of SGD

- **Theorem**:
  - (see Nemirovski et al '09 from readings)
  - Let ℓ be a strongly convex stochastic function with param $\gamma > 0$
  - Assume gradient of ℓ is Lipschitz continuous and bounded

    $\forall x \quad \|\nabla \ell(w,x) - \nabla \ell(w',x)\|_2 \leq L \|w - w'\|_2 \quad L > 0$

    and $\quad \|\nabla \ell\|_2^2 \leq M^2$
  - Then, for step sizes:

    $\eta_t = K/t \quad K > 0$
  - The expected loss decreases as O(1/t):

    e.g. $K = \frac{1}{\gamma}$

    where we started

    $$E[\ell(w^{(t)}) - \ell(w^*)] \leq \frac{1}{t} L \left(\frac{M^2}{\gamma^2} + \|w^{(0)} - w^*\|_2^2\right)$$

    how much closer getting to w^* (in exp.)

    $O(\frac{1}{t})$

    opt.

©Emily Fox 2015

Convergence Rates for Gradient Descent/Ascent vs. SGD

- Number of Iterations to get to accuracy

  $$\ell(\mathbf{w}^*) - \ell(\mathbf{w}) \leq \epsilon$$

  $O(Nd \ln \frac{1}{\epsilon}) \longleftrightarrow O(\frac{d}{\epsilon})$

  GD: N data points — minibatch (a few obs.) — SGD: 1 data point

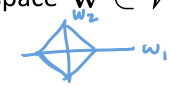
- Gradient descent:
  - If func is strongly convex: O(ln(1/ϵ)) iterations
- Stochastic gradient descent:
  - If func is strongly convex: O(1/ϵ) iterations
- Seems exponentially worse, but much more subtle:
  - Total running time, e.g., for logistic regression:
    - Gradient descent: $O(\ln \frac{1}{\epsilon})$ iterations @ $O(Nd)$/iter $\rightarrow O(Nd \ln \frac{1}{\epsilon})$
    - SGD: $O(\frac{1}{\epsilon})$ iters @ $O(d)$/iter $\rightarrow O(d/\epsilon)$
    - SGD can win when we have a lot of data
  - See readings for more details

©Emily Fox 2015

Constrained SGD: Projected Gradient

- Consider an arbitrary restricted feature space $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^d$

e.g. $W: \|w\|_1 \leq R$



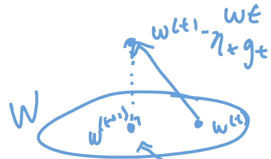
- Optimization objective: $\underset{w \in W}{\arg\min}\ \ell(w)$

let $g_t \triangleq \nabla \ell(w, x^t)$

previously: $w^{(t+1)} \leftarrow w^{(t)} - \eta_t g_t$ ... now: ?

- If $\mathbf{w} \in \mathcal{W}$, can use ***projected gradient*** for (sub)gradient descent

$$\mathbf{w}^{(t+1)} = \underset{w \in W}{\arg\min} \|w - (w^{(t)} - \eta_t g_t)\|_2^2$$



closest point in W to $w^{(t)} - \eta_t g_t$

efficient in some cases:
e.g. $W: \|w\|_1 \leq R$
$\|w\|_2 \leq R$

©Emily Fox 2015

Motivating AdaGrad (Duchi, Hazan, Singer 2011)

adaptive gradient

- Assuming $\mathbf{w} \in \mathbb{R}^d$, standard stochastic (sub)gradient descent updates are of the form:

$$w_i^{(t+1)} \leftarrow w_i^{(t)} - \eta_t g_{t,i}$$

"step size"
"learning rate"

- Should all features share the same learning rate?

