Online Learning: A Brief Intro.

Chen Huang

Data Mining Lab,
Big Data Research Center, UESTC
huangc.uestc@gmail.com

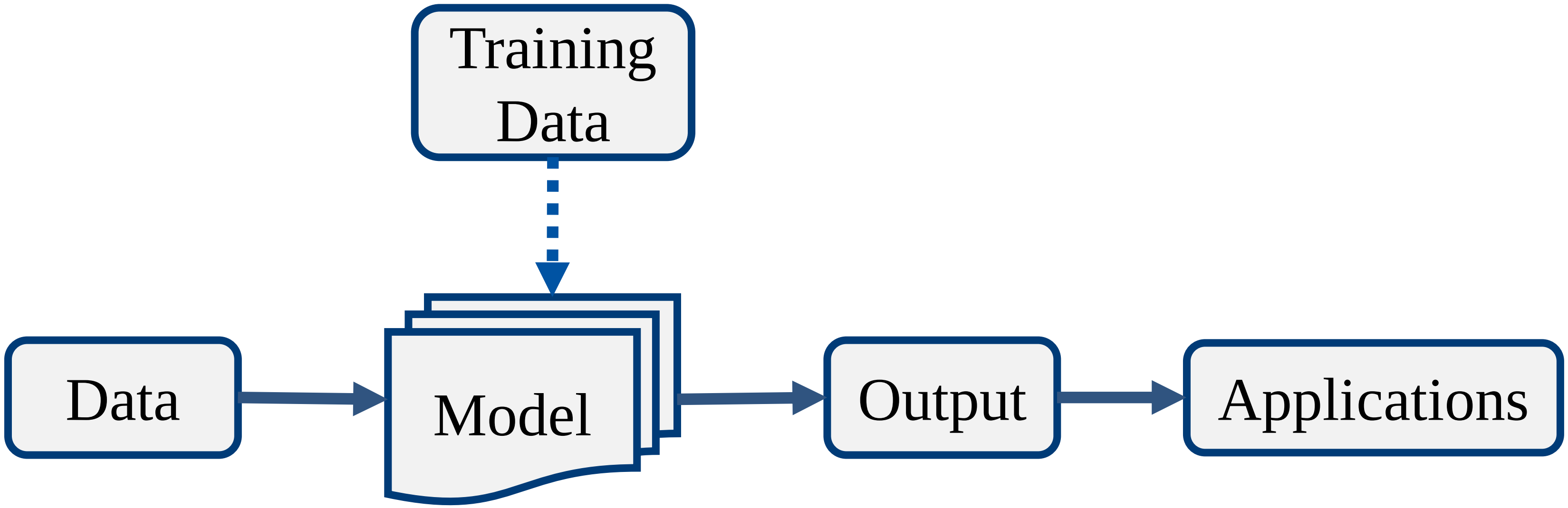
Outline

- Online learning
- Online learning methods
- Applications $_{\text{list}}$
- Conclusion
- Clues for my next work?

内容杂而泛，有兴趣的同学可以线下深入了解~

Offline Learning

Traditional (offline) learning

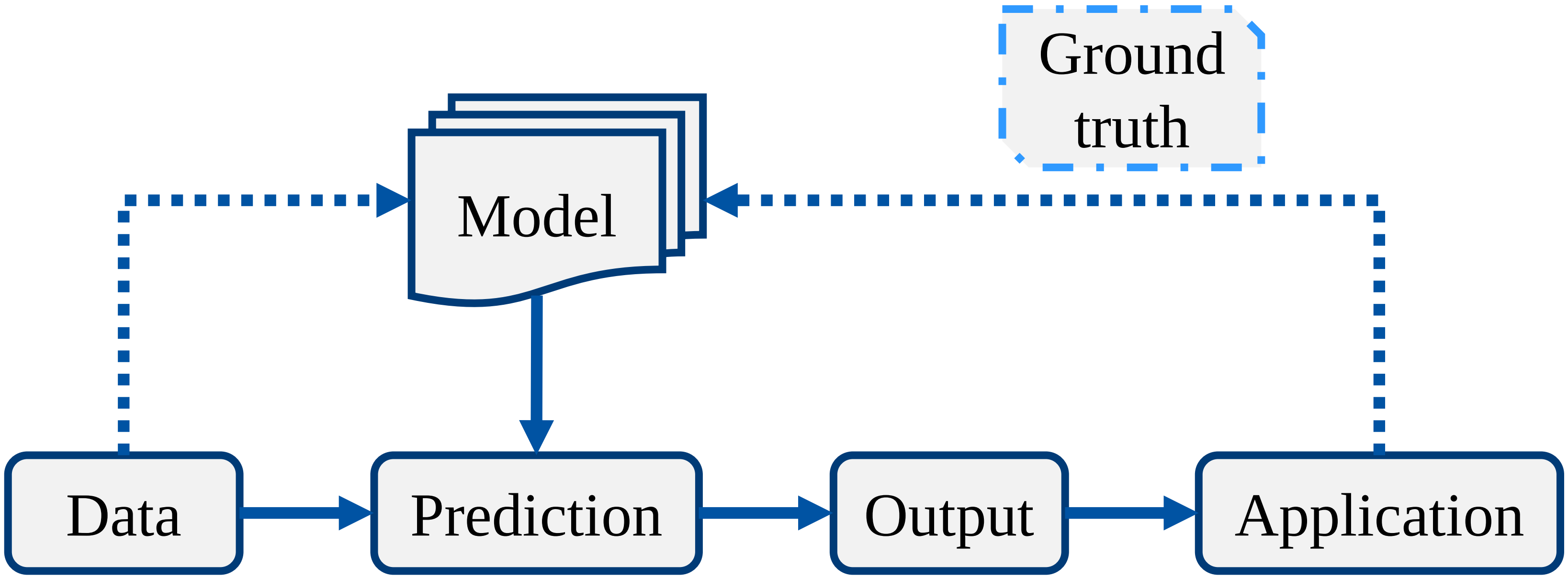


Challenge: Real-time stream data

– Evolving / Concept drift
– Constraints in terms of memory and running time
– Tradeoff between Accuracy and Efficiency
– Distributed application and Result visualization

Online Learning

Online learning



Model update, real-time, scalability

Online Learning

Online learning task

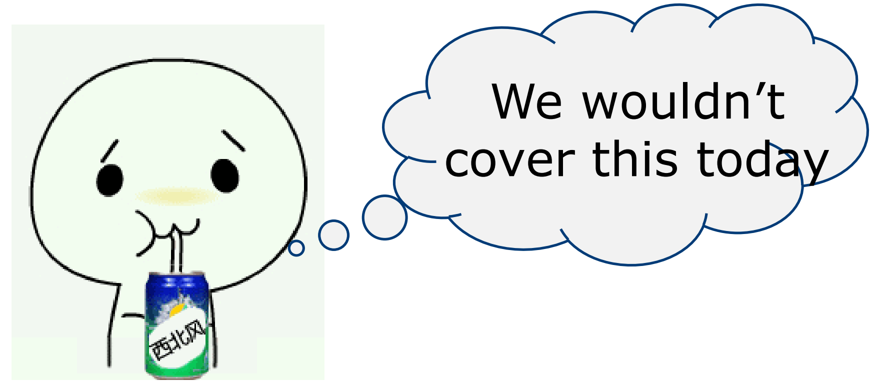
For t=1, 2, …, **T**

- Receive $\mathbf{x}_t$
- Predict $\widehat{y}_t = \mathrm{sgn}(f_t(\mathbf{x}_t))$
- Receive y_t
- Suffer loss $\ell(y_t, f_t(\mathbf{x}_t))$
- Update $f_t(\mathbf{x}) \rightarrow f_{t+1}(\mathbf{x})$

Goal: To minimize:

$$\sum_{t=1}^{T} \ell(y_t, f_t(\mathbf{x}_t))$$

Theoretical Analysis



Regret analysis

- Given all the data, we could find the optimal classifier, denoted as

$$f^* = \arg\min_{f \in H} \sum_{t=1}^{T} L\big(y_t, f(x_t)\big)$$

- Online learning **regrets** that why wouldn't I choose the f^* at the first place.

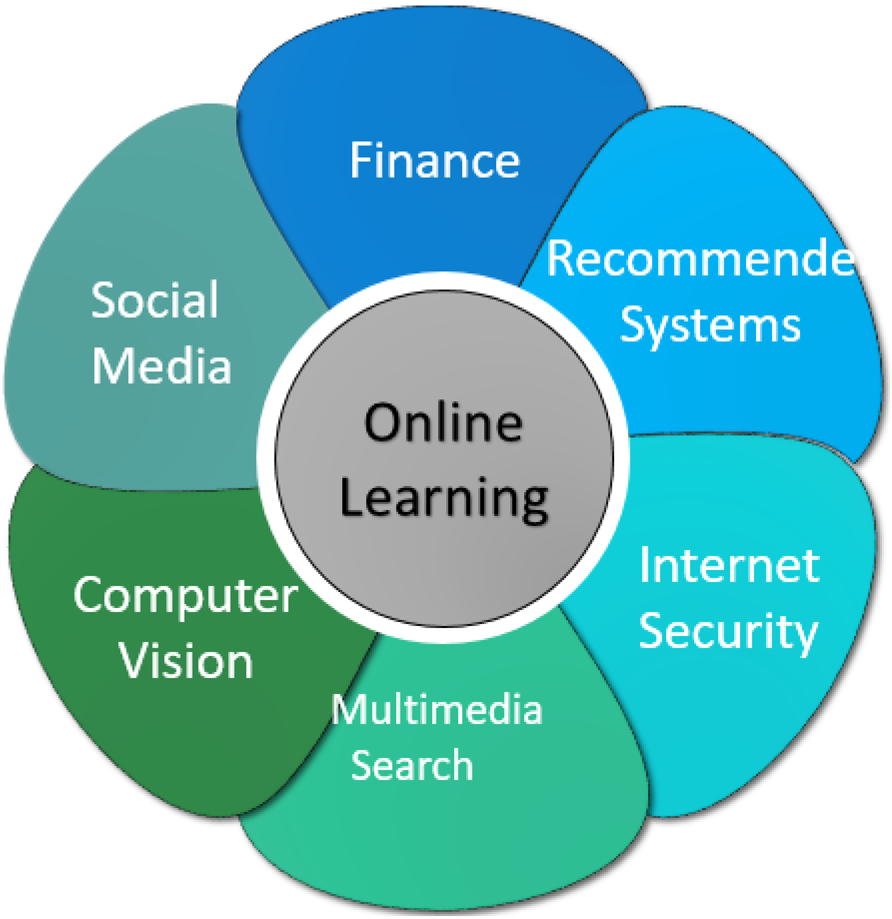
$$regret = \frac{1}{T} \sum_{t=1}^{T} \Big(L\big(y_t, f_t(x_t)\big) - L\big(y_t, f^*(x_t)\big)\Big)$$

