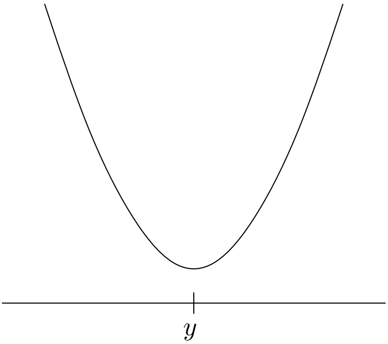
# Examples with importance weights

- Sometimes some examples are more important.
- Importance weights pop up in: boosting, differing train/test distributions, active learning, etc.
- John can reduce everything to importance weighted binary classification.

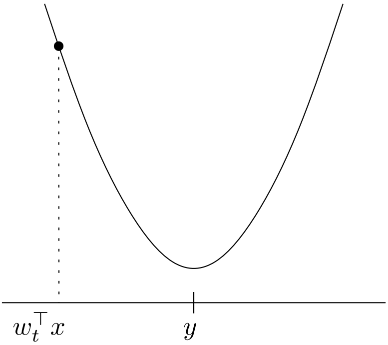
## Principle

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

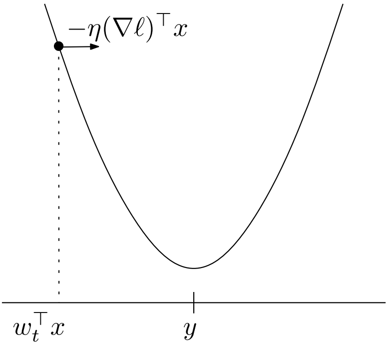
# Learning with importance weights



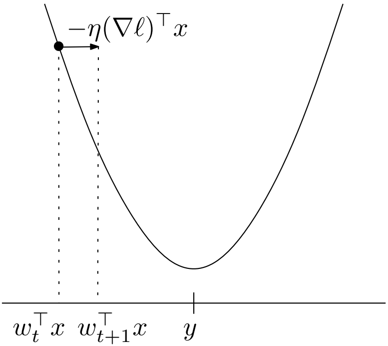
# Learning with importance weights



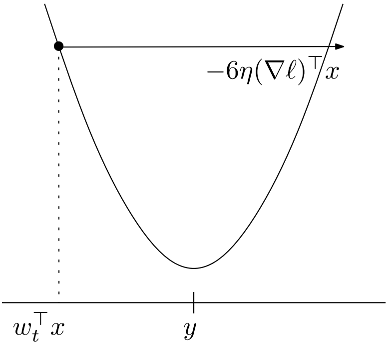
# Learning with importance weights



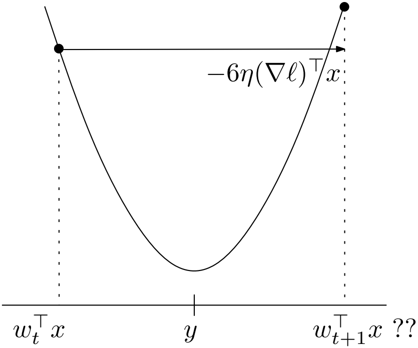
# Learning with importance weights



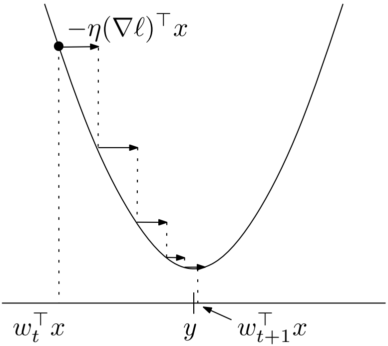
# Learning with importance weights



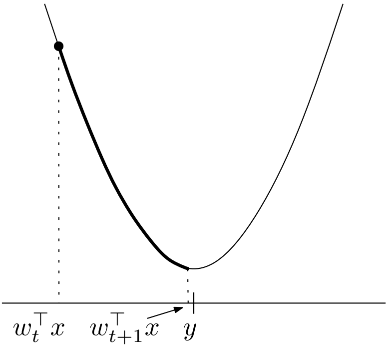
# Learning with importance weights



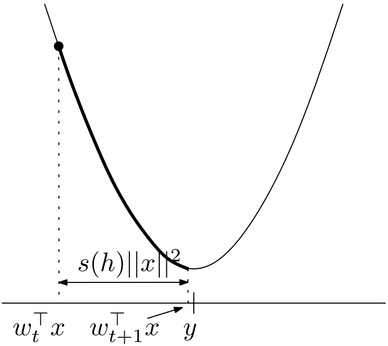
# Learning with importance weights



# Learning with importance weights



# Learning with importance weights



# What is $s(\cdot)$?

- Losses for linear models $\ell(w^\top x, y)$. $\nabla_w \ell = \frac{\partial \ell(p,y)}{\partial p} x$
- Update must be given by

$$w_{t+1} = w_t - s(h)x$$

- $s(h)$ must satisfy

$$s(h+\epsilon) = s(h) + \epsilon\eta \left.\frac{\partial \ell(p,y)}{\partial p}\right|_{p=(w_t - s(h)x)^\top x}$$