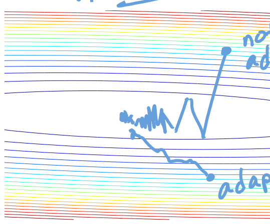
maybe instead: $\eta_{t,i}$ specific to feature i

- Often have high-dimensional feature spaces
  - Many features are irrelevant → small learning rate
  - Rare features are often very informative
- Adagrad provides a feature-specific adaptive learning rate by incorporating knowledge of the geometry of past observations

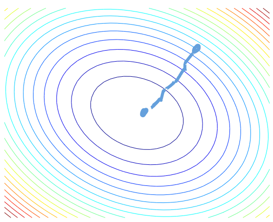
©Emily Fox 2015

Why Adapt to Geometry?

geometry



Hard



Nice

rare features

y_t	$x_{t,1}$	$x_{t,2}$	$x_{t,3}$
1	1	0	0
-1	.5	0	1
1	-.5	1	0
-1	0	0	0
1	.5	0	0
-1	1	0	0
1	-1	1	0
-1	-.5	0	1

Examples from Duchi et al. ISMP 2012 slides

1. Frequent, irrelevant
2. Infrequent, predictive
3. Infrequent, predictive

©Emily Fox 2015

Not All Features are Created Equal

- Examples:

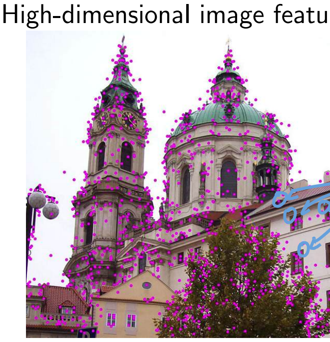
Text data:

```
The most unsung birthday
in American business and
technological history
this year may be the 50th
anniversary of the Xerox
914 photocopier.ᵃ
```

[a] *The Atlantic*, July/August 2010.

rare word

High-dimensional image features

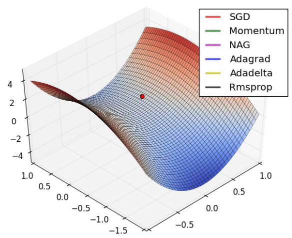


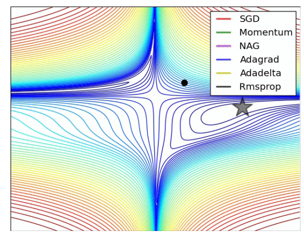
corners are rare, but informative

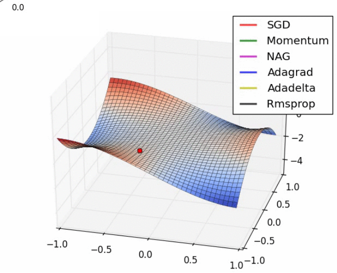
Images from Duchi et al. ISMP 2012 slides

©Emily Fox 2015

Visualizing Effect







Credit:
http://imgur.com/a/Hqolp

Regret Minimization

- How do we assess the performance of an online algorithm?
- Algorithm iteratively predicts $\mathbf{w}^{(t)}$ ← ad setting $\hat{y}^t$ click?
- Incur ***loss*** $\ell_t(\mathbf{w}^{(t)})$ ← either click or not
- ***Regret***:
  What is the total incurred loss of algorithm relative to the best choice of $\mathbf{w}$ that could have been made ***retrospectively***

"regret" cummulative loss based on seq. $w^{(1)}, w^{(2)}, \ldots$ w^*

$$R(T) = \sum_{t=1}^{T} \ell_t(\mathbf{w}^{(t)}) - \underbrace{\inf_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(\mathbf{w})}_{\text{best achievable loss for a single } w \text{ in retrospect}}$$

Typically, $\frac{R(T)}{T} \to 0$ as $T \to \infty$
$\Rightarrow w^{(t)}, w^{(t+1)}, \ldots$ as good as w^*
"no regret algorithm"