- We wish the regret to be small and bounded, and it's **no-regret** if

$$\lim_{T \to \infty} \frac{regret(T)}{T} \to 0$$

Online Learning

Applications



Online Learning

Online update

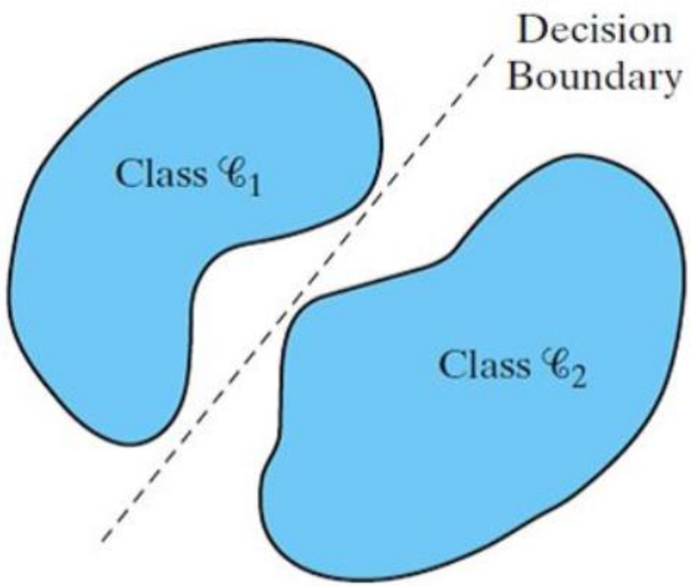
- **When to update model?**
  - Mistake driven
  - Confidence in prediction
  - ……
- **How to update model?**
  - Re-training ? ✖
  - Basically,

$$W_t = W_{t-1} + \Delta$$

Linear Classifier Revisit
From Batch to Online

Perceptron

– Minimize the sum of the **functional margins** (-_-!) of those misclassified data points.



$$f(x) = sign(wx + b)$$

$$sign(\mathrm{x}) = \begin{cases} -1, x < 0 \\ +1, x \geq 0 \end{cases}$$

$$L(w, b) = -\sum_{x_i \in M} y_i(wx_i + b)$$

Stochastic Gradient Descent Revisit

SGD

– **Stochastic approximation** of the gradient descent method for minimizing an objective function that is written as a sum of **differentiable sub-functions**:

$$\min \sum_{i=1}^{m} f_i(x)$$

$$\boldsymbol{SGD}: \quad x^{(k)} = x^{(k-1)} - t_k g_r^{(k-1)}(x)$$

$$\boldsymbol{GD}: \quad x^{(k)} = x^{(k-1)} - t_k \sum_{i=1}^{m} g_i^{(k-1)}(x) \quad where\ g_i^{(k-1)} \in \partial f_i^{(k-1)}$$

Perceptron Revisit

SGD for perceptron

– **Objective**

$$L(w,b) = -\sum_{x_i \in M} y_i (w x_i + b)$$

– **Solver**

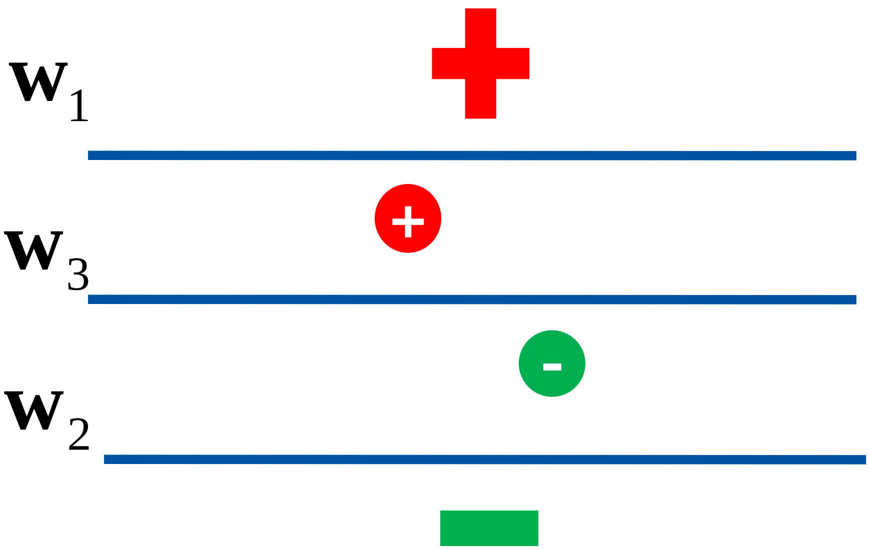
$$\nabla_w L(w,b) = -\sum_{x_i \in M} y_i x_i \qquad \nabla_b L(w,b) = -\sum_{x_i \in M} y_i$$

– **Update**

$$w \leftarrow w + \gamma y_i x_i \qquad b \leftarrow b + \gamma y_i$$

Perceptron Revisit

SGD for perceptron



$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}_t$$

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}_t$$

Online Perceptron

SGD for online update

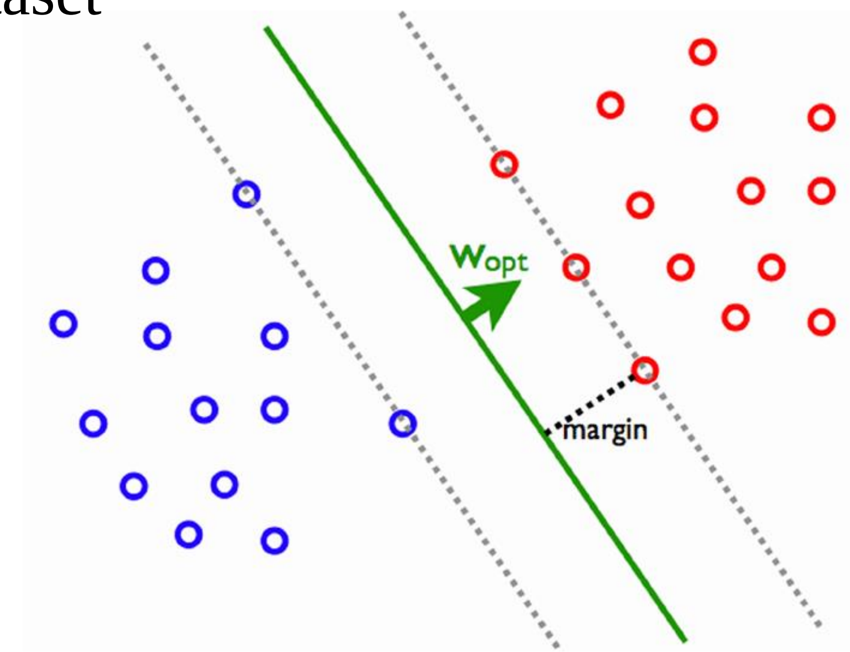
1. Start with the all-zeroes weight vector $\mathbf{w}_1 = \mathbf{0}$, and initialize t to 1.
2. Given example $\mathbf{x}$, predict positive iff $\mathbf{w}_t \cdot \mathbf{x} > 0$.
3. On a mistake, update as follows:
   - Mistake on positive: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}$.
   - Mistake on negative: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}$.

   $t \leftarrow t + 1$.

Online Perceptron

Novikoff Theorem

- For linearly separable dataset
  - Margin is γ
  - $\|x\| \leq R$
- Then,
  - $\#mistakes \leq \left(\frac{R}{\gamma}\right)^2$
  - #update = #mistakes

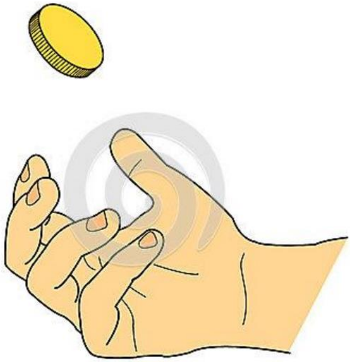


Bayesian Conjugate Revisit

Conjugate prior

– If the posterior distributions $p(\theta|x)$ are in the same family as the prior distribution $p(\theta)$, the prior and posterior are then called **Conjugate Distributions**

