Support Vector Machines

CS4780 – Machine Learning
Fall 2009

Thorsten Joachims
Cornell University

Reading: Schoelkopf/Smola Chapter 7.3, 7.5
Cristianini/Shawe-Taylor Chapter 2-2.1.1

(Batch) Perceptron Algorithm

Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$, $\vec{x}_i \in \Re^N$, $y_i \in \{-1, 1\}$, $\eta \in \Re$, $I \in [1, 2, ..]$

Algorithm:

- $\vec{w}_0 = \vec{0}$, $k = 0$
- repeat
  - FOR i=1 TO n
    - IF $y_i(\vec{w}_k \cdot \vec{x}_i) \leq 0$ ### makes mistake
      - $\vec{w}_{k+1} = \vec{w}_k + \eta y_i \vec{x}_i$
      - $k = k + 1$
    - ENDIF
  - ENDFOR
- until I iterations reached

Dual (Batch) Perceptron Algorithm

Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$, $\vec{x}_i \in \Re^N$, $y_i \in \{-1, 1\}$, $I \in [1, 2, ..]$

Dual Algorithm:

- $\forall i \in [1..n] : \alpha_i = 0$
- repeat
  - FOR i=1 TO n
    - IF $y_i\left(\sum_{j=1}^n \alpha_j y_j (\vec{x}_j \cdot \vec{x}_i)\right) \leq 0$
      - $\alpha_i = \alpha_i + 1$
    - ENDIF
  - ENDFOR
- until I iterations reached

Primal Algorithm:

- $\vec{w} = \vec{0}$, $k = 0$
- repeat
  - FOR i=1 TO n
    - IF $y_i(\vec{w} \cdot \vec{x}_i) \leq 0$
      - $\vec{w} = \vec{w} + y_i \vec{x}_i$
    - ENDIF
  - ENDFOR
- until I iterations reached

SVM Solution as Linear Combination

- **Primal OP:** minimize: $P(\vec{w}, b, \vec{\xi}) = \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum_{i=1}^n \xi_i$
  subject to: $\forall_{i=1}^n : y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i$
  $\forall_{i=1}^n : \xi_i > 0$
- **Theorem:** The solution w^* can always be written as a linear combination
  $$\vec{w}^* = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \text{ with } 0 \leq \alpha_i \leq C$$
  of the training vectors.
- **Properties:**
  - Factor α_i indicates "influence" of training example (x_i, y_i).
  - If $\xi_i > 0$, then $\alpha_i = C$.
  - If $0 \leq \alpha_i < C$, then $\xi_i = 0$.
  - (x_i, y_i) is a Support Vector, if and only if $\alpha_i > 0$.
  - If $0 < \alpha_i < C$, then $y_i(x_i\, w + b) = 1$.
  - SVM-light outputs α_i using the "-a" option

Dual SVM Optimization Problem

- **Primal Optimization Problem**
  minimize: $P(\vec{w}, b, \vec{\xi}) = \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum_{i=1}^n \xi_i$
  subject to: $\forall_{i=1}^n : y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i$
  $\forall_{i=1}^n : \xi_i > 0$
- **Dual Optimization Problem**
  maximize: $D(\vec{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j (\vec{x}_i \cdot \vec{x}_j)$
  subject to: $\sum_{i=1}^n y_i \alpha_i = 0$
  $\forall_{i=1}^n : 0 \leq \alpha_i \leq C$
- **Theorem:** If w^* is the solution of the Primal and α^* is the solution of the Dual, then $\vec{w}^* = \sum_{i=1}^n \alpha_i^* y_i \vec{x}_i$

Leave-One-Out (i.e. n-fold CV)

Training Set: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$

Approach: Repeatedly leave one example out for testing.

train on	test on
$(\vec{x}_2, y_2), (\vec{x}_3, y_3), (\vec{x}_4, y_4), ..., (\vec{x}_n, y_n)$	$(\vec{x}_1, y_1)$
$(\vec{x}_1, y_1), (\vec{x}_3, y_3), (\vec{x}_4, y_4), ..., (\vec{x}_n, y_n)$	$(\vec{x}_2, y_2)$
$(\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_4, y_4), ..., (\vec{x}_n, y_n)$	$(\vec{x}_3, y_3)$
...	...
$(\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3), ..., (\vec{x}_{n-1}, y_{n-1})$	$(\vec{x}_n, y_n)$

Estimate: $Err_{loo}(A) = \frac{1}{n} \sum_{i=1}^n \Delta(h_i(\vec{x}_i), y_i)$

Question: Is there a cheaper way to compute this estimate?

