

Optimization for Machine Learning

Lecture 2: Support Vector Machine Training

S.V. N. (vishy) Vishwanathan

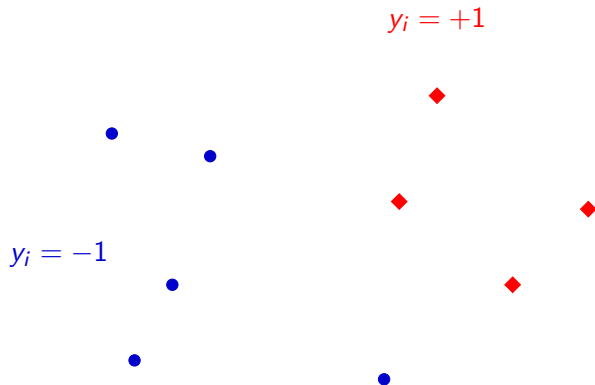
Purdue University
vishy@purdue.edu

July 11, 2012

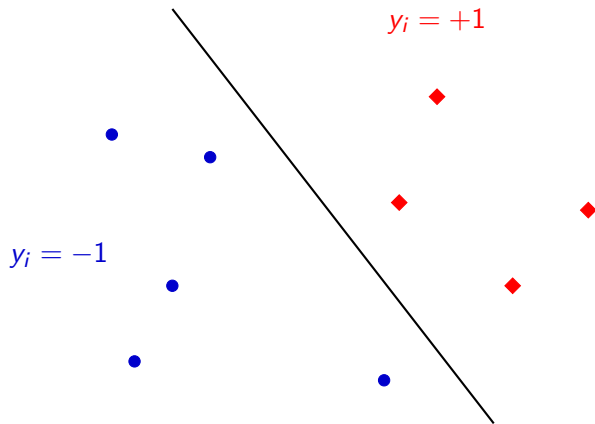
Outline

- 1 Linear Support Vector Machines**
- 2 Stochastic Optimization
- 3 Implicit Updates
- 4 Dual Problem

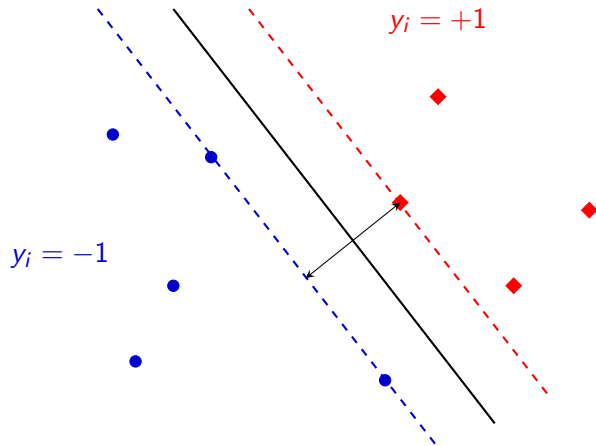
Binary Classification



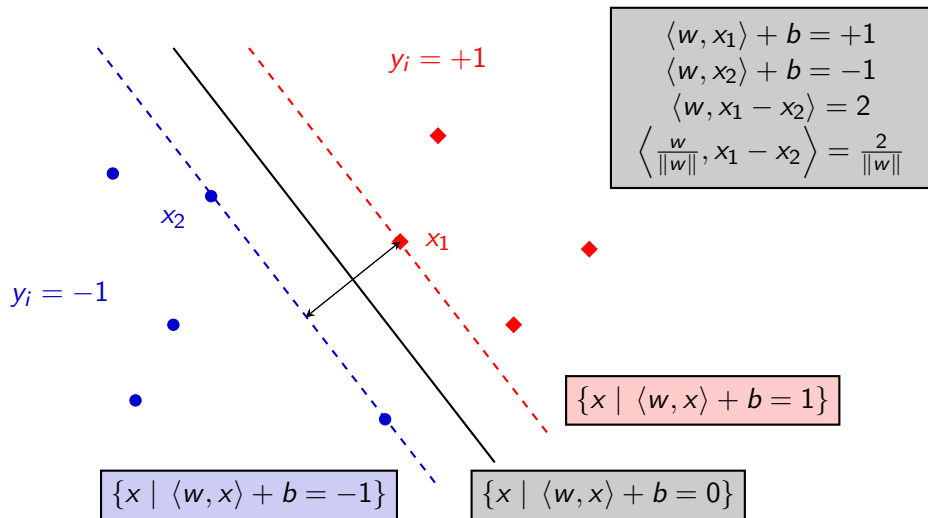
Binary Classification



Binary Classification



Binary Classification



Linear Support Vector Machines

Optimization Problem

$$\min_{w, b, \xi} \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i$$

$$\text{s.t. } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \text{ for all } i$$
$$\xi_i \geq 0$$