– The prior is called a **Conjugate Prior** for the likelihood function



Example: Toss a coin

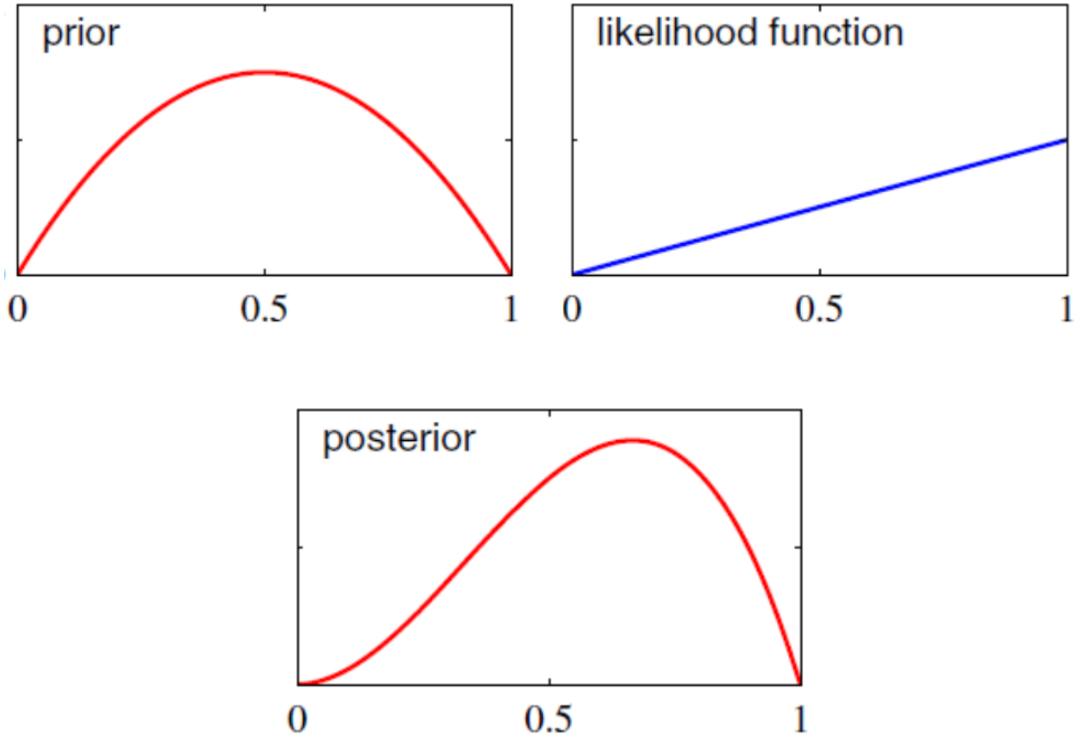
Priori: $Beta(\alpha, \beta)$

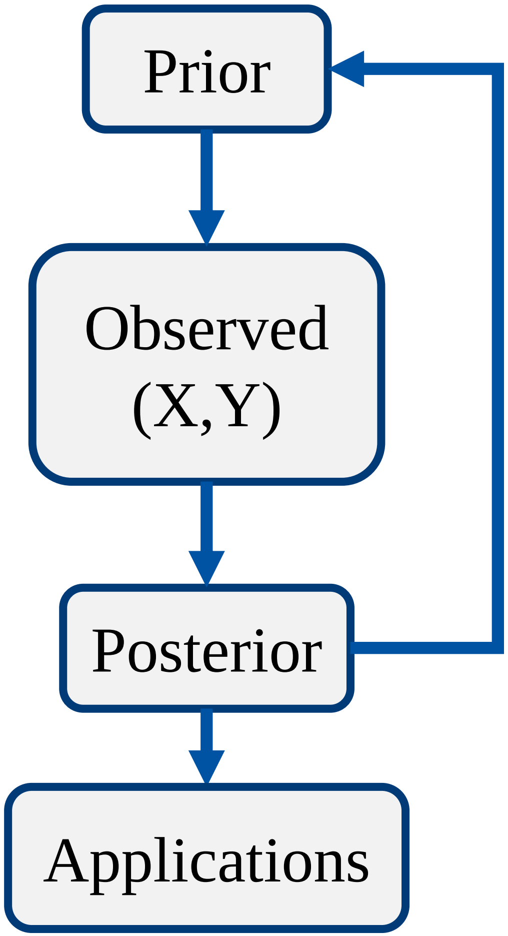
Likelihood: $Bernoulli(p)$

Posteriori: $Beta(\alpha + \text{he}ads, \beta + tails)$

Bayesian Online Learning

Sequential update





*Bayesian Online Learning For Non-conjugate Prior

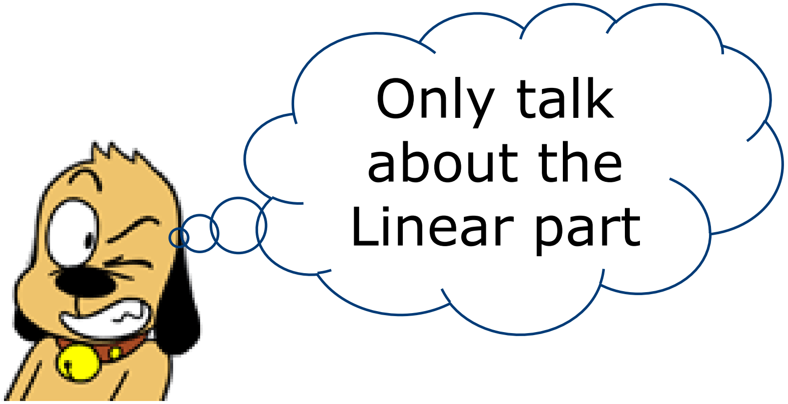
*Online Bayesian Probit Regression

– **Linear Gaussian model** (*Kalman Filter*) with $Y_t = \{1, -1\}$

	$P(X_t \mid X_{t-1})$	$P(Y_t \mid X_t)$	$P(X_0)$	**Example**
Discrete State DM	矩阵形式	Any	π	Hidden Markov Model
Linear Gaussian DM	$N(AX_{t-1} + B, Q)$	$N(HX_t + C, R)$	$N(\mu_0, \varepsilon_0)$	Kalman Filter
Non-linear Non-Gaussian DM	$f(X_{t-1})$	$g(X_t)$	$f(X_0)$	Particle Filter

– **KL divergence** to approximate Gaussian posterior

Online learning methods



Overview

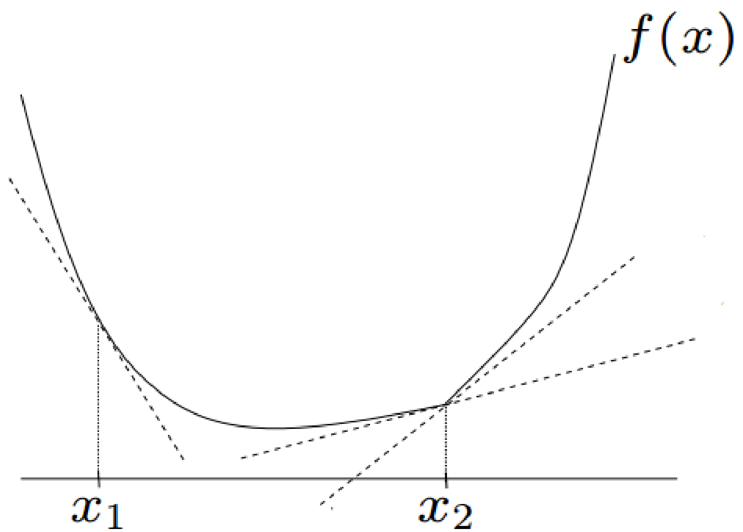
- **Linear methods**
  - ✓ First-order algorithms (Perceptron, Passive-Aggressive)
  - ✓ Second-order algorithms (Confidence weighted)
  - ✓ Sparse online learning algorithms (FOBOS, RDA, FTRL)
- **Non-linear methods**
  - ✓ Kernel based online learning (Kernel perceptron)
  - ✓ Local online learning
  - ✓ Deep online learning (-_-!!!)
- ***Multiclass online learning**
- ***Centralized/decentralized distributed online learning**

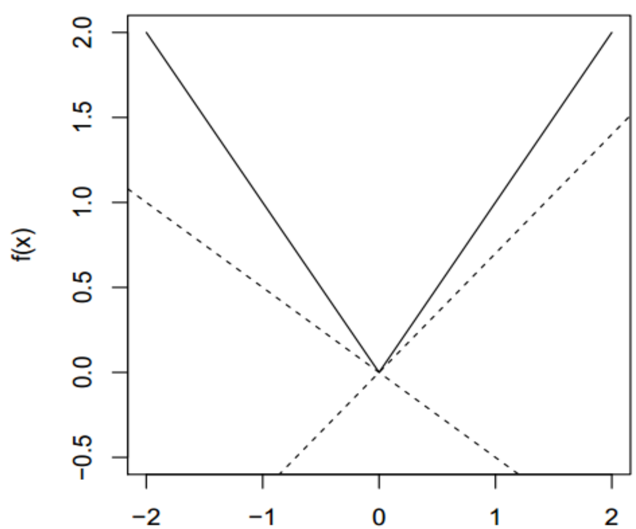
Prior Knowledge Revisit

Subgradient

– g is a **subgradient** of f (not necessarily convex) at x if

$$f(y) \geq f(x) + \nabla g^T \ (y - x) \qquad \forall y$$





$$f(x^\star) = \inf_x f(x) \Longleftrightarrow 0 \in \partial f(x^\star)$$

Prior Knowledge Revist

Strong duality and KKT

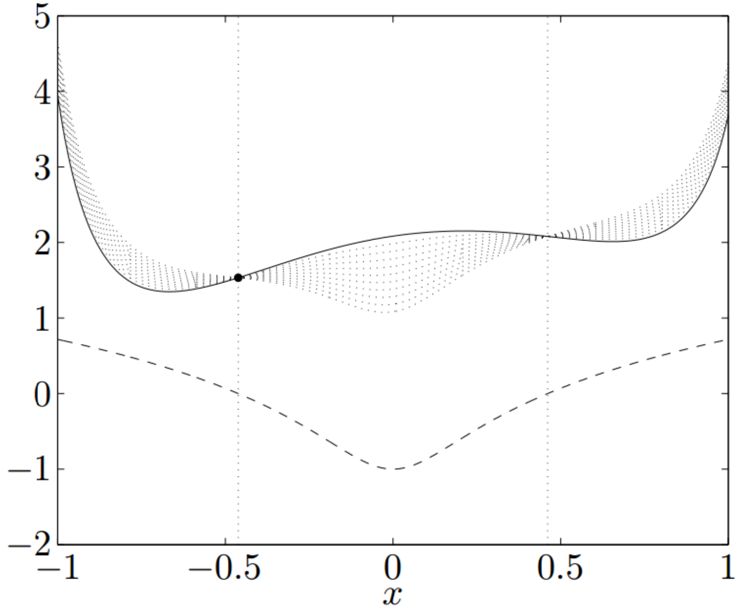
$$\textbf{Stationarity}: \ 0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial l_j(x)$$

$$\textbf{Complementary}: \ u_i h_i(x) = 0 \quad for\ all\ i$$

$$\textbf{Primal feasibility}: \ h_i(x) \le 0, \ \ l_j(x) = 0 \quad for\ all\ i, j$$

$$\textbf{Dual feasibility}: \ u_i \ge 0 \ \ for\ all\ i$$

$$\begin{aligned} &\min \ \ f(x) \\ s.t.\ \ & h_i(x) \le 0, \ \ i = 1 \ldots, m \\ & l_j(x) = 0, \ \ \ j = 1 \ldots . r \end{aligned}$$

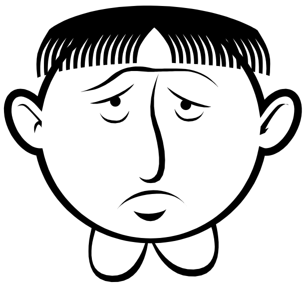


First-order Methods

Passive-Aggressive learning *(JMLR 2006)*

– Utilizes the **margin** to modify the current classifier. The update of the classifier is performed by solving a constrained optimization problem

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) = 0.$$



– **Passive** when hinge loss is zero, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$ *or*

– **Aggressively** forces w_{t+1} to satisfy the constraint $\ell(w_{t+1};(x_t,y_t)) = 0$ regardless of the step-size required.

First-order Methods

$$s.t.\ \ h_i(x) \le 0,\ \ i = 1 \dots, m$$
$$l_j(x) = 0,\ \ j = 1 \dots . r$$

KKT for PA problem

- **Convex Problem + *Slater's condition***
- Finding the problem's optimum is equivalent to satisfying the KKT condition
- So, for the aggressive part

$$\mathcal{L}(\mathbf{w},\tau) = \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big)$$

$$0 = \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w},\tau) = \mathbf{w}-\mathbf{w}_t-\tau y_t\mathbf{x}_t \quad\Longrightarrow\quad \mathbf{w} = \mathbf{w}_t+\tau y_t\mathbf{x}_t .$$

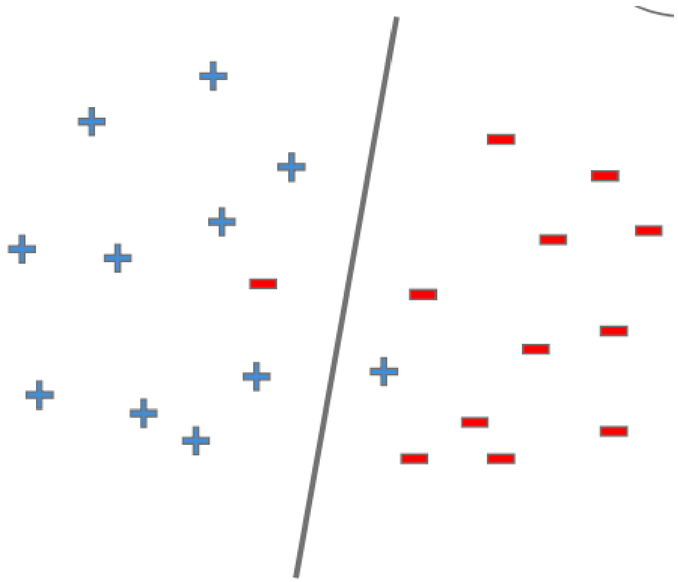
$$\mathcal{L}(\tau) = -\frac{1}{2}\tau^2\|\mathbf{x}_t\|^2 + \tau\big(1-y_t(\mathbf{w}_t\cdot\mathbf{x}_t)\big)$$

$$0 = \frac{\partial\mathcal{L}(\tau)}{\partial\tau} = -\tau\|\mathbf{x}_t\|^2 + \big(1-y_t(\mathbf{w}_t\cdot\mathbf{x}_t)\big) \quad\Longrightarrow\quad \tau = \frac{1-y_t(\mathbf{w}_t\cdot\mathbf{x}_t)}{\|\mathbf{x}_t\|^2}.$$

First-order Methods

Label noise: PA-I & PA-II

– Recall soft margin of SVM



$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) \leq \xi \text{ and } \xi \geq 0.$$

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi^2 \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) \leq \xi.$$

First-order Methods

PA & PA-I & PA-II

– Closed-form update

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2} \qquad \text{(PA)}$$

$$\tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\} \qquad \text{(PA-I)}$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}} \qquad \text{(PA-II)}$$

INPUT: aggressiveness parameter $C > 0$
INITIALIZE: $\mathbf{w}_1 = (0, \ldots, 0)$
For $t = 1, 2, \ldots$

- receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
- predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$
- receive correct label: $y_t \in \{-1, +1\}$
- suffer loss: $\ell_t = \max\{0\ ,\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$
- update:

  1. set:

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2} \qquad \text{(PA)}$$

$$\tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\} \qquad \text{(PA-I)}$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}} \qquad \text{(PA-II)}$$

  2. update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$

First-order Methods

Meaning behind the update

– Closed-form update

Step size

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2} \quad \text{(PA)}$$

$$\tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\} \quad \text{(PA-I)}$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}} \quad \text{(PA-II)}$$