Necessary Condition for Leave-One-Out Error

Lemma: For SVM, $[h_i(\vec{x}_i) \neq y_i] \Longrightarrow [2\alpha_i R^2 + \xi_i \geq 1]$

Input:

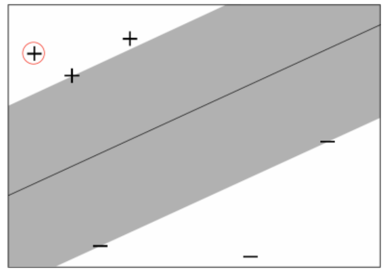
- α_i dual variable of example i
- ξ_i slack variable of example i
- $||x|| \leq R$ bound on length

Example:

Value of $2\,\alpha_i\,R^2 + \xi_i$	Leave-one-out Error?
0.0	Correct
0.7	Correct
3.5	Error
0.1	Correct
1.3	Correct
…	…

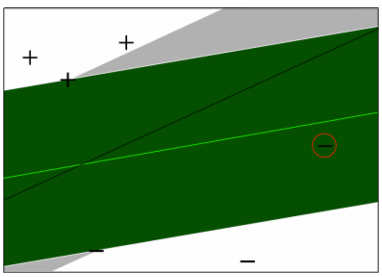
Case 1: Example is not SV

Criterion: $(\alpha_i = 0) \Rightarrow (\xi_i=0) \Rightarrow (2\,\alpha_i\,R^2 + \xi_i < 1) \Rightarrow$ Correct



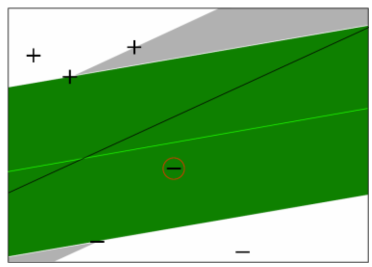
Case 2: Example is SV with Low Influence

Criterion: $(\alpha_i<0.5/R^2 < C) \Rightarrow (\xi_i=0) \Rightarrow (2\alpha_i R^2+\xi_i < 1) \Rightarrow$ Correct



Case 3: Example has Small Training Error

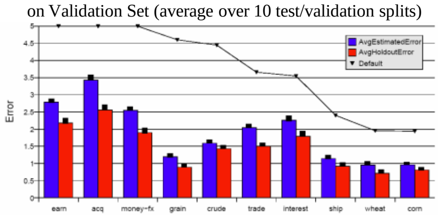
Criterion: $(\alpha_I = C) \Rightarrow (\xi_i < 1\text{-}2CR^2) \Rightarrow (2\alpha_i R^2+\xi_i < 1) \Rightarrow$ Correct



Experiment: Reuters Text Classification

Experiment Setup

- 6451 Training Examples
- 6451 Validation Examples to estimate true Prediction Error
- Comparison between Leave-One-Out upper bound and error on Validation Set (average over 10 test/validation splits)



Fast Leave-One-Out Estimation for SVMs

Lemma: Training errors are always Leave-One-Out Errors.

Algorithm:

- $(R,\alpha,\xi) = \text{trainSVM}(S_{train})$
- FOR $(x_i,y_i) \in S_{train}$
  - IF ξ_i >1 THEN loo++;
  - ELSE IF $(2\,\alpha_i\,R^2 + \xi_i < 1)$ THEN loo = loo;
  - ELSE trainSVM($S_{train} \setminus \{(x_i,y_i)\}$) and test explicitly

Experiment:

Training Sample	Retraining Steps (%)	CPU-Time (sec)
Reuters (n=6451)	0.58%	32.3
WebKB (n=2092)	20.42%	235.4
Ohsumed (n=10000)	2.56%	1132.3