Linear Support Vector Machines

Optimization Problem

$$\min_{w,b} \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i(\langle w, x_i \rangle + b))$$

Linear Support Vector Machines

Optimization Problem

$$\min_{w,b} \underbrace{\frac{\lambda}{2} \|w\|^2}_{\lambda\Omega(w)} + \underbrace{\frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i(\langle w, x_i \rangle + b))}_{R_{\text{emp}}(w)}$$

Outline

- 1 Linear Support Vector Machines
- 2 Stochastic Optimization**
- 3 Implicit Updates
- 4 Dual Problem

Stochastic Optimization Algorithms

Optimization Problem (with no bias)

$$\min_w \underbrace{\frac{\lambda}{2} \|w\|^2}_{\Omega(w)} + \underbrace{\frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i \langle w, x_i \rangle)}_{R_{\text{emp}}(w)}$$

- Unconstrained
- Nonsmooth
- Convex

Pegasos: Stochastic Gradient Descent

Require: T

- 1: $w_0 \leftarrow 0$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\eta_t \leftarrow \frac{1}{\lambda t}$
- 4: **if** $y_t \langle w_t, x_t \rangle < 1$ **then**
- 5: $w'_t \leftarrow (1 - \eta_t \lambda) w_t + \eta_t y_t x_t$
- 6: **else**
- 7: $w'_t \leftarrow (1 - \eta_t \lambda) w_t$
- 8: **end if**
- 9: **end for**
- 10: $w_{t+1} \leftarrow \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|w'_t\|} \right\} w'_t$

Understanding Pegasos

Objective Function Revisited

$$J(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i \langle w, x_i \rangle)$$

Subgradient

- If $y_t \langle w, x_t \rangle < 1$ then

$$\partial_w J_t(w) = \lambda w - y_t x_t$$

- else

$$\partial_w J_t(w) = \lambda w$$

Understanding Pegasos

Objective Function Revisited

$$J(w) \approx J_t(w) = \frac{\lambda}{2} \|w\|^2 + \max(0, 1 - y_t \langle w, x_t \rangle)$$

Subgradient

- If $y_t \langle w, x_t \rangle < 1$ then

$$\partial_w J_t(w) = \lambda w - y_t x_t$$

- else

$$\partial_w J_t(w) = \lambda w$$

Understanding Pegasos

Objective Function Revisited

$$J(w) \approx J_t(w) = \frac{\lambda}{2} \|w\|^2 + \max(0, 1 - y_t \langle w, x_t \rangle)$$

Subgradient

- If $y_t \langle w, x_t \rangle < 1$ then

$$\partial_w J_t(w) = \lambda w - y_t x_t$$

- else

$$\partial_w J_t(w) = \lambda w$$

Understanding Pegasos

Explicit Update

- If $y_t \langle w, x_t \rangle < 1$ then

$$w'_t = w_t - \eta_t \partial_w J_t(w_t) = (1 - \lambda \eta_t) w_t + y_t x_t$$

- else

$$w'_t = w_t - \eta_t \partial_w J_t(w_t) = (1 - \lambda \eta_t) w_t$$

Projection

Project w'_t onto the set

$$B = \left\{ w \text{ s.t. } \|w\| \leq 1/\sqrt{\lambda} \right\}$$

Motivating Stochastic Gradient Descent

How are the Updates Derived?

- Minimize the following objective function

$$w_{t+1} = \operatorname{argmin}_w \frac{1}{2} \|w - w_t\|^2 + \eta_t J_t(w)$$

- This gives us

Motivating Stochastic Gradient Descent

How are the Updates Derived?

- Minimize the following objective function

$$w_{t+1} = \operatorname{argmin}_w \frac{1}{2} \|w - w_t\|^2 + \eta_t J_t(w)$$

- This gives us

$$w_{t+1} = w_t - \eta_t \partial_w J_t(w_{t+1})$$

Motivating Stochastic Gradient Descent

How are the Updates Derived?

- Minimize the following objective function

$$w_{t+1} = \operatorname{argmin}_w \frac{1}{2} \|w - w_t\|^2 + \eta_t J_t(w)$$

- This gives us

$$w_{t+1} = w_t - \eta_t \partial_w J_t(w_{t+1})$$

Motivating Stochastic Gradient Descent

How are the Updates Derived?