Mistake driven step size

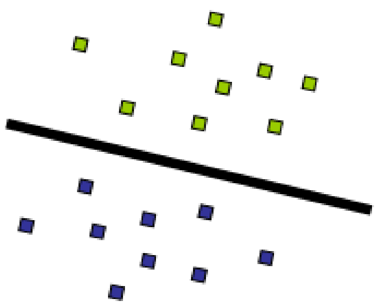
Mistake driven step size with a fixed upper bound

PA update with on new x_t (increasing dimension from m to $m + T$ with $x_{m+t} = \sqrt{1/(2C)}$ and the remaining $T - 1$ to zero)

First-order Methods

PA for other tasks

Classification

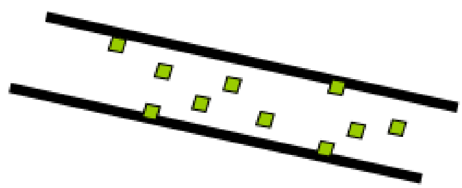


$$y_t \mathbf{w} \cdot \mathbf{x}_t \geq \tilde{\epsilon}$$

$$\mathbf{z}_t = (\mathbf{x}_t, y_t)$$

$$(\mathbf{x}_t \in \mathcal{R}^n, y_t \in \{-1, 1\})$$

Regression

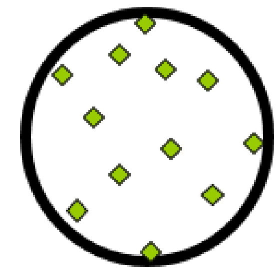


$$|\mathbf{w} \cdot \mathbf{x}_t - y_t| \leq \tilde{\epsilon}$$

$$\mathbf{z}_t = (\mathbf{x}_t, y_t)$$

$$(\mathbf{x}_t \in \mathcal{R}^n, y_t \in \mathcal{R})$$

Uniclass



$$\|\mathbf{y}_t - \mathbf{w}\| \leq \tilde{\epsilon}$$

$$\mathbf{z}_t = \mathbf{y}_t$$

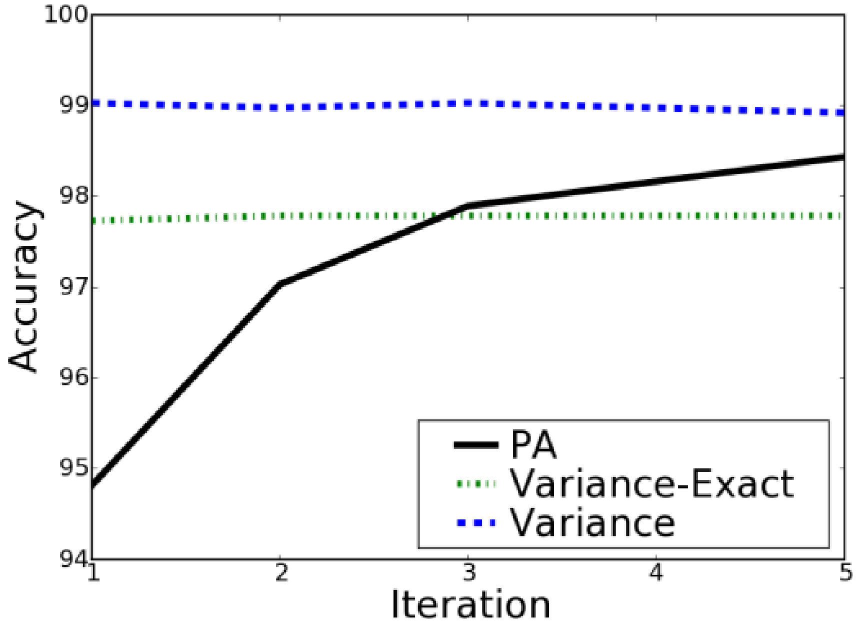
$$\mathbf{y}_t \in \mathcal{R}^n$$

First-order Methods

Pros and Cons

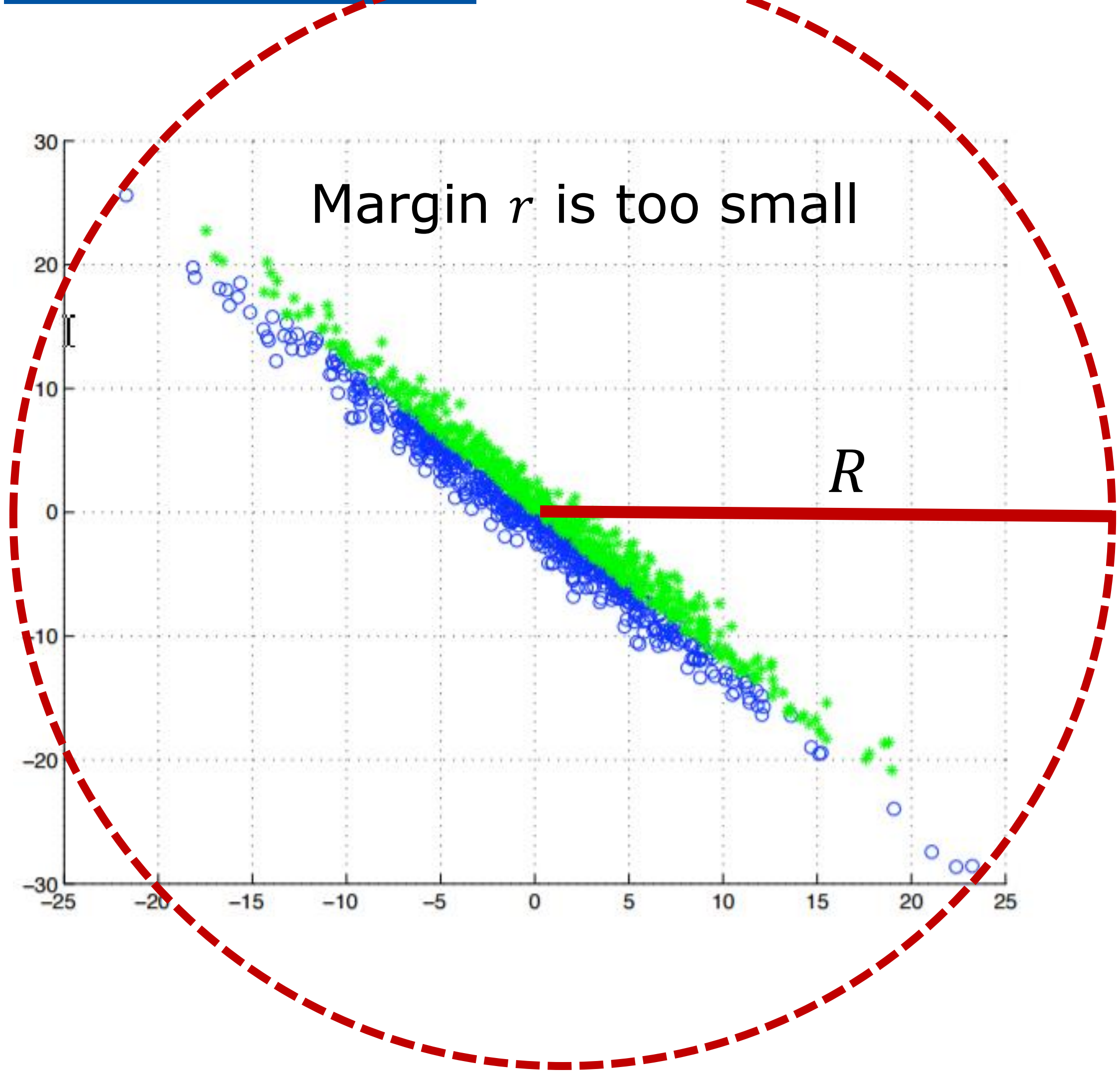
☺Simple and easy to implement
☺ Efficient and scalable for high-dimensional data

☹ Relatively slow convergence rate



First-order Methods

Pros and Cons



$$\#mistakes \leq \left(\frac{R}{\gamma}\right)^2$$

Second-order Methods

Confidence weighted learning *(ICML 2008)*

- Add parameter confidence to linear classifiers
- **Less confident parameters are updated more aggressively than more confident ones**

$$w \sim N(\mu, \Sigma)$$

μ_j: $parameter\ knowledge$

Σ_{jj}: $confidence$ $(\Sigma_{ij} = 0)$

- Parameter confidence is updated for each new training instance so that the probability of correct classification for that instance under the updated distribution meets a specified confidence.

Second-order Methods

Confidence weighted learning *(ICML 2008)*

- Linear classifier

$$y = w \cdot x \qquad w \sim N(\mu, \Sigma)$$

- Margin M

$$y_i(w \cdot x_i)$$
$$M \sim N\left(y_i(\mu \cdot x_i), x_i^T \Sigma x_i\right)$$

- Recall PA

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0.$$

- Correct prediction for CW

$$\Pr_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)} [M \geq 0] = \Pr_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)} [y_i (\boldsymbol{w} \cdot \boldsymbol{x}_i) \geq 0]$$

Second-order Methods

Confidence weighted learning *(ICML 2008)*

– Recall PA

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) = 0.$$

– CW

$$(\boldsymbol{\mu}_{i+1}, \Sigma_{i+1}) = \min \mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{\mu},\Sigma) \,\|\, \mathcal{N}(\boldsymbol{\mu}_i,\Sigma_i)\right)$$
$$\text{s.t. } \Pr\left[y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i\right) \geq 0\right] \geq \eta \;.$$

– Further form (**Never expected**)

$$\Pr\left[M \leq 0\right] = \Pr\left[\boxed{\frac{M-\mu_M}{\sigma_M}} \leq \frac{-\mu_M}{\sigma_M}\right]$$

$N(0,1)$

Second-order Methods

Confidence weighted learning *(ICML 2008)*

- CW

$$(\boldsymbol{\mu}_{i+1}, \Sigma_{i+1}) = \min \mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right) \,\|\, \mathcal{N}\left(\boldsymbol{\mu}_i, \Sigma_i\right)\right)$$
$$\text{s.t. } \Pr\left[y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i\right) \geq 0\right] \geq \eta \ .$$

- Further form (**Never expected**)

$N(0,1)$

$$\Pr\left[M \leq 0\right] = \Pr\left[\boxed{\frac{M - \mu_M}{\sigma_M}} \leq \frac{-\mu_M}{\sigma_M}\right]$$

$$\frac{-\mu_M}{\sigma_M} \leq \Phi^{-1}\left(1 - \eta\right) = -\Phi^{-1}\left(\eta\right)$$

$$y_i(\boldsymbol{\mu} \cdot \boldsymbol{x}_i) \geq \phi \sqrt{\boldsymbol{x}_i^\top \Sigma \boldsymbol{x}_i} \qquad \phi = \Phi^{-1}\left(\eta\right)$$

Second-order Methods

Confidence weighted learning *(ICML 2008)*

– CW

$$(\boldsymbol{\mu}_{i+1}, \Sigma_{i+1}) = \min \frac{1}{2} \log\left(\frac{\det \Sigma_i}{\det \Sigma}\right) + \frac{1}{2} \mathrm{Tr}\left(\Sigma_i^{-1}\Sigma\right) + \frac{1}{2}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^\top \Sigma_i^{-1}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)$$

$$\text{s.t.} \quad y_i(\boldsymbol{\mu} \cdot \boldsymbol{x}_i) \geq \phi\left(\boldsymbol{x}_i^\top \Sigma \boldsymbol{x}_i\right) .$$

– Optimization

Second-order Methods

Confidence weighted learning *(ICML 2008)*

– Update

$$w \leftarrow w + \gamma y_i x_i$$

Algorithm 1 Variance Algorithm (Approximate)

