Online Importance Weight Aware Updates

Nikos Karampatziakis John Langford

July 17th 2011

What this talk is about

- Importance weights encode relative importance of examples
- They appear in many settings:
  - Boosting
  - Covariate shift correction
  - Learning reductions
  - Active learning

What this talk is about

- Importance weights encode relative importance of examples
- They appear in many settings:
  - Boosting
  - Covariate shift correction
  - Learning reductions
  - Active learning
- Online gradient descent popular, sound optimizer

What this talk is about

- Importance weights encode relative importance of examples
- They appear in many settings:
  - Boosting
  - Covariate shift correction
  - Learning reductions
  - Active learning
- Online gradient descent popular, sound optimizer
- Interplay between importance weights and OGD

Online Gradient Descent – Linear Model

- Loss $\ell(w_t^\top x_t, y_t)$

$$w_{t+1} = w_t - \eta_t \left.\frac{\partial \ell(p, y_t)}{\partial p}\right|_{p=w_t^\top x_t} x_t$$

Online Gradient Descent – Linear Model

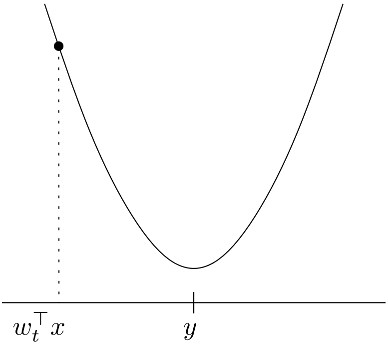
- Loss $\ell(w_t^\top x_t, y_t)$

$$w_{t+1} = w_t - \eta_t \left.\frac{\partial \ell(p, y_t)}{\partial p}\right|_{p=w_t^\top x_t} x_t$$

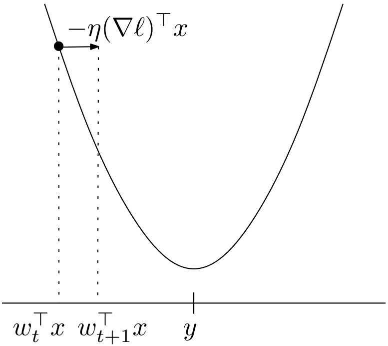
- Naive approach: define loss $i_t \ell(w_t^\top x_t, y_t)$

$$w_{t+1} = w_t - \eta_t i_t \left.\frac{\partial \ell(p, y_t)}{\partial p}\right|_{p=w_t^\top x_t} x_t$$

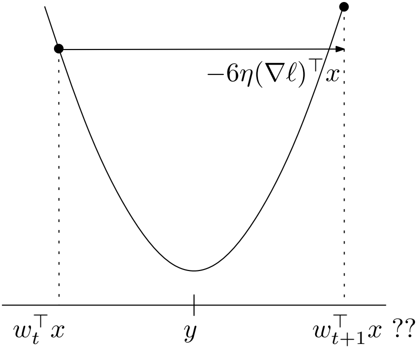
Failure of Naive Approach



Failure of Naive Approach



Failure of Naive Approach



Our principle

Principle

Having an example with importance weight i should be equivalent to having the example i times in the dataset.

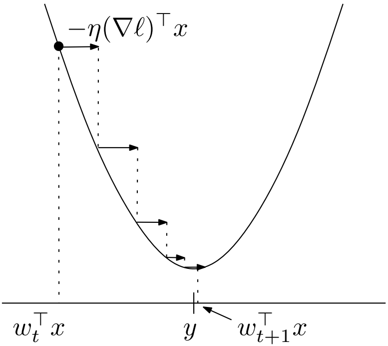
Our principle

Principle

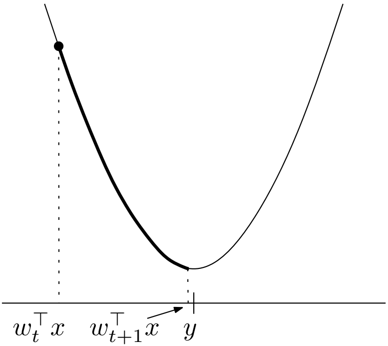
Having an example with importance weight i should be equivalent to having the example i times in the dataset.

- Present the t-th example i_t times in a row.
- Cumulative effect of this process?
- Limit of this process?