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

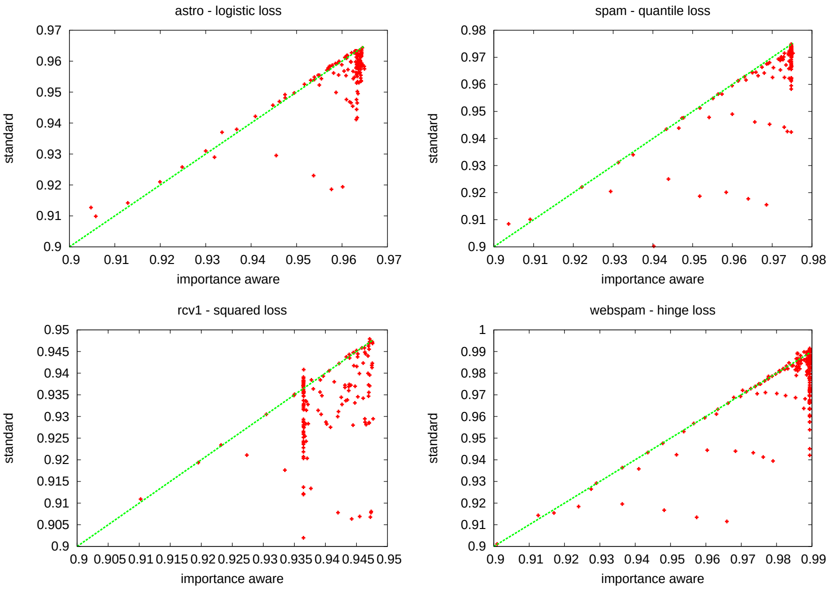
Finally

$$s(0) = 0$$

# Many loss functions

| Loss | $\ell(p, y)$ | Update $s(h)$ |
|---|---|---|
| Squared | $(y - p)^2$ | $\frac{p-y}{x^\top x}\left(1 - e^{-h\eta x^\top x}\right)$ |
| Logistic | $\log(1 + e^{-yp})$ | $\frac{W(e^{h\eta x^\top x + yp + e^{yp}}) - h\eta x^\top x - e^{yp}}{y x^\top x}$ for $y \in \{-1, 1\}$ |
| Exponential | $e^{-yp}$ | $\frac{py - \log(h\eta x^\top x + e^{py})}{x^\top x y}$ for $y \in \{-1, 1\}$ |
| Logarithmic | $y \log \frac{y}{p} + (1-y) \log \frac{1-y}{1-p}$ | if $y = 0$ $\quad \frac{p - 1 + \sqrt{(p-1)^2 + 2h\eta x^\top x}}{x^\top x}$<br>if $y = 1$ $\quad \frac{p - \sqrt{p^2 + 2h\eta x^\top x}}{x^\top x}$ |
| Hellinger | $(\sqrt{p} - \sqrt{y})^2 - (\sqrt{1-p} - \sqrt{1-y})^2$ | if $y = 0$ $\quad \frac{p - 1 + \frac{1}{4}(12h\eta x^\top x + 8(1-p)^{3/2})^{2/3}}{x^\top x}$<br>if $y = 1$ $\quad \frac{p - \frac{1}{4}(12h\eta x^\top x + 8p^{3/2})^{2/3}}{x^\top x}$ |
| Hinge | $\max(0, 1 - yp)$ | $-y \min\left(h\eta, \frac{1-yp}{x^\top x}\right)$ for $y \in \{-1, 1\}$ |
| $\tau$-Quantile | if $y > p$ $\quad \tau(y - p)$<br>if $y \le p$ $\quad (1-\tau)(p - y)$ | if $y > p$ $\quad -\tau \min(h\eta, \frac{y-p}{\tau x^\top x})$<br>if $y \le p$ $\quad (1-\tau)\min(h\eta, \frac{p-y}{(1-\tau) x^\top x})$ |

# Robust results for unweighted problems



# And now something completely different

- Adaptive, individual learning rates in VW.
- It's really GD separately on each coordinate $i$ with

$$\eta_{t,i} = \frac{1}{\sqrt{\sum_{s=1}^{t} \left( \frac{\partial \ell(w_s^\top x_s, y_s)}{\partial w_{s,i}} \right)^2}}$$

- Coordinate-wise scaling of the data less of an issue
- Can state this formally (Duchi, Hazan, and Singer / McMahan and Streeter, COLT 2010)

# Some tricks involved

- Store sum of squared gradients w.r.t $w_i$ near $w_i$.
- 
```
float InvSqrt(float x){
  float xhalf = 0.5f * x;
  int i = *(int*)&x;
  i = 0x5f3759d5 - (i >> 1);
  x = *(float*)&i;
  x = x*(1.5f - xhalf*x*x);
  return x;
}
```
  Special SSE rsqrt instruction is a little better

# Experiments

- Raw Data

```
./vw --adaptive -b 24 --compressed -d tmp/spam_train.gz
average loss = 0.02878
./vw -b 24 --compressed -d tmp/spam_train.gz -l 100
average loss = 0.03267
```

- TFIDF scaled data

```
./vw --adaptive -b 24 --compressed -d tmp/rcv1_train.gz -l 1
average loss = 0.04079
./vw -b 24 --compressed -d tmp/rcv1_train.gz -l 256
average loss = 0.04465
```