- Minimize the following objective function

$$w_{t+1} = \operatorname{argmin}_w \frac{1}{2} \|w - w_t\|^2 + \eta_t J_t(w)$$

- This gives us

$$w_{t+1} \approx w_t - \eta_t \partial_w J_t(w_t)$$

Outline

- 1 Linear Support Vector Machines
- 2 Stochastic Optimization
- 3 Implicit Updates**
- 4 Dual Problem

Implicit Updates

What if we did not approximate $\partial_w J_t(w_{t+1})$?

$$w_{t+1} = w_t - \eta_t \partial_w J_t(w_{t+1})$$

Subgradient

$$\partial_w J_t(w) = \lambda w - \gamma y_t x_t$$

- If $y_t \langle w, x_t \rangle < 1$ then $\gamma = 1$
- If $y_t \langle w, x_t \rangle = 1$ then $\gamma \in [0, 1]$
- If $y_t \langle w, x_t \rangle > 1$ then $\gamma = 0$

Implicit Updates

What if we did not approximate $\partial_w J_t(w_{t+1})$?

$$w_{t+1} = w_t - \eta_t \partial_w J_t(w_{t+1})$$

Subgradient

$$\partial_w J_t(w) = \lambda w - \gamma y_t x_t$$

- If $y_t \langle w, x_t \rangle < 1$ then $\gamma = 1$
- If $y_t \langle w, x_t \rangle = 1$ then $\gamma \in [0, 1]$
- If $y_t \langle w, x_t \rangle > 1$ then $\gamma = 0$

Implicit Updates

What if we did not approximate $\partial_w J_t(w_{t+1})$?

$$w_{t+1} = w_t - \eta_t \lambda w_{t+1} + \gamma \eta_t y_t x_t$$

Subgradient

$$\partial_w J_t(w) = \lambda w - \gamma y_t x_t$$

- If $y_t \langle w, x_t \rangle < 1$ then $\gamma = 1$
- If $y_t \langle w, x_t \rangle = 1$ then $\gamma \in [0, 1]$
- If $y_t \langle w, x_t \rangle > 1$ then $\gamma = 0$

Implicit Updates

What if we did not approximate $\partial_w J_t(w_{t+1})$?

$$(1 + \eta_t \lambda) w_{t+1} = w_t + \gamma \eta_t y_t x_t$$

Subgradient

$$\partial_w J_t(w) = \lambda w - \gamma y_t x_t$$

- If $y_t \langle w, x_t \rangle < 1$ then $\gamma = 1$
- If $y_t \langle w, x_t \rangle = 1$ then $\gamma \in [0, 1]$
- If $y_t \langle w, x_t \rangle > 1$ then $\gamma = 0$

Implicit Updates

What if we did not approximate $\partial_w J_t(w_{t+1})$?

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

Subgradient

$$\partial_w J_t(w) = \lambda w - \gamma y_t x_t$$

- If $y_t \langle w, x_t \rangle < 1$ then $\gamma = 1$
- If $y_t \langle w, x_t \rangle = 1$ then $\gamma \in [0, 1]$
- If $y_t \langle w, x_t \rangle > 1$ then $\gamma = 0$

Implicit Updates: Case 1

The Implicit Update Condition

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

Case 1

- Suppose $1 + \eta_t \lambda < y_t \langle w_t, x_t \rangle$. Set

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} w_t$$

- Verify $y_t \langle w_{t+1}, x_t \rangle > 1$ which implies that $\gamma = 0$ and the implicit update condition is satisfied

Implicit Updates: Case 2

The Implicit Update Condition

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

Case 2

- Suppose $y_t \langle w_t, x_t \rangle < 1 + \eta_t \lambda - \eta_t \langle x_t, x_t \rangle$. Set

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \eta_t y_t x_t]$$

- Verify $y_t \langle w_{t+1}, x_t \rangle < 1$ which implies that $\gamma = 1$ and the implicit update condition is satisfied

Implicit Updates: Case 3

The Implicit Update Condition

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

Case 3

- Suppose $1 + \eta_t \lambda - \eta_t \langle x_t, x_t \rangle \leq y_t \langle w_t, x_t \rangle \leq 1 + \eta_t \lambda$. Set

$$\gamma = \frac{1 + \eta_t \lambda - y_t \langle w_t, x_t \rangle}{\eta_t \langle x_t, x_t \rangle}$$

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

- Verify $\gamma \in [0, 1]$ and $y_t \langle w_{t+1}, x_t \rangle = 1$

Implicit Updates: Summary

Summary

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

- If $1 + \eta_t \lambda < y_t \langle w_t, x_t \rangle$ then $\gamma = 0$
- If $1 + \eta_t \lambda - \eta_t \langle x_t, x_t \rangle \leq y_t \langle w_t, x_t \rangle \leq 1 + \eta_t \lambda$ then

$$\gamma = \frac{1 + \eta_t \lambda - y_t \langle w_t, x_t \rangle}{\eta_t \langle x_t, x_t \rangle}$$