Regret Bounds for Standard SGD

- Standard projected gradient stochastic updates:

$$\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w} \in \mathcal{W}} ||\mathbf{w} - (\mathbf{w}^{(t)} - \eta g_t)||_2^2$$

same norm

- Standard regret bound:

$$\underbrace{\sum_{t=1}^{T} \ell_t(\mathbf{w}^{(t)}) - \ell_t(\mathbf{w}^*)}_{R(T)} \leq \frac{1}{2\eta} \underbrace{||\mathbf{w}^{(1)} - \mathbf{w}^*||_2^2}_{\text{error of where you started}} + \frac{\eta}{2} \sum_{t=1}^{T} \underbrace{||g_t||_2^2}_{\text{magnitude of gradients}}$$

similar to Nemirovski

Projected Gradient using Mahalanobis

- Standard projected gradient stochastic updates:

$$\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w} \in \mathcal{W}} ||\mathbf{w} - (\mathbf{w}^{(t)} - \eta g_t)||_2^2$$

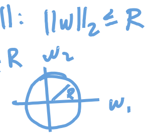
- What if instead of an L_2 metric for projection, we considered the ***Mahalanobis*** norm

$A \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$ care more about w_1 gradient

$$\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w} \in \mathcal{W}} ||\mathbf{w} - (\mathbf{w}^{(t)} - \eta A^{-1} g_t)||_A^2$$

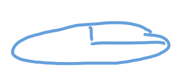
L_2 ball: $||w||_2 \leq R$

$\sqrt{w^T w} \leq R$



$||w||_A \leq R$ $A: \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$

$\sqrt{w^T A w} \leq R$



$||w||_A \leq R$ $A = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}$

$A \succeq 0$

positive semidefinite

Mahalanobis Regret Bounds

$$\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}\in\mathcal{W}} ||\mathbf{w} - (\mathbf{w}^{(t)} - \eta A^{-1} g_t)||_A^2$$

$w^{*T} A w^{*}$ $||g_t||^2_{A^{-1}} = g_t^T A^{-1} g_t$

- **What *A* to choose?**
- Regret bound now:

In Id: $w^{*2} \cdot a \to \infty$ as $a \to \infty$

$$R(T) = \sum_{t=1}^T \ell_t(\mathbf{w}^{(t)}) - \ell_t(\mathbf{w}^*) \leq \frac{1}{2\eta}||\cancel{\mathbf{w}^{(1)}}^{\,0} - \mathbf{w}^*||_A^2 + \frac{\eta}{2}\sum_{t=1}^T ||g_t||_{A^{-1}}^2$$

avoid by not letting A get too big

In Id: $||g_t||^2_{A^{-1}} = \frac{g_t^2}{a} \to 0$ as $a \to \infty$

- What if we minimize upper bound on regret w.r.t. *A* in hindsight?

$$\min_A \sum_{t=1}^T g_t^T A^{-1} g_t \quad \text{s.t. } tr(A) \leq C$$

$tr(A) = \sum_i A_{ii}$

Mahalanobis Regret Minimization

- Objective:

$$\min_A \sum_{t=1}^T g_t^T A^{-1} g_t \qquad \text{subject to } A \succeq 0, \text{tr}(A) \leq C$$

- Solution:

if $Q \succeq 0$, $\exists V$ s.t. $Q = V^T V$

$$A = c\left(\sum_{t=1}^T g_t g_t^T\right)^{\frac{1}{2}}$$

outer product of gradient

square root matrix "$V = Q^{1/2}$"

For proof, see Appendix E, Lemma 15 of Duchi et al. 2011.
Uses "trace trick" and Lagrangian.

