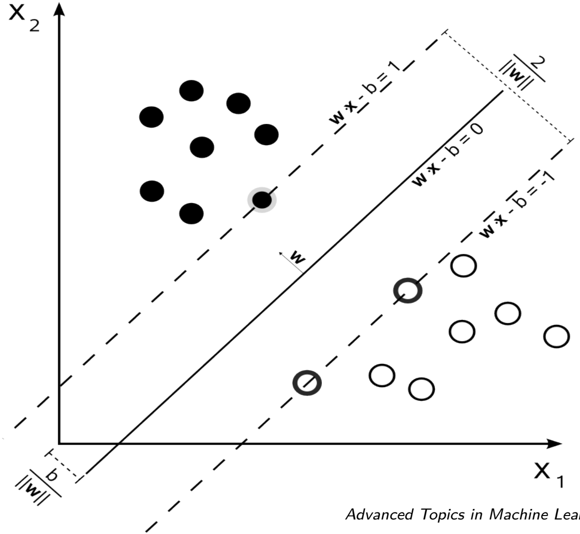
Primal estimated sub-gradient solver for SVM

Lei Zhong

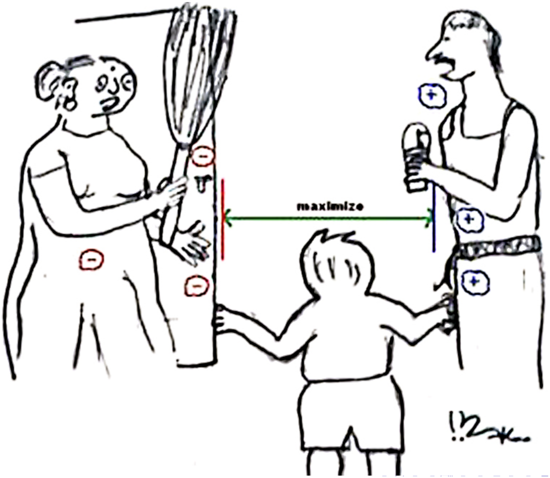
Advanced Topics in Machine Learning

Nov. 4, 2014

Motivating example



Family Support Machine



Support Vector Machine: Primal Problem

Data:

$$\{(\boldsymbol{x}_i, y_i) \in R^d \times \{+1, -1\} : i \in S \overset{def}{=} \{1, 2, \dots, n\}\}$$

- Example: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ (assumption: $\max_i \|\boldsymbol{x}_i\|_2 \leq 1$)
- Labels: $y_i \in \{+1, -1\}$

Optimization formulation of SVM:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \{f(\boldsymbol{w}) := \hat{L}_S(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2\}$$

where

- $\hat{L}_A(\boldsymbol{w}) \overset{def}{=} \frac{1}{|A|}\sum_{i \in A} L_i$ (average loss on examples in A)

Loss Function and Subgradient

Definition

- Loss: $L_i := \ell(\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle, y_i)$
- Subgradient: $l'(\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle, y_i)$

Use the notation $z = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle$, sample loss functions:

Loss function	Subgradient
$l(z, y_i) = \max\{0, 1 - y_i z\}$	ss
$l(z, y_i) = \log(1 + e^{-y_i z})$	1
$l(z, y_i) = \max\{0, \|y_i - z\| - \epsilon\}$	1

Prevous Work

- Dual-based methods
  - Interior Point
    - Memory: m^2, time: $m^3 log(log(1/\epsilon))$
  - Decomposition
    - Memory: m, time: super-linear in m
- Online learning & Stochastic Gradient
  - Memory: $O(1)$, time: $1/\epsilon^2$ (linear kernel)

Better rates for finite dimensional instances (Murata, Bottou)

Typically, online learning algorithms do not converge to the optimal solution of SVM

Basic Pegasos Algorithm (SGD)

1. Choose $\boldsymbol{w}_1 = 0 \in \mathbb{R}^d$
2. Iterate for $t = 1, 2, \ldots, T$
   1. Choose $A_t \subset S = \{1, 2, \ldots, n\}, |A_t| = b$, uniformly at random
   2. Set stepsize $\eta_t \leftarrow \frac{1}{\lambda t}$
   3. Update $w^{(t+1)} \leftarrow w^{(t)} - \eta_t \partial f_{A_t}(\boldsymbol{w}^{(t)})$

Theorem

For $\overline{\boldsymbol{w}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{w}_t$, we have:

$$\mathbb{E}[f(\overline{\boldsymbol{w}})] \leq f(w^*) + c \log(T) \times \frac{1}{\lambda T}$$

where $c = (\sqrt{(}\lambda) + 1)^2$.