- If $y_t \langle w_t, x_t \rangle < 1 + \eta_t \lambda - \eta_t \langle x_t, x_t \rangle$ then $\gamma = 1$

Implicit Updates: Summary

Summary

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

$$\gamma = \min \left(1, \max \left(0, \frac{1 + \eta_t \lambda - y_t \langle w_t, x_t \rangle}{\eta_t \langle x_t, x_t \rangle} \right) \right)$$

Outline

- 1 Linear Support Vector Machines
- 2 Stochastic Optimization
- 3 Implicit Updates
- 4 Dual Problem**

Deriving the Dual

Lagrangian

- Recall the primal problem without bias

$$\begin{aligned} \min_{w, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i \langle w, x_i \rangle \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \end{aligned}$$

- Introduce non-negative dual variables α and β

$$\begin{aligned} L(w, \xi, \alpha, \beta) = & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i - \sum_i \alpha_i (y_i \langle w, x_i \rangle - 1 + \xi_i) \\ & - \sum_i \beta_i \xi_i \end{aligned}$$

Deriving the Dual

Lagrangian

- Recall the primal problem without bias

$$\begin{aligned} \min_{w, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i \langle w, x_i \rangle \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \end{aligned}$$

- Introduce non-negative dual variables α and β

$$\begin{aligned} L(w, \xi, \alpha, \beta) = & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i - \sum_i \alpha_i (y_i \langle w, x_i \rangle - 1 + \xi_i) \\ & - \sum_i \beta_i \xi_i \end{aligned}$$

Deriving the Dual

Take Gradients and Set to Zero

- Write the gradients

$$\nabla_w L(w, \xi, \alpha, \beta) = \lambda w - \sum_i \alpha_i y_i x_i = 0$$

$$\nabla_{\xi_i} L(w, \xi, \alpha, \beta) = \frac{1}{m} - \beta_i - \alpha_i = 0$$

- Conclude that

$$w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$$

$$0 \leq \alpha_i \leq \frac{1}{m}$$

Deriving the Dual

Plug back into Lagrangian

- Plug $w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$ and $\beta_i + \alpha_i = \frac{1}{m}$ into the Lagrangian

$$\begin{aligned} \max_{\alpha} \quad & -D(\alpha) := -\frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{m} \end{aligned}$$

Deriving the Dual

Plug back into Lagrangian

- Plug $w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$ and $\beta_i + \alpha_i = \frac{1}{m}$ into the Lagrangian

$$\begin{aligned} \min_{\alpha} D(\alpha) &:= \frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_i \alpha_i \\ \text{s.t.} \quad &0 \leq \alpha_i \leq \frac{1}{m} \end{aligned}$$

Deriving the Dual

Plug back into Lagrangian

- Plug $w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$ and $\beta_i + \alpha_i = \frac{1}{m}$ into the Lagrangian

$$\min_{\alpha} D(\alpha) := \frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_i \alpha_i$$

$$\text{s.t. } 0 \leq \alpha_i \leq \frac{1}{m}$$

- Quadratic objective
- Linear (box) constraints

Coordinate Descent in the Dual

One dimensional function

$$\hat{D}(\alpha_t) = \frac{\alpha_t^2}{2\lambda} \langle x_t, x_t \rangle + \frac{1}{\lambda} \sum_i \alpha_t \alpha_i y_i y_t \langle x_i, x_t \rangle - \alpha_t + \text{const.}$$

$$\text{s.t. } 0 \leq \alpha_t \leq \frac{1}{m}$$

Take Gradients and set to Zero

$$\nabla \hat{D}(\alpha_t) = \frac{\alpha_t}{\lambda} \langle x_t, x_t \rangle + \frac{1}{\lambda} \sum_i \alpha_i y_i y_t \langle x_i, x_t \rangle - 1 = 0$$

Coordinate Descent in the Dual

One dimensional function

$$\hat{D}(\alpha_t) = \frac{\alpha_t^2}{2\lambda} \langle x_t, x_t \rangle + \frac{1}{\lambda} \sum_i \alpha_t \alpha_i y_i y_t \langle x_i, x_t \rangle - \alpha_t + \text{const.}$$

$$\text{s.t. } 0 \leq \alpha_t \leq \frac{1}{m}$$

Take Gradients and set to Zero

$$\nabla \hat{D}(\alpha_t) = \frac{\alpha_t}{\lambda} \langle x_t, x_t \rangle + \frac{1}{\lambda} y_t \left\langle \underbrace{w_t}_{:= \sum_i y_i \alpha_i x_i}, x_t \right\rangle - 1 = 0$$