Input: confidence parameter $\phi = \Phi^{-1}(\eta)$
initial variance parameter $a > 0$
Initialize: $\boldsymbol{\mu}_1 = \mathbf{0}$, $\Sigma_1 = aI$
for $i = 1, 2 \ldots$ **do**
Receive $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$
Set the following variables:
α_i as in Lemma 1
$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \alpha_i y_i \Sigma_i \boldsymbol{x}_i$ (11)
$\Sigma_{i+1}^{-1} = \Sigma_i^{-1} + 2\alpha_i \phi\ \mathbf{x}_i \mathbf{x}_i^\top$ (17)
end for

Large confidence, small step size

Second-order Methods

Confidence weighted learning *(ICML 2008)*

– Update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \gamma y_i \boldsymbol{x}_i$$

Algorithm 1 Variance Algorithm (Approximate)

Input: confidence parameter $\phi = \Phi^{-1}(\eta)$
initial variance parameter $a > 0$
Initialize: $\boldsymbol{\mu}_1 = \mathbf{0}$, $\Sigma_1 = aI$
for $i = 1, 2 \dots$ **do**
Receive $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$
Set the following variables:
α_i as in Lemma 1
$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \alpha_i y_i \Sigma_i \boldsymbol{x}_i$ (11)
$\Sigma_{i+1}^{-1} = \Sigma_i^{-1} + 2\alpha_i \phi \, \mathbf{x}_i \mathbf{x}_i^{\top}$ (17)
end for

$$\alpha_i = \max\{\gamma_i, 0\}$$

$$\gamma_i = \frac{-(1+2\phi M_i) + \sqrt{(1+2\phi M_i)^2 - 8\phi(M_i - \phi V_i)}}{4\phi V_i}$$

$$M_i = y_i(\boldsymbol{x}_i \cdot \boldsymbol{\mu}_i) \qquad V_i = \boldsymbol{x}_i^{\top} \Sigma_i \boldsymbol{x}_i$$

Data-driven parameters

Second-order Methods

Confidence weighted learning

- **Cons**
  - Non-separable or label noise

$$(\boldsymbol{\mu}_{i+1}, \Sigma_{i+1}) = \min \mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \,\|\, \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)\right)$$
$$\text{s.t. } \Pr\left[y_i\left(\boldsymbol{w} \cdot \boldsymbol{x}_i\right) \geq 0\right] \geq \eta \, .$$

- Adaptive Regularization of Weight Vectors (**AROW**) *(NIPS'09)*

$$\mathcal{C}(\boldsymbol{\mu}, \Sigma) = \mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \,\|\, \mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})\right) + \underbrace{\lambda_1 \ell_{\mathrm{h}^2}\left(y_t, \boldsymbol{\mu} \cdot \boldsymbol{x}_t\right)}_{\text{Squared hinge loss}} + \underbrace{\lambda_2 \boldsymbol{x}_t^\top \Sigma \boldsymbol{x}_t}_{\Sigma_{ij} \neq 0}$$

- Adaptive Regularization for Weight Matrices (**AROWA**) *(ICML'12)*
  - Handle the problem of Σ is a huge matrix

Second-order Methods

Soft confidence weighted learning *(ICML 2012)*

- **Adaptive soft margin**
- Recall CW

$$(\boldsymbol{\mu}_{i+1}, \Sigma_{i+1}) = \min \frac{1}{2} \log\left(\frac{\det \Sigma_i}{\det \Sigma}\right) + \frac{1}{2} \mathrm{Tr}\left(\Sigma_i^{-1} \Sigma\right) + \frac{1}{2} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^\top \Sigma_i^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)$$

$$\text{s.t.} \quad y_i(\boldsymbol{\mu} \cdot \boldsymbol{x}_i) \geq \phi\left(\boldsymbol{x}_i^\top \Sigma \boldsymbol{x}_i\right) \ .$$

- Adaptive hinge loss

$$\ell^{\phi}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma); (\mathbf{x}_t, y_t)\right) = \max\left(0, \boxed{\phi\sqrt{\mathbf{x}_t^\top \Sigma \mathbf{x}_t}} - y_t \boldsymbol{\mu} \cdot \mathbf{x}_t\right)$$

$$(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \arg\min_{\boldsymbol{\mu}, \Sigma} D_{KL}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\right)$$

$$s.t. \ \ell^{\phi}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma); (\mathbf{x}_t, y_t)\right) = 0, \ \phi > 0$$

Second-order Methods

Soft confidence weighted learning *(ICML 2012)*

- **Adaptive soft margin** (recall PA-I and PA-II)
- SCW-I

$$
\begin{aligned}
(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) &= \arg\min_{\boldsymbol{\mu},\Sigma} D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\big) \\
&+ C\ell^{\phi}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma); (\mathbf{x}_t, y_t)\big)
\end{aligned}
$$

- SCW-II

$$
\begin{aligned}
(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) &= \arg\min_{\boldsymbol{\mu},\Sigma} D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\big) \\
&+ C\ell^{\phi}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma); (\mathbf{x}_t, y_t)\big)^2
\end{aligned}
$$

Second-order Methods

Soft confidence weighted learning *(ICML 2012)*

Algorithm	Large Margin	Confi-dence	Non-Separable	Adaptive Margin
PA	Yes	No	Yes	No
SOP	No	Yes	Yes	No
IELLIP	No	Yes	Yes	No
CW	Yes	Yes	No	Yes
AROW	Yes	Yes	Yes	No
NHERD	Yes	Yes	Yes	No
NAROW	Yes	Yes	Yes	No
SCW	Yes	Yes	Yes	Yes

Second-order Methods

Soft confidence weighted learning *(ICML 2012)*

Algorithm 1 SCW learning algorithms (**SCW**)

INPUT: parameters $C > 0$, $\eta > 0$.
INITIALIZATION: $\mu_0 = (0, \ldots, 0)^\top$, $\Sigma_0 = I$.
for $t = 1, \ldots, T$ **do**
  Receive an example $\mathbf{x}_t \in \mathbb{R}^d$;
  Make prediction: $\hat{y}_t = sgn(\boldsymbol{\mu}_{t-1} \cdot \mathbf{x}_t)$;
  Receive true label y_t;
  suffer loss $\ell^\phi(\mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}); (\mathbf{x}_t, y_t))$;
  if $\ell^\phi(\mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}); (\mathbf{x}_t, y_t)) > 0$ **then**
    $\boxed{\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t y_t \Sigma_t \mathbf{x}_t, \Sigma_{t+1} = \Sigma_t - \beta_t \Sigma_t \mathbf{x}_t{}^T \mathbf{x}_t \Sigma_t}$
    where α_t and β_t are computed by either Proposition 1 (SCW-I) or Proposition 2 (SCW-II);
  end if
end for

Like PA-I and PA-II

- ✓ SCW-I limits the biggest step size
- ✓ SCW-II performs feature dimension extension

Second-order Methods

Pros and Cons

☺ Learn both **first order** and **second order** info
☺ Faster **convergence rate**

☹ Relatively **sensitive to noise**
☹ **Expensive** for high-dimensional data (w and Σ)

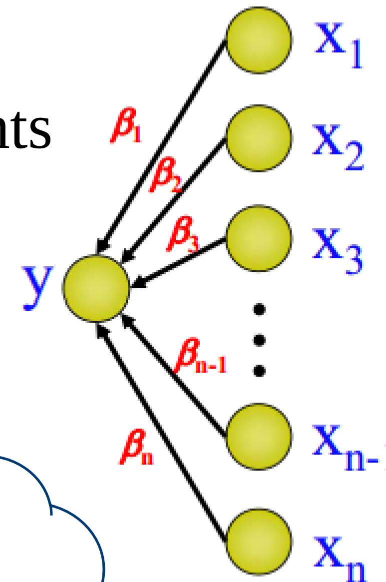
Says we have a large matrix Σ (and/or vector w), any troubles??

– Slow prediction
– Storage issue
– ……

Sparse Online Methods

Motivations

- Sparsity for high-dimensional data
- Faster online prediction
- Test computational cost / test-time constraints
- Space constraints
- ……



Methods

- Truncated gradient
- FOBOS
- RDA
- FTRL
- ……

Sparse Online Methods

Sparsity

- **Three options for sparsity**
  - **Simple Coefficient Rounding**
    - w_i is small because ?

Aggressive

  - **L1 norm**
    - Gradient update has the form $a + b$ where a and b are two floats

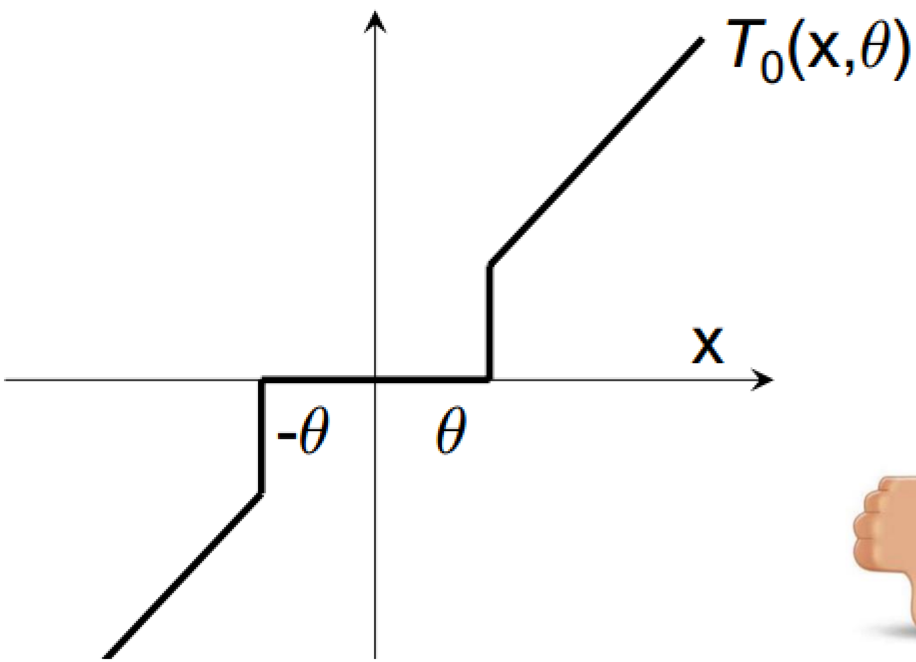
  - **Black-box wrapper feature selection**
    - Run an algorithm many times which is particularly undesirable with large data sets

Sparse Online Methods

Truncated gradient *(JMLR 2009)*

- **Simple Coefficient Rounding**
  - If $t/K == 1$ do

$$f(w_i) = T_0(w_i - \eta \nabla_1 L(w_i, z_i), \theta)$$



$$T_0(v_j, \theta) = \begin{cases} 0 & \text{if } |v_j| \le \theta \\ v_j & \text{otherwise} \end{cases}$$