Basic Pegasos Algorithm (SGD)

1. Choose $\boldsymbol{w}_1 = 0 \in \mathbb{R}^d$
2. Iterate for $t = 1, 2, \ldots, T$
   1. Choose $A_t \subset S = \{1, 2, \ldots, n\}, |A_t| = b$, uniformly at random
   2. Set stepsize $\eta_t \leftarrow \frac{1}{\lambda t}$
   3. Update $w^{(t+1)} \leftarrow (1 - \eta\lambda) w^{(t)} + \frac{\eta_t}{b} \sum_{i \in A_t} l' \boldsymbol{x}_i$

Theorem

For $\overline{\boldsymbol{w}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{w}_t$, we have:

$$\mathbb{E}[f(\overline{\boldsymbol{w}})] \leq f(w^*) + c \log(T) \times \frac{1}{\lambda T}$$

where $c = (\sqrt{(}\lambda) + 1)^2$.

Run-Time of Pegasos

- Choosing $|A_t| = 1$
  - $\rightarrow$ Run-time required for Pegasos to find ϵ accurate solution w.p. , $1 - \delta$

$$\tilde{O}(\frac{n}{\delta\lambda\epsilon})$$

- Run-time does not depends on #examples
- Depends on "difficulty" of problem (λ and ϵ)

Formal Properties

- Definition: $\boldsymbol{w}$ is ϵ accurate if $f(\boldsymbol{w}) - f(\boldsymbol{w}^*) \leq \epsilon$
- Theorem 1: Pegasos finds ϵ accurate solution w.p., $1 - \delta$ after at most

  $$\tilde{O}(\frac{1}{\delta \lambda \epsilon})$$

  iterations.
- Theorem 2: Pegasos finds $\log(1/\delta)$ solutions $\mathrm{s.t.}$ w.p., at least one of them is ϵ accurate after

  $$\tilde{O}(\frac{\log(1/\delta)}{\lambda \epsilon})$$

  iterations.

Proof Sketch

- Logarithmic Regret for OCP (Hazan et al 06)

Experiments

- 3 datasets (provided by Joachims)
  - Reuters CCAT (800K examples, 47k features)
  - Physics ArXiv (62k examples, 100k features)
  - Covertype (581k examples, 54 features)
- 4 competing algorithms
  - SVM-light (Joachims)
  - SVM-Perf (Joachims'06)
  - Norma (Kivinen, Smola, Williamson '02)
  - Zhang'04 (stochastic gradient descent)

Training Time(in seconds)

	Pegasos	SVM-Perf	SVM-Light
Reuters	**2**	77	20,075
Covertype	**6**	85	25,514
Astro-Physics	**2**	5	80

Compare to Norma (on Physics)

Compare to Zhang (on Physics)

Effect of $k = |A_t|$ when T is fixed

Effect of $k = |A_t|$ when kT is fixed

I want my kernels!

- Pegasos can seamlessly be adapted to employ non-linear kernels while working solely on the primal objective function
- No need to switch to the dual problem
- Number of support vectors is bounded by

$$\tilde{O}(\frac{1}{\lambda\epsilon})$$

Complex Decision Problems

- Pegasos works whenever we know how to calculate subgradients of loss func. $l(\boldsymbol{w};(\boldsymbol{x},y))$
- Example: Structured output prediction

$$l(\boldsymbol{w};(\boldsymbol{x},y)) = \max_{y'}[\gamma(y,y') - \langle \boldsymbol{w}, \phi(\boldsymbol{x},y) - \phi(\boldsymbol{x},y')\rangle]_+$$

- Subgradient is $\phi(\boldsymbol{x},y') - \phi(\boldsymbol{x},y)$ where y' is the maximizer in the definition of l

Bias term

- Popular approach: increase dimension of x
  Cons: "pay" for b in the regularization term
- Calculate subgradients w.r.t w and w.r.t b:
  Cons: convergence rate is $1/\epsilon^2$
- Define: $L(\boldsymbol{w}) = \min_b \sum_{(\boldsymbol{x},y)\in S}[1 - y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - b)]_+$
- Search b in an outer loop
  Cons: evaluating objective is $1/\epsilon^2$

Discussion

- Pegasos: Simple & Efficient solver for SVM
- Sample vs. computational complexity
  - Sample complexity: How many examples do we need as a function of VC-dim(λ), accuracy(ϵ), and confidence(δ)
  - in Pegasos, we aim at analyzing computational complexity based on λ, ϵ, δ (also in Bottou & Bousquet)
- Finding argmin vs. calculating min: It seems that Pegasos finds the argmin more easily than it requires to calculate the min value

Thank You!

Q&A

Acknowledgement:

Thanks to Martin for helpful discussions, suggestions and chips!!!

Reference

[1] Shalev-Shwartz, Shai, et al. "Pegasos: Primal estimated sub-gradient solver for svm." Mathematical programming 127.1 (2011): 3-30.

[2] Lacoste-Julien, Simon, Mark Schmidt, and Francis Bach. "A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method." arXiv preprint arXiv:1212.2002 (2012).

[3] Takáč Martin, et al. "Mini-batch primal and dual methods for SVMs." arXiv preprint arXiv:1303.2314 (2013).

[4] Zhao, Peilin, and Tong Zhang. "Stochastic optimization with importance sampling." arXiv preprint arXiv:1401.2753 (2014).