Coordinate Descent in the Dual

One dimensional function

$$\hat{D}(\alpha_t) = \frac{\alpha_t^2}{2\lambda} \langle x_t, x_t \rangle + \frac{1}{\lambda} \sum_i \alpha_t \alpha_i y_i y_t \langle x_i, x_t \rangle - \alpha_t + \text{const.}$$

$$\text{s.t. } 0 \leq \alpha_t \leq \frac{1}{m}$$

Take Gradients and set to Zero

$$\alpha_t = \min \left(\max \left(0, \frac{\lambda - y_t \langle w_t, x_t \rangle}{\langle x_t, x_t \rangle} \right), \frac{1}{m} \right)$$

Contrast with Implicit Updates

Coordinate Descent in the Dual

$$w_t = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$$

$$\alpha_t = \min \left(\frac{1}{m}, \max \left(0, \frac{\lambda - y_t \langle w_t, x_t \rangle}{\langle x_t, x_t \rangle} \right) \right)$$

Implicit Updates

$$w_{t+1} = \frac{1}{1 + \eta_t \lambda} [w_t + \gamma \eta_t y_t x_t]$$

$$\gamma = \min \left(1, \max \left(0, \frac{1 + \eta_t \lambda - y_t \langle w_t, x_t \rangle}{\eta_t \langle x_t, x_t \rangle} \right) \right)$$

Outline

- 1 Linear Support Vector Machines
- 2 Stochastic Optimization
- 3 Implicit Updates
- 4 Dual Problem

What if Data Does not Fit in Memory?

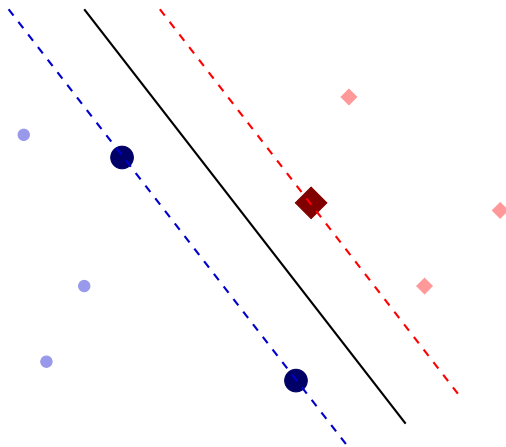
Idea 1: Block Minimization [Yu et al., KDD 2010]

- Split data into blocks $B_1, B_2 \dots$ such that B_j fits in memory
- Compress and store each block separately
- Load one block of data at a time and optimize only those α_j 's

Idea 2: Selective Block Minimization [Chang and Roth, KDD 2011]

- Split data into blocks $B_1, B_2 \dots$ such that B_j fits in memory
- Compress and store each block separately
- Load one block of data at a time and optimize only those α_j 's
- Retain *informative samples* from each block in main memory

What are Informative Samples?



Some Observations

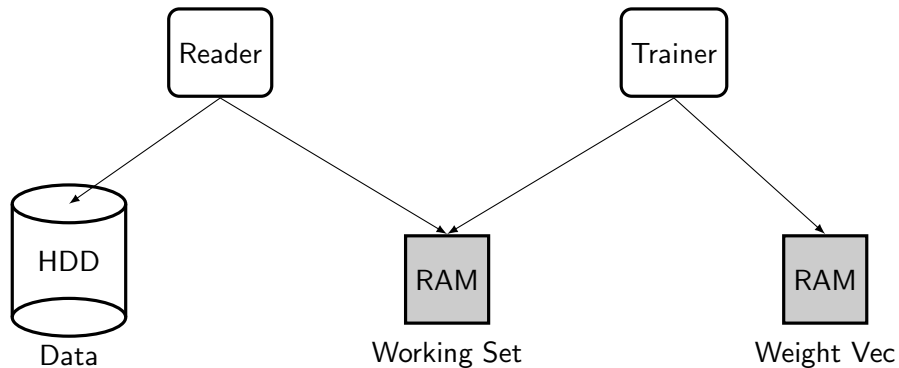
SBM and BM are wasteful

- Both split data into blocks and compress the blocks
 - This requires reading the entire data at least once (expensive)
- Both pause optimization while a block is loaded into memory

Hardware 101

- Disk I/O is slower than CPU (sometimes by a factor of 100)
- Random access on HDD is terrible
 - sequential access is reasonably fast (factor of 10)
- Multi-core processors are becoming commonplace
- How can we exploit this?