Sensitive K without theoretical guarantee

Sparse Online Methods

Truncated gradient *(JMLR 2009)*

– **L1 norm**

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} L(w, z_i) + g\|w\|_1$$

$$f(w_i) = w_i - \eta \nabla_1 L(w_i, z_i) - \eta g \,\mathrm{sgn}(w_i)$$

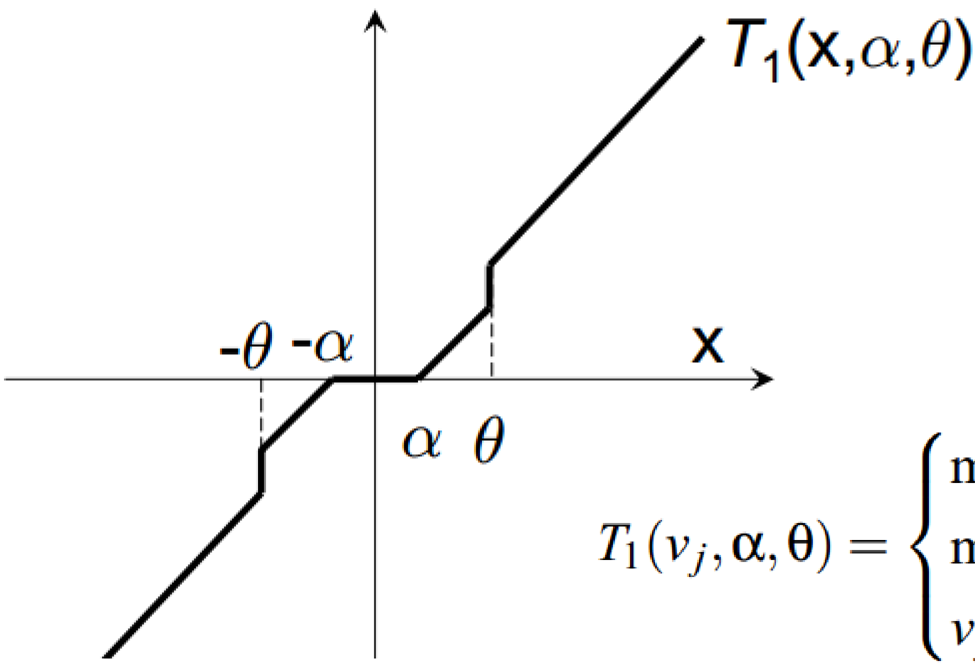
Sparse Online Methods

Truncated gradient *(JMLR 2009)*

- Combine simple rounding and L1 norm method
- Perform TG at K^{th} time with *gravity* parameter $g_i > 0$

$$f(w_i) = T_1(w_i - \eta \nabla_1 L(w_i, z_i), \eta g_i, \theta) \qquad g_i = Kg$$

Sparsity!

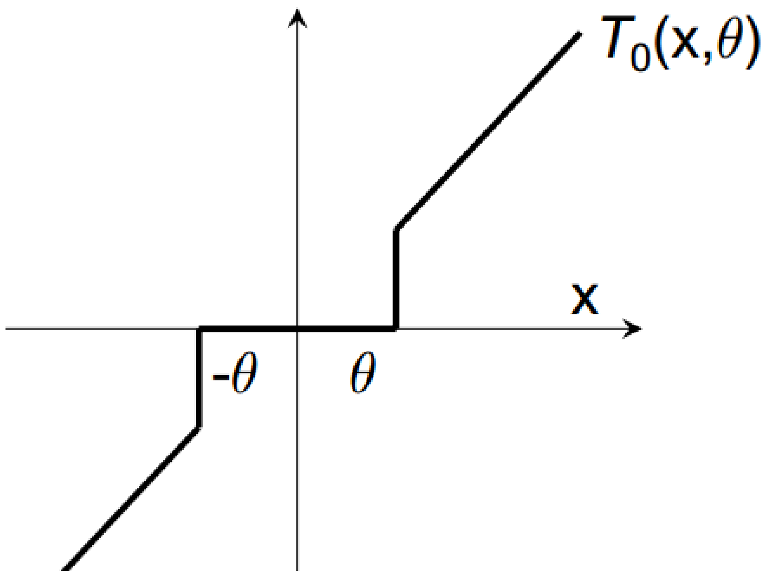


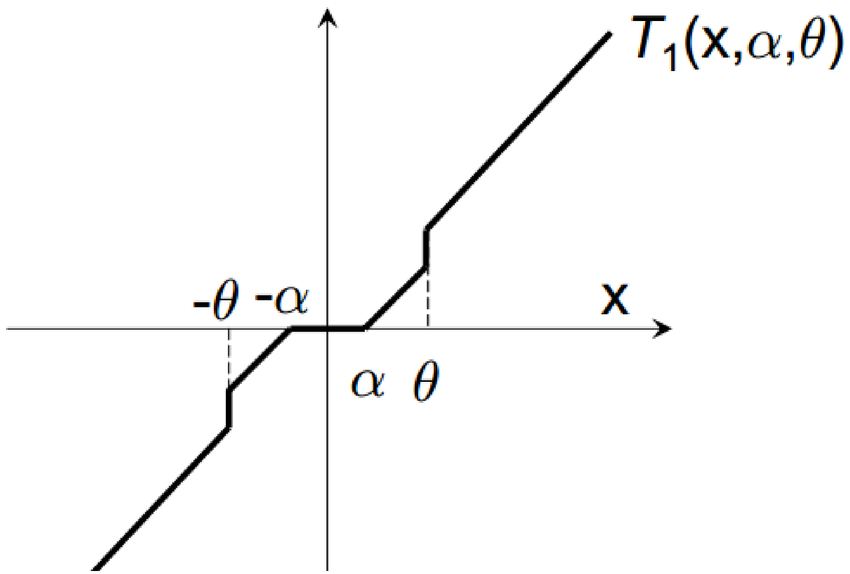
$$T_1(v_j, \alpha, \theta) = \begin{cases} \max(0, v_j - \alpha) & \text{if } v_j \in [0, \theta] \\ \min(0, v_j + \alpha) & \text{if } v_j \in [-\theta, 0] \\ v_j & \text{otherwise} \end{cases}$$

Sparse Online Methods

TG & simple rounding

If $\alpha \geq \theta$, TG = simple rounding



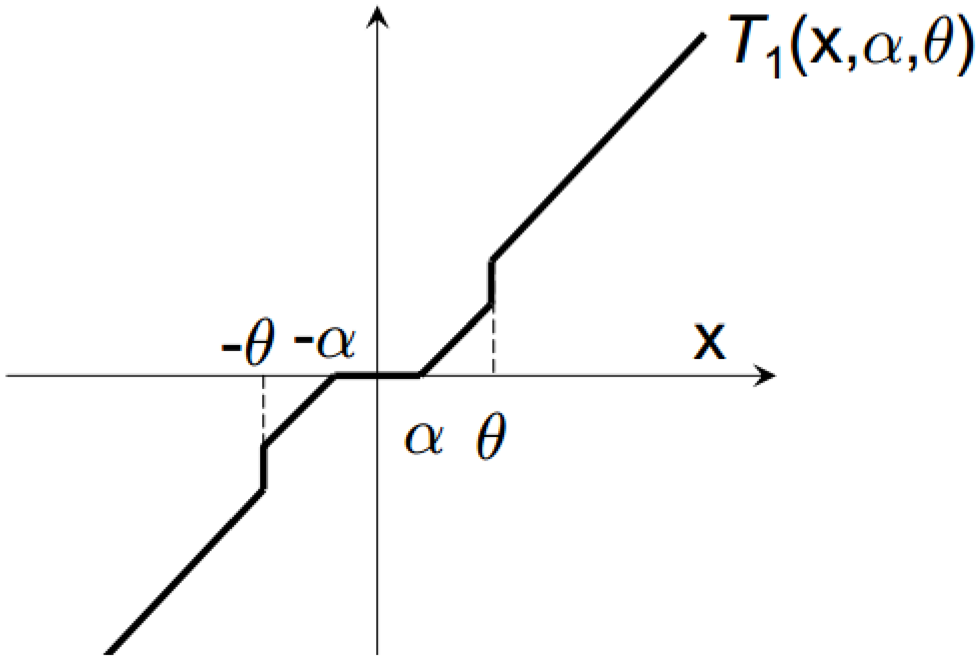


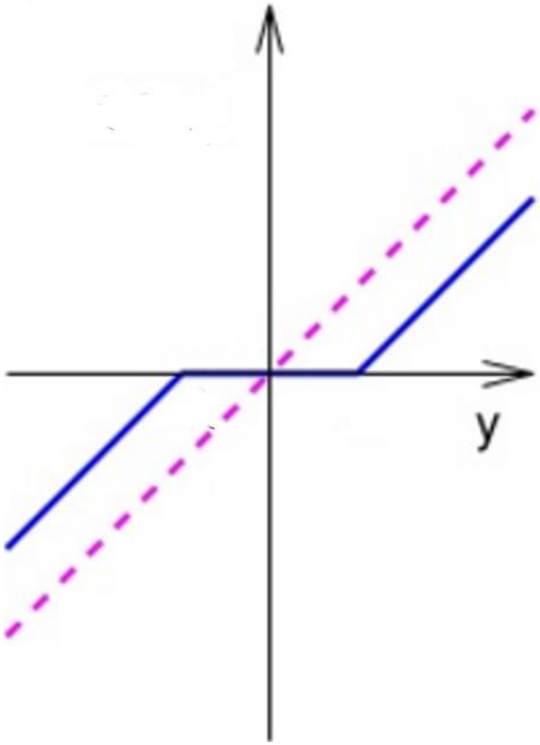
Sparse Online Methods

TG & L1-norm

– If $\theta = \infty$ and $K = 1$

TG = L1 Norm

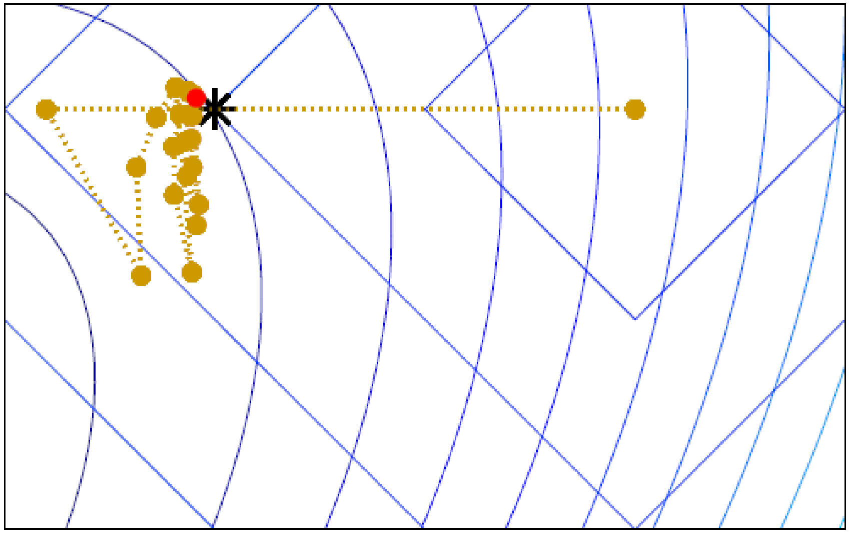




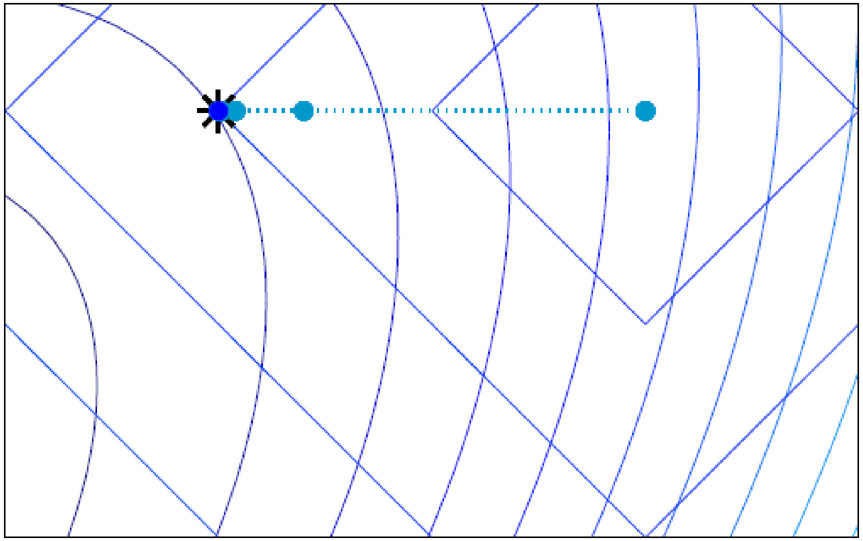
Sparse Online Methods
Forward-Backward Splitting

FOBOS *(JMLR 2009)*

Minimize $\frac{1}{2}\boldsymbol{w}^\top A\boldsymbol{w} + \boldsymbol{c}^\top\boldsymbol{w} + \lambda\|\boldsymbol{w}\|_1$. True solution: $\boldsymbol{w}^* = [-1\ 0]^\top$.