Multiple updates



Multiple updates



Importance Invariant Updates

- Gradients point to the same direction $\nabla \ell = \frac{\partial \ell(p, y_t)}{\partial p} x_t$
- w_{t+1} is in span of w_t and x_t. Letting $p = w_t^\top x_t$

$$w_{t+1} = w_t - s(p, i_t) x_t$$

Importance Invariant Updates

- Gradients point to the same direction $\nabla\ell = \frac{\partial \ell(p, y_t)}{\partial p} x_t$
- w_{t+1} is in span of w_t and x_t. Letting $p = w_t^\top x_t$

$$w_{t+1} = w_t - s(p, i_t) x_t$$

- Scaling needs to satisfy:

$$\frac{\partial s}{\partial i} = \eta \left. \frac{\partial \ell(\rho, y)}{\partial \rho} \right|_{\rho = (w_t - s(p,i)x)^\top x} \qquad s(\rho, 0) = 0$$

- Invariance:
$s(p, a+b) = s(p, a) + s(p - s(p, a)||x||^2, b)$

Importance Invariant Updates

- Gradients point to the same direction $\nabla \ell = \frac{\partial \ell(p, y_t)}{\partial p} x_t$
- w_{t+1} is in span of w_t and x_t. Letting $p = w_t^\top x_t$

$$w_{t+1} = w_t - s(p, i_t) x_t$$

- Scaling needs to satisfy:

$$\frac{\partial s}{\partial i} = \eta \left. \frac{\partial \ell(\rho, y)}{\partial \rho} \right|_{\rho = (w_t - s(p,i)x)^\top x} \qquad s(\rho, 0) = 0$$

- Invariance:
  $s(p, a+b) = s(p, a) + s(p - s(p, a)||x||^2, b)$
- OGD: just an Euler integrator — Paul Mineiro

Closed Form for Many Losses

Loss	$\ell(p, y)$	Update $s(p, i)$
Squared	$(y - p)^2$	$\frac{p-y}{x^\top x}\left(1 - e^{-i\eta x^\top x}\right)$
Logistic	$\log(1 + e^{-yp})$	$\frac{W(e^{i\eta x^\top x + yp + e^{yp}}) - i\eta x^\top x - e^{yp}}{y x^\top x}$
Hinge	$\max(0, 1 - yp)$	$-y \min\left(i\eta, \frac{1-yp}{x^\top x}\right)$

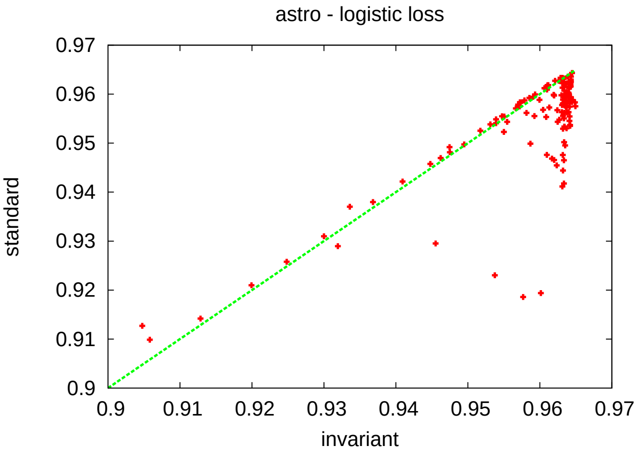
- Other losses with closed form updates: exponential, logarithmic, Hellinger, τ-quantile, and others.
- Interesting even if $i = 1$ (satisfies regret guarantee)
- Safety: update never overshoots.

Other ways of dealing with large importance weights

- Sampling: slow, inefficient
- Go through the data many times: very slow
- Solve $w_{t+1} = \operatorname{argmin} \frac{1}{2}||w - w_t||^2 + i\eta\ell(w^\top x_t, y_t)$
  - Known as *implicit* updates
  - Qualitatively very similar
  - Safe, regret guarantee
  - Typically not closed form, not invariant
- Replace ℓ above with quadratic approximation
  - works for some losses

Experiments with small weights

Do the importance invariant updates help when $i_t = 1$?



Almost eliminates search for good learning rate

Experiments with Active Learning

Importance Weighted Active Learning

$w_1 = 0$

for $t = 1, \ldots, T$

1. Receive unlabeled example x_t.
2. Predict $\text{sign}(w_t^\top x_t)$.
3. Choose a probability of labeling p_t.
   - Let $w' = \text{argmin}_{w:\text{sign}(w^\top x_t) \neq \text{sign}(w_t^\top x_t)} ||w - w_t||^2$.
   - Let Δ_t be error rate difference between w_t and w'_t.
   - $p_t = \min\left\{1, O\left(\left(\frac{1}{\Delta_t^2} + \frac{1}{\Delta_t}\right)\frac{\log t}{t}\right)\right\}$
4. With probability p_t get label y_t, and update w_t with $(x_t, y_t, \frac{1}{p_t})$. Otherwise $w_{t+1} = w_t$