Dual Cached Loops [Matsushima, Vishwanathan, Smola]



Underlying Philosophy

Iterate over the data in main memory while streaming data from disk. Evict primarily examples from main memory that are “uninformative”.

Reader

```

for  $k = 1, \dots, \text{max\_iter}$  do
  for  $i = 1, \dots, n$  do
    if  $|A| = \Omega$  then
      randomly select  $i' \in A$ 
       $A = A \setminus \{i'\}$ 
      delete  $y_i, Q_{ii}, x_i$  from RAM
    end if
    read  $y_i, x_i$  from Disk
    calculate  $Q_{ii} = \langle x_i, x_i \rangle$ 
    store  $y_i, Q_{ii}, x_i$  in RAM
     $A = A \cup \{i\}$ 
  end for
  if stopping criterion is met then
    exit
  end if
end for

```

Trainer

$\alpha^1 = \mathbf{0}, w^1 = \mathbf{0}, \varepsilon = 9, \varepsilon^{\text{new}} = 0, \beta = 0.9$

while stopping criterion is not met **do**

for $t = 1, \dots, n$ **do**

If $|A| > 0.9 \times \Omega$ **then** $\varepsilon = \beta\varepsilon$

randomly **select** $i \in A$ and read y_i, Q_{ii}, x_i from RAM

compute $\nabla_i D := y_i \langle w^t, x_i \rangle - 1$

if $(\alpha_i^t = 0 \text{ and } \nabla_i D > \varepsilon)$ or $(\alpha_i^t = C \text{ and } \nabla_i D < -\varepsilon)$ **then**

$A = A \setminus \{i\}$ and delete y_i, Q_{ii}, x_i from RAM

continue

end if

$\alpha_i^{t+1} = \text{median} \left(0, C, \alpha_i^t - \frac{\nabla_i D}{Q_{ii}} \right), w^{t+1} = w^t + (\alpha_i^{t+1} - \alpha_i^t) y_i x_i$

$\varepsilon^{\text{new}} = \max(\varepsilon^{\text{new}}, |\nabla_i D|)$

end for

Update stopping criterion

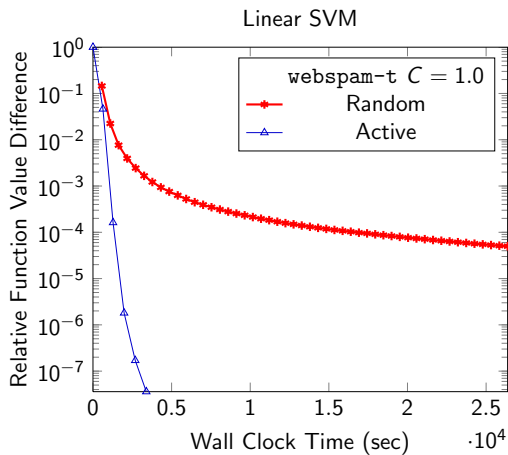
$\varepsilon = \varepsilon^{\text{new}}$

end while

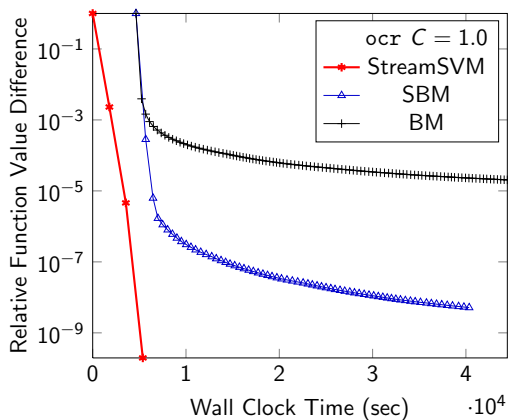
Experiments

dataset	n	d	$s(\%)$	$n_+ : n_-$	Datasize
ocr	3.5 M	1156	100	0.96	45.28 GB
dna	50 M	800	25	$3e-3$	63.04 GB
webspam-t	0.35 M	16.61 M	0.022	1.54	20.03 GB
kddb	20.01 M	29.89 M	$1e-4$	6.18	4.75 GB

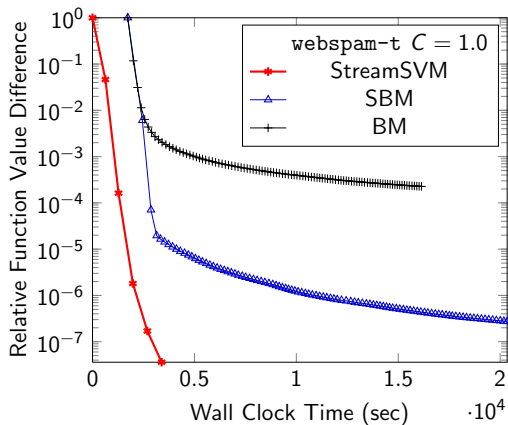
Does Active Eviction Work?