Subgradient



Fobos

Sparse Online Methods
Forward-Backward Splitting

FOBOS *(JMLR 2009)*

- **Objectives**

$$\underbrace{f(x)}_{loss} + \underbrace{\Psi(x)}_{regularization}$$

- **Motivation**
  - have the iterates w_t attain points of non-differentiability of the function Ψ
- **Two-step update**

$$W^{(t+\frac{1}{2})} = W^{(t)} - \eta^{(t)} G^{(t)}$$

$$W^{(t+1)} = \underset{W}{argmin} \left\{ \frac{1}{2} \left\| W - W^{(t+\frac{1}{2})} \right\|^2 + \eta^{(t+\frac{1}{2})} \Psi(W) \right\}$$

Sparse Online Methods
Forward-Backward Splitting

FOBOS *(JMLR 2009)*

– **Objectives**

$$W^{(t+\frac{1}{2})} = W^{(t)} - \eta^{(t)} G^{(t)}$$

$$W^{(t+1)} = \underset{W}{argmin} \left\{ \frac{1}{2} \left\| W - W^{(t+\frac{1}{2})} \right\|^2 + \eta^{(t+\frac{1}{2})} \Psi(W) \right\}$$

$$\Downarrow$$

$$W^{(t+1)} = \underset{W}{argmin} \left\{ \frac{1}{2} \left\| W - W^{(t)} + \eta^{(t)} G^{(t)} \right\|^2 + \eta^{(t+\frac{1}{2})} \Psi(W) \right\}$$

$$\Downarrow$$

$$0 \in \partial F(W) = W - W^{(t)} + \eta^{(t)} G^{(t)} + \eta^{(t+\frac{1}{2})} \partial \Psi(W)$$

$$W^{(t+1)} = W^{(t)} - \eta^{(t)} G^{(t)} - \eta^{(t+\frac{1}{2})} \partial \Psi(W^{(t+1)})$$

Sparse Online Methods
Forward-Backward Splitting

FOBOS-L1 *(JMLR 2009)*

Algorithm 4. Forward-Backward Splitting with L1 Regularization

input λ

initial $W \in \mathbb{R}^N$

for $t = 1,2,3\ldots$ *do*

$G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$

refresh W *according to*

$$\boxed{w_i = sgn\left(w_i - \eta^{(t)} g_i\right) max\left\{0, \left|w_i - \eta^{(t)} g_i\right| - \eta^{\left(t+\frac{1}{2}\right)} \lambda\right\}}$$

end

return W

New version of TG

Sparse Online Methods
Regularized Dual Averaging *@Microsoft*

RDA *(JMLR 2010)*

- **Objectives**

$$\underbrace{f(x)}_{loss} + \underbrace{\Psi(x)}_{regularization}$$

- **Update**

$$W^{(t+1)} = \underset{W}{argmin}\left\{\boxed{\frac{1}{t}\sum_{r=1}^{t}\langle G^{(r)}, W\rangle} + \Psi(W) + \boxed{\frac{\beta^{(t)}}{t}h(W)}\right\}$$

Averaging gradient

Additional strong convex function

$\{\boldsymbol{\beta}^{(t)}\}_{t\geq 1}$: non-negative & non-decreasing input sequence

Sparse Online Methods
Regularized Dual Averaging

RDA *(JMLR 2010)*

- **Steps**
  - compute a subgradient

$$\mathbf{g}_t = \nabla_{\mathbf{w}} \ell(y_t, \mathbf{w}_t^\top \mathbf{x}_t)$$

  - Update average subgradient

$$\bar{\mathbf{g}}_t = \frac{t-1}{t}\bar{\mathbf{g}}_{t-1} + \frac{1}{t}\mathbf{g}_t$$

  - Compute the next weight vector

$$\langle \bar{\mathbf{g}}_t, \mathbf{w} \rangle + \lambda \|\mathbf{w}\|_1 + \frac{\beta_t}{2t}\|\mathbf{w}\|_2^2$$

Sparse Online Methods
Regularized Dual Averaging

RDA-L1

– **Objectives**

$$W^{(t+1)} = \underset{W}{argmin}\left\{\frac{1}{t}\sum_{r=1}^{t}\langle G^{(r)}, W\rangle + \lambda\|W\|_1 + \frac{\gamma}{2\sqrt{t}}\|W\|_2^2\right\} \quad \beta^{(t)} = \gamma\sqrt{t}$$

Algorithm 5. Regularized Dual Averaging with L1 Regularization

input γ, λ

initialize $W \in \mathbb{R}^N, G = 0 \in \mathbb{R}^N$

for $t = 1,2,3\ldots$ *do*

$$G = \frac{t-1}{t}G + \frac{1}{t}\nabla_W \ell(W, X^{(t)}, y^{(t)})$$

refresh W *according to*

$$\boxed{w_i^{(t+1)} = \begin{cases} 0 & \text{if } |g_i| < \lambda \\ -\frac{\sqrt{t}}{\gamma}(g_i - \lambda sgn(g_i)) & otherwise \end{cases}}$$

end

return W

Sparse Online Methods
Regularized Dual Averaging

RDA-L1 V.S. FOBOS-L1

$$w_i^{(t+1)} = \begin{cases} 0 & if \ \left| w_i^{(t)} - \eta^{(t)} g_i^{(t)} \right| < \boxed{\eta^{\left(t+\frac{1}{2}\right)} \lambda} \\ \left(w_i^{(t)} - \eta^{(t)} g_i^{(t)} \right) - \eta^{\left(t+\frac{1}{2}\right)} \lambda sign\left(w_i^{(t)} - \eta^{(t)} g_i^{(t)} \right) & otherwise \end{cases}$$

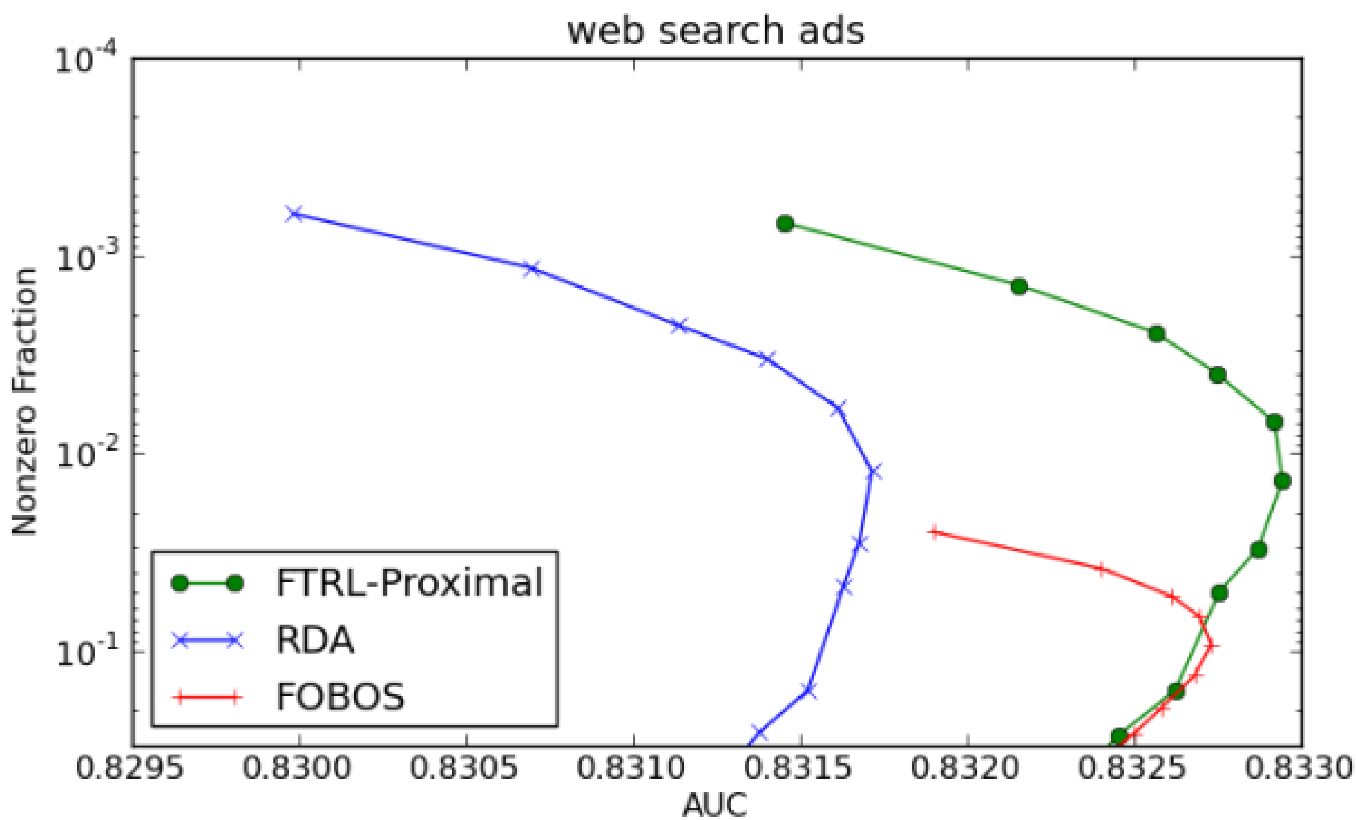
$$w_i^{(t+1)} = \begin{cases} 0 & if \ |\bar{g}_i| < \boxed{\lambda} \\ -\frac{\sqrt{t}}{r} \left(\boxed{\bar{g}_i} - \lambda sign(\bar{g}_i) \right) & otherwise \end{cases}$$

RDA Use the cumulative mean of gradients and it's more aggressive to obtain sparsity

Sparse Online Methods
Follow The Regularized Leader *@Google*

FTRL *(AISTATS, 2011)*

– Combine FOBOS (accuracy) and RDA (sparsity)



Sparse Online Methods
Follow The Regularized Leader

FTRL *(COLT'10, AISTATS'11, KDD'13)*

– Combine FOBOS (stabilization constraint) and RDA (regularization)

$$\begin{array}{lllllll}
 & & & (A) & & (B) & (C) \\
\text{FOBOS} & x_{t+1} & = \arg\min_x & g_{1:t} \cdot x & + & \phi_{1:t-1} \cdot x + \Psi(x) & + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}} (x - x_s)\|_2^2 \\
\text{AOGD} & x_{t+1} & = \arg\min_x & g_{1:t} \cdot x & + & \phi_{1:t-1} \cdot x + \Psi(x) & + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}} (x - 0)\|_2^2 \\
\text{RDA} & x_{t+1} & = \arg\min_x & g_{1:t} \cdot x & + & t\Psi(x) & + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}} (x - 0)\|_2^2 \\
\text{FTRL-Proximal} & x_{t+1} & = \arg\min_x & g_{1:t} \cdot x & + & t\Psi(x) & + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}} (x - x_s)\|_2^2
\end{array}$$