Estimating Difference in Error Rates

- Suppose x_t doesn't have the label preferred by w_t
- Let $y_a = -\operatorname{sign}(w_t^\top x_t)$.
- Find i_t s.t. $w_{t+1} = w_t'$ after (x_t, y_a, i_t) update

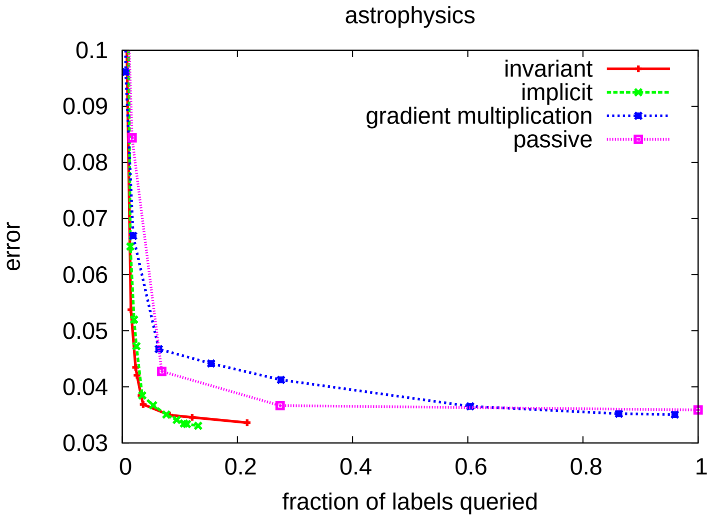
Estimating Difference in Error Rates

- Suppose x_t doesn't have the label preferred by w_t
- Let $y_a = -\operatorname{sign}(w_t^\top x_t)$.
- Find i_t s.t. $w_{t+1} = w_t'$ after (x_t, y_a, i_t) update
- For example, for logistic loss with invariant updates

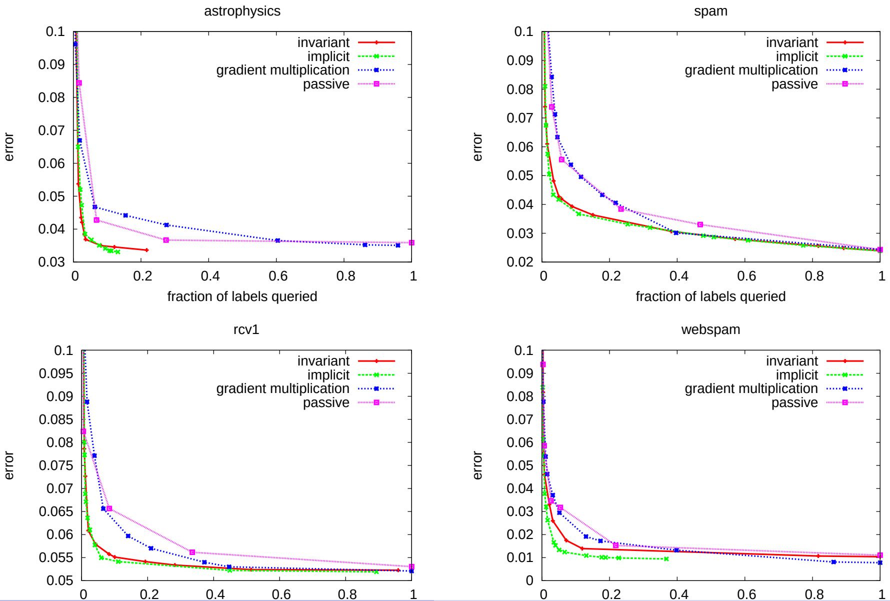
$$i_t = \frac{1 - e^{y_a w_t^\top x_t} - y_a w_t^\top x_t}{\eta_t}$$

- Then $\Delta_t \approx i_t / t$

Active Learning Results



Active Learning Results



How fast is it?

- Demo on RCV1 ($\approx$780K docs 77 features/doc) ...

How fast is it?

- Demo on RCV1 ($\approx$780K docs 77 features/doc) ...
- As fast as (passive) online gradient descent
  - Active learning takes 2.6 sec.
  - Passive online gradient descent takes 2.5 sec
  - 91K queries (11%)

Conclusions

- New updates from first principles
- You should use them because they
  - work well, even for $i = 1$
  - are robust w.r.t. learning rate
  - are fast (closed form)
  - have useful properties (invariance, safety)
  - are simple to code
- Check out implementation in Vowpal Wabbit
  `http://github.com/JohnLangford/vowpal_wabbit`