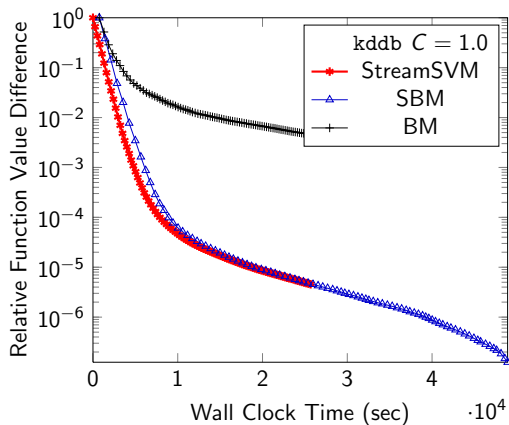
Comparison with Block Minimization



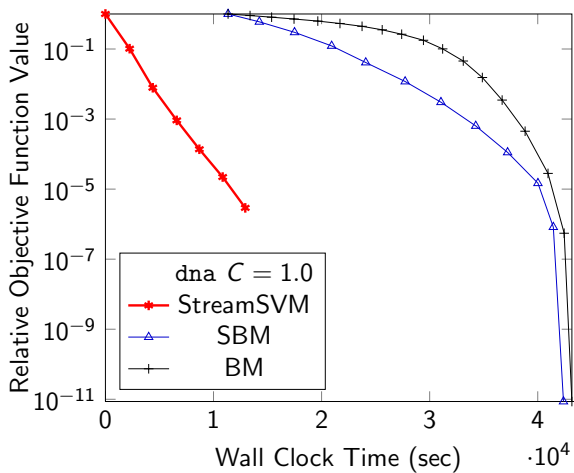
Comparison with Block Minimization



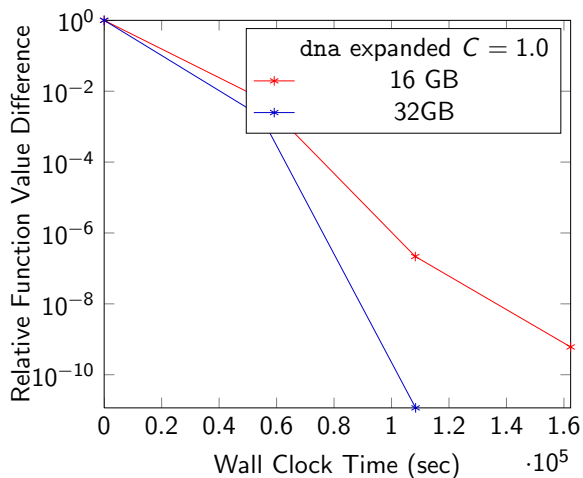
Comparison with Block Minimization



Comparison with Block Minimization



Expanding Features



Outline

- 1 Linear Support Vector Machines
- 2 Stochastic Optimization
- 3 Implicit Updates
- 4 Dual Problem

Let us Bring Back the Bias

Lagrangian

- Recall the primal problem

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \end{aligned}$$

- Introduce non-negative dual variables α and β

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) = & \frac{\lambda}{2} \|w\|^2 - \sum_i \beta_i \xi_i \\ & + \frac{1}{m} \sum_{i=1}^m \xi_i - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - 1 + \xi_i) \end{aligned}$$

Let us Bring Back the Bias

Lagrangian

- Recall the primal problem

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \end{aligned}$$

- Introduce non-negative dual variables α and β

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) = & \frac{\lambda}{2} \|w\|^2 - \sum_i \beta_i \xi_i \\ & + \frac{1}{m} \sum_{i=1}^m \xi_i - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - 1 + \xi_i) \end{aligned}$$

Let us Bring Back the Bias

Take Gradients and Set to Zero

- Write the gradients

$$\nabla_w L(w, b, \xi, \alpha, \beta) = \lambda w - \sum_i \alpha_i y_i x_i = 0$$

$$\nabla_b L(w, b, \xi, \alpha, \beta) = \sum_i \alpha_i y_i = 0$$

$$\nabla_{\xi_j} L(w, b, \xi, \alpha) = \frac{1}{m} - \beta_j - \alpha_j = 0$$

- Conclude that

$$w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$$

$$\sum_i \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq \frac{1}{m}$$

Let us Bring Back the Bias

Plug back into Lagrangian

- Plug $w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$ and $\beta_i + \alpha_i = \frac{1}{m}$ into the Lagrangian

$$\begin{aligned} \max_{\alpha} \quad & -D(\alpha) := -\frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq \frac{1}{m} \end{aligned}$$