Gradient part of loss function

Sub-gradient part of regularization

Additional stabilization

Sparse Online Methods
Follow The Regularized Leader

FTRL with L1 & L2 norm

– **Objectives**

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(1:t)} \cdot W + \lambda_1 \|W\|_1 + \lambda_2 \frac{1}{2}\|W\|_2^2 + \frac{1}{2}\sum_{s=1}^{t} \sigma^{(s)} \left\|W - W^{(s)}\right\|_2^2\right\}$$

$$\frac{1}{learning\ rate}$$

$$w_i^{(t+1)} = \begin{cases} 0 & \text{if } \left|z_i^{(t)}\right| < \lambda_1 \\ -\left(\lambda_2 + \boxed{\sum_{s=1}^{t} \sigma^{(s)}}\right)^{-1} \left(z_i^{(t)} - \lambda_1 sgn(z_i^{(t)})\right) & otherwise \end{cases}$$

$$Z^{(t)} = G^{(1:t)} - \sum_{s=1}^{t} \sigma^{(s)} W^{(s)}$$

$$\sigma^{(s)} = \frac{1}{\eta^{(s)}} - \frac{1}{\eta^{(s-1)}}, \quad \sigma^{(1:t)} = \frac{1}{\eta^{(t)}}$$

Sparse Online Methods
Follow The Regularized Leader

FTRL with L1 & L2 norm

Algorithm 1 Per-Coordinate FTRL-Proximal with L_1 and L_2 Regularization for Logistic Regression

 # With per-coordinate learning rates of Eq. (2).
Input: parameters α, β, λ_1, λ_2
$(\forall i \in \{1, \dots, d\})$, initialize $z_i = 0$ and $n_i = 0$
for $t = 1$ **to** T **do**
  Receive feature vector $\mathbf{x}_t$ and let $I = \{i \mid x_i \neq 0\}$
  For $i \in I$ compute

$$w_{t,i} = \begin{cases} 0 & \text{if } |z_i| \leq \lambda_1 \\ -\left(\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda_2\right)^{-1} (z_i - \text{sgn}(z_i)\lambda_1) & \text{otherwise.} \end{cases}$$

  Predict $p_t = \sigma(\mathbf{x}_t \cdot \mathbf{w})$ using the $w_{t,i}$ computed above
  Observe label $y_t \in \{0, 1\}$
  for all $i \in I$ **do**
    $g_i = (p_t - y_t)x_i$ *#gradient of loss w.r.t.* w_i
    $\sigma_i = \frac{1}{\alpha}\left(\sqrt{n_i + g_i^2} - \sqrt{n_i}\right)$ *#equals* $\frac{1}{\eta_{t,i}} - \frac{1}{\eta_{t-1,i}}$
    $z_i \leftarrow z_i + g_i - \sigma_i w_{t,i}$
    $n_i \leftarrow n_i + g_i^2$
  end for
end for

Per-coordinate learning rate $\eta_{t,i}$

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

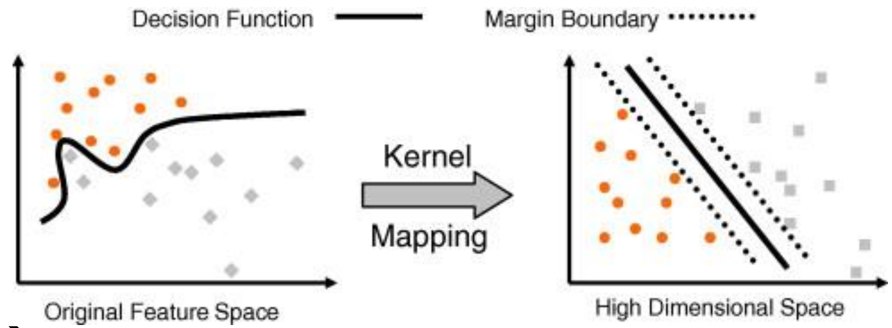
Says feature i varies a lot, (large gradient), then it should have a large $\eta_{t,i}$

Further Topics
*Non-linear Online Learning

Online kernel learning

– **Objectives**

$$f_t(\cdot) = \sum_{i=1}^{B} \alpha_i^t y_i^t \kappa(\mathbf{x}_i^t, \cdot)$$



– **Challenges**

– Unbounded support vectors → **Budget B**

– **Methods**

– SV removal *(NIPS'03, NIPS'05, Machine Learning'07)*

– SV projection *(ICML'08)*

– SV merging *(ICDM'09)*

– Kernel approximation *(JMLR'16)*

$$f(\mathbf{x}) = \sum_{i=1}^{B} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \approx \sum_{i=1}^{B} \alpha_i \mathbf{z}(\mathbf{x}_i)^\top \mathbf{z}(\mathbf{x}) = \mathbf{w}^\top \mathbf{z}(\mathbf{x})$$

Further Topics
*Non-linear Online Learning

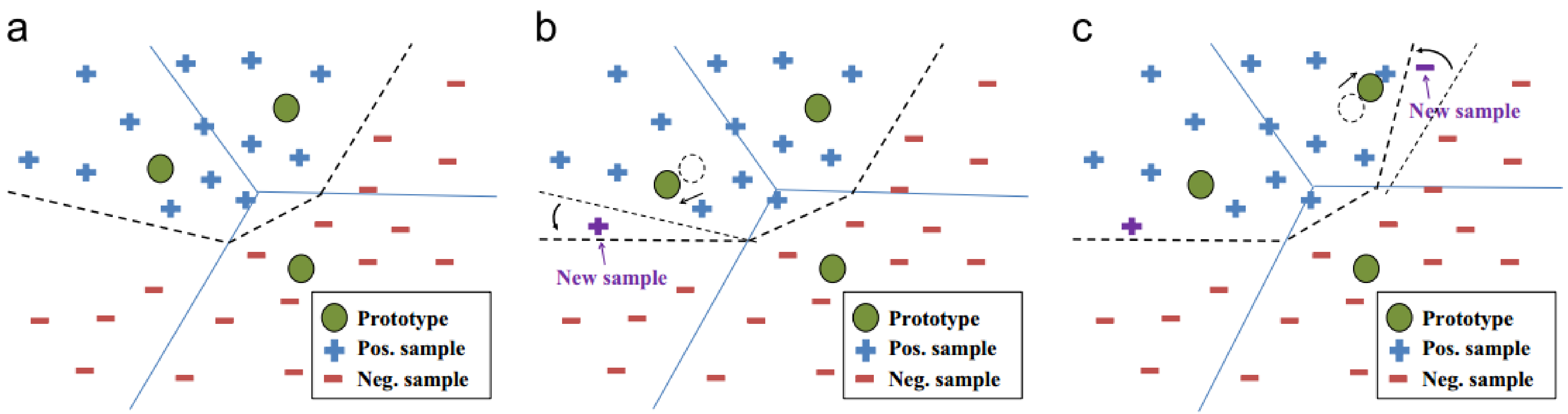
Local online learning *(Pattern Recognition'15)*

– **Idea**

– Although data is not always globally linearly separable, it's still possible that they are **locally linearly separable**

– Jointly **learning multiple local hyper-planes**

$$\mathbf{w}_i = \mathbf{w} + \mathbf{u}_i$$



Further Topics
*Multi-class Online Learning

Online multi-class learning

– **Objectives**

– Computes a **similarity score** between each prototype and the input instance

– **Methods**

– Learn a function f^r for each of the classes $r \in Y$

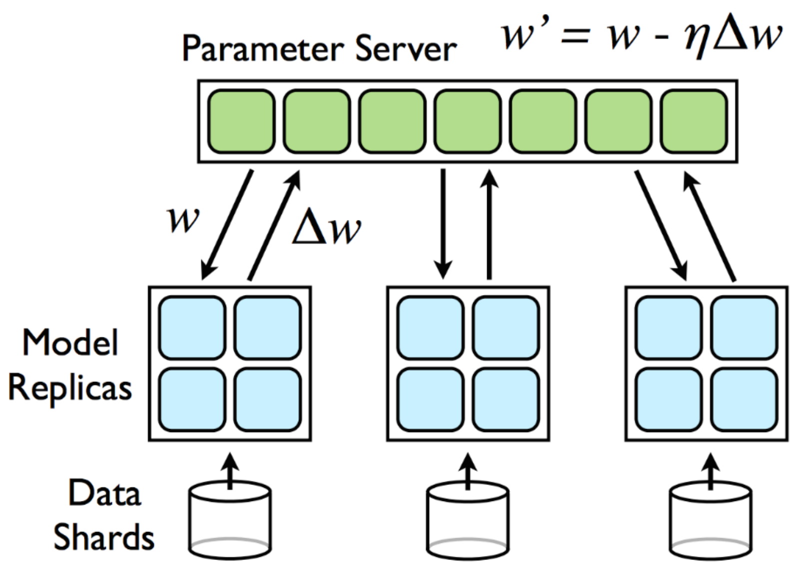
– Similarity-based margin loss

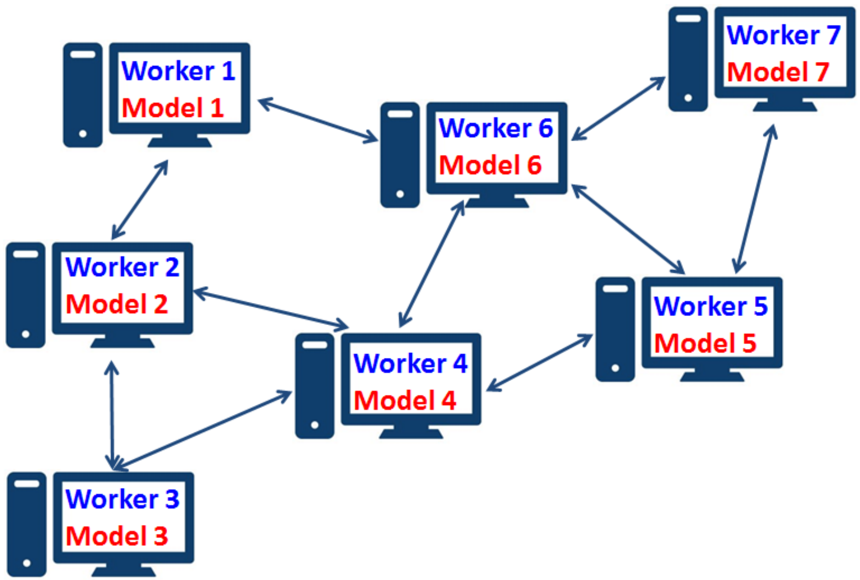
$$l\left(\{f^i\}_{1:k}, x_t, y_t\right) = \max(0, 1 - r_t)$$
$$r_t = \underset{r \in Y}{arg\max} f_t^r(x_t) - \underset{r \in Y, r \neq y_t}{arg\max} f_t^r(x_t)$$

Further Topics

More issues to deal with

– **Centralized/Decentralized Distributed Online Learning**





Applications

Online learning applications

- Online AUC Maximization (*AAAI'15*)
- Cost-Sensitive online learning (*ICDM'12, ICDM'15*)
- Online collaborative filtering (*ICDM'05*)
- Online metric/similarity learning (*ICDM'15, ICML'12*)
- Online multi-task learning (*JMLR'14*)
- Online manifold learning (*PKDD'08*)
- Online semi-supervised learning (*AAAI'11*)
- Online time series prediction (*JMLR'13*)
- Online NMF (*CIKM'16*)

Take Home Messages

Online learning

- **What is online learning**
- **Regret analysis ?**
- **Update rule**
  - When to update
  - How to update

- **Several famous methods**
  - First-order (PA, PA-I, PA-II)
  - Second order (CW, SCW, AROW)
  - Sparsity (RDA, FTRL)

- **Focus on the online algorithms of your field or interests**

Thanks

By HC