- *A* defines the norm of the metric space we should be operating in

AdaGrad Algorithm

- At time *t*, estimate optimal (sub)gradient modification *A* by

up to time t

estimate of A at time t

$$A_t = \left(\sum_{\tau=1}^{t} g_\tau g_\tau^T\right)^{\frac{1}{2}}$$

in d dims, matrix $\sqrt{\ }$ is $O(d^3)$

- For *d* large, A_t is computationally intensive to compute. Instead,

$\text{diag}(A_t)$ $\quad A_t = \begin{pmatrix} A_{ii} & & 0 \\ & \ddots & \\ 0 & & A_{dd} \end{pmatrix}$ $\quad A_{t,ii} = \sqrt{\sum_{\tau=1}^{t} g_{\tau,i}^2}$

- Then, algorithm is a simple modification of normal updates:

$$\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}\in\mathcal{W}} ||\mathbf{w} - (\mathbf{w}^{(t)} - \eta\text{diag}(A_t)^{-1}g_t)||^2_{\text{diag}(A_t)}$$

$\eta_t \rightarrow \eta A^{-1} \rightarrow \eta\, \text{diag}(A_t)^{-1}$

weigh dimensions by sqrt of sum of past grad. in that dim.

$x^t = (1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0 \ldots)$

AdaGrad in Euclidean Space

- For $\mathcal{W} = \mathbb{R}^d$, $\quad w^{(t+1)} \leftarrow w^{(t)} - \eta\, \text{diag}(A_t)^{-1} g_t$

  no constraints on $\mathcal{W}$

- For each feature dimension,

adaptive step size for feature i

$$w_i^{(t+1)} \leftarrow w_i^{(t)} - \eta_{t,i} g_{t,i}$$

where

$$\eta_{t,i} = \eta / A_{t,ii}$$

- That is,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} - \frac{\eta}{\sqrt{\sum_{\tau=1}^{t} g_{\tau,i}^2}} g_{t,i}$$

In sparse case, stepsize larger for rare features

- Each feature dimension has it's own learning rate!
  - Adapts with *t*
  - Takes geometry of the past observations into account
  - Primary role of η is determining rate the first time a feature is encountered

AdaGrad Theoretical Guarantees

- AdaGrad regret bound:

radius of space

$$R_\infty := \max_t ||\mathbf{w}^{(t)} - \mathbf{w}^*||_\infty$$

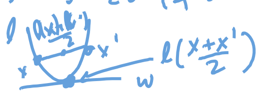
$$R(T) = \sum_{t=1}^{T} \ell_t(\mathbf{w}^{(t)}) - \ell_t(\mathbf{w}^*) \leq 2R_\infty \sum_{i=1}^{d} ||g_{1:T,i}||_2$$

Jensen's ineq.

  – In stochastic setting: $\ell(w) = E_x[\ell(w,x)]$ and $\ell_t(w) = \ell(w, x^t)$. Then,

$$\frac{E R(T)}{T} = \mathbb{E}\left[\ell\left(\frac{1}{T}\sum_{t=1}^{T} w^{(t)}\right)\right] - \ell(\mathbf{w}^*) \leq \frac{2R_\infty}{T}\sum_{i=1}^{d}\mathbb{E}[||g_{1:T,j}||_2]$$

$$E\left[\frac{1}{T}\sum_t \ell(w^{(t)}, x^t)\right] \geq E\left[\ell\left(\frac{1}{T}\sum w^{(t)}, x^t\right)\right]$$



$$\frac{1}{T} E_{x^{1:T}}\left[\sum_t \ell(w^*, x^t)\right] = \frac{1}{T}\sum_t \underbrace{E_{x^t}[\ell(w^*, x^t)]}_{\ell(w^*)} = \ell(w^*)$$

- This really is used in practice!
- Many cool examples. Let's just examine one…

©Emily Fox 2015

What you should know about Logistic Regression (LR) and Click Prediction

- Click prediction problem:
  – Estimate probability of clicking
  – Can be modeled as logistic regression
- Logistic regression model: Linear model
- Gradient ascent to optimize conditional likelihood
- Overfitting + regularization
- Regularized optimization
  – Convergence rates and stopping criterion
- Stochastic gradient ascent for large/streaming data
  – Convergence rates of SGD
- AdaGrad motivation, derivation, and algorithm

©Emily Fox 2015