Let us Bring Back the Bias

Plug back into Lagrangian

- Plug $w = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i$ and $\beta_i + \alpha_i = \frac{1}{m}$ into the Lagrangian

$$\min_{\alpha} D(\alpha) := \frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq \frac{1}{m}$$

Coordinate Descent in the Dual

One Dimensional Function

- Cannot pick one coordinate so pick two!
- Call the two coordinates t_1 and t_2

$$\begin{aligned} \hat{D}(\eta_{t_1}, \eta_{t_2}) &= \frac{\eta_{t_1}^2}{2\lambda} \langle x_{t_1}, x_{t_1} \rangle + \frac{\eta_{t_2}^2}{2\lambda} \langle x_{t_2}, x_{t_2} \rangle \\ &\quad + \frac{\eta_{t_1}}{\lambda} \sum_i \alpha_i \langle x_i, x_{t_1} \rangle + \frac{\eta_{t_2}}{\lambda} \sum_i \alpha_i \langle x_i, x_{t_2} \rangle \\ &\quad + \frac{\eta_{t_1} \eta_{t_2}}{\lambda} \langle x_{t_1}, x_{t_2} \rangle - \eta_{t_1} - \eta_{t_2} + \text{const.} \\ \text{s.t. } &y_{t_1} \eta_{t_1} + y_{t_2} \eta_{t_2} = 0 \\ &0 \leq \alpha_{t_1} + \eta_{t_1} \leq \frac{1}{m} \\ &0 \leq \alpha_{t_2} + \eta_{t_2} \leq \frac{1}{m} \end{aligned}$$

Coordinate Descent in the Dual

One Dimensional Function

- Cannot pick one coordinate so pick two!
- Call the two coordinates t_1 and t_2

$$\eta_{t_1} = -\frac{y_{t_2}}{y_{t_1}}\eta_{t_2} = \eta$$

Coordinate Descent in the Dual

One Dimensional Function

- Cannot pick one coordinate so pick two!
- Call the two coordinates t_1 and t_2

$$\begin{aligned}
 \hat{D}(\eta_{t_1}, \eta_{t_2}) &= \frac{\eta^2}{2\lambda} \langle x_{t_1}, x_{t_1} \rangle + \frac{\eta^2}{2\lambda} \langle x_{t_2}, x_{t_2} \rangle \\
 &\quad + \frac{\eta}{\lambda} \sum_i \alpha_i \langle x_i, x_{t_1} \rangle - \frac{\eta y_{t_1}}{\lambda y_{t_2}} \sum_i \alpha_i \langle x_i, x_{t_2} \rangle \\
 &\quad - \frac{\eta^2 y_{t_1}}{\lambda y_{t_2}} \langle x_{t_1}, x_{t_2} \rangle - \eta + \frac{y_{t_1}}{y_{t_2}} \eta + \text{const.} \\
 \text{s.t. } &0 \leq \alpha_{t_1} + \eta \leq \frac{1}{m} \\
 &0 \leq \alpha_{t_2} - \frac{y_{t_1}}{y_{t_2}} \eta \leq \frac{1}{m}
 \end{aligned}$$

Software

- LibSVM: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- LibLinear: <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

References (Incomplete)

- Implicit Updates
 - Kivinen and Warmuth, *Exponentiated Gradient Versus Gradient Descent for Linear Predictors*, Information and Computation, 1997.
 - Kivinen, Warmuth, and Hassibi, *The p -norm generalization of the LMS algorithm for adaptive filtering*, IEEE Transactions on Signal Processing, 2006.
 - Cheng, Vishwanathan, Schuurmans, Wang, and Caelli, *Implicit Online Learning With Kernels*, NIPS 2006.
 - Hsieh, Chang, Lin, Keerthi, and Sundararajan, *A Dual Coordinate Descent Method for Large-scale Linear SVM*, ICML 2008.
- SMO
 - Platt, *Fast Training of Support Vector Machines using Sequential Minimal Optimization*, Advances in Kernel Methods—Support Vector Learnin, 1999.
- Dual Cached Loops
 - Matsushima, Vishwanathan, and Smola, *Linear Support Vector Machines via Dual Cached Loops*, KDD 2012.

References (Incomplete)

- Slides are loosely based on lecture notes from
 - <http://learning.stat.purdue.edu/wiki/courses/sp2011/598a/lectures>
 - <http://www.ee.ucla.edu/~vandenbe/shortcourses.html>