Announcements - Homework

• Homework 1 is graded, please collect at end of lecture

• Homework 2 due today

• Homework 3 out soon (watch email)
  • Ques 1 – midterm review
HW1 score distribution

HW1 total score

0~10 10~20 20~30 30~40 40~50 50~60 60~70 70~80 80~90 90~100 100~110
Announcements - Midterm

• When: Wednesday, 10/20
• Where: In Class
• What: You, your pencil, your textbook, your notes, course slides, your calculator, your good mood :)
• What NOT: No computers, iphones, or anything else that has an internet connection.
• Material: Everything from the beginning of the semester, until, and including SVMs and the Kernel trick
Recitation Tomorrow!

• Boosting, SVM (convex optimization),
  Midterm review!
• Strongly recommended!!
• Place: NSH 3305 (Note: change from last time)
• Time: 5-6 pm
Support Vector Machines

Aarti Singh

Machine Learning 10-701/15-781
Oct 13, 2010
At Pittsburgh G-20 summit ...
Linear classifiers – which line is better?
Pick the one with the largest margin!
Parameterizing the decision boundary

\[ \mathbf{w} \cdot \mathbf{x} = \sum_j w^{(j)} x^{(j)} \quad \mathbf{w} \cdot \mathbf{x} + b > 0 \]

\[ \mathbf{w} \cdot \mathbf{x} + b < 0 \]

Example \( i (= 1, 2, \ldots, n) \):

\[ \langle x^{(1)}_i, \ldots, x^{(m)}_i \rangle \quad \text{— m features} \]

\( y_i \in \{-1, +1\} \quad \text{— class} \)

Data:

\[ \langle x^{(1)}_1, \ldots, x^{(m)}_1, y_1 \rangle \]

\[ \vdots \]

\[ \langle x^{(1)}_n, \ldots, x^{(m)}_n, y_n \rangle \]
Parameterizing the decision boundary

$w \cdot x + b > 0$

$w \cdot x + b < 0$

"confidence" $= (w \cdot x_j + b) y_j$
Maximizing the margin

\[ \text{margin} = \gamma = \frac{2a}{\|w\|} \]

Distance of closest examples from the line/hyperplane
Maximizing the margin

\[ w \cdot x + b > 0 \quad \text{and} \quad w \cdot x + b < 0 \]

Distance of closest examples from the line/hyperplane

\[ \text{margin} = \gamma = \frac{2a}{\|w\|} \]

\[ \max_{w,b} \gamma = \frac{2a}{\|w\|} \]
\[ \text{s.t.} \ (w \cdot x_j + b) y_j \geq a \ \forall j \]

Note: ‘a’ is arbitrary (can normalize equations by a)
Support Vector Machines

\[ w \cdot x + b > 0 \]
\[ w \cdot x + b < 0 \]

\[ w \cdot x + b = 1 \]
\[ w \cdot x + b = 0 \]
\[ w \cdot x + b = -1 \]

\[ \min_{w,b} w \cdot w \]
\[ \text{s.t.} \ (w \cdot x_j + b) y_j \geq 1 \ \forall j \]

Solve efficiently by quadratic programming (QP)
- Well-studied solution algorithms

Linear hyperplane defined by “support vectors”
Support Vectors

\[ w \cdot x + b > 0 \quad \text{and} \quad w \cdot x + b < 0 \]

Linear hyperplane defined by “support vectors"

Moving other points a little doesn’t effect the decision boundary

only need to store the support vectors to predict labels of new points

How many support vectors in linearly separable case?

\[ \leq m+1 \]
What if data is not linearly separable?

Use features of features of features of features....

$x_1^2, x_2^2, x_1x_2, \ldots, \exp(x_1)$

But run risk of overfitting!
What if data is still not linearly separable?

Allow “error” in classification

\[
\min_{w,b} w \cdot w + C \text{ # mistakes} \\
\text{s.t. } (w \cdot x_j + b) y_j \geq 1 \quad \forall j
\]

Maximize margin and minimize # mistakes on training data

C - tradeoff parameter

Not QP 😞

0/1 loss (doesn’t distinguish between near miss and bad mistake)
What if data is still not linearly separable?

Allow “error” in classification

\[
\min_{\mathbf{w}, b} \mathbf{w} \cdot \mathbf{w} + C \sum_{j} \xi_j \\
\text{s.t. } (\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1 - \xi_j \quad \forall j \\
\xi_j \geq 0 \quad \forall j
\]

\(\xi_j\) - “slack” variables
\(= (>1 \text{ if } x_j \text{ misclassified})\)

Pay linear penalty if mistake

C - tradeoff parameter (chosen by cross-validation)

Still QP 😊
Slack variables – Hinge loss

Complexity penalization

\[ \xi_j = \text{loss}(f(x_j), y_j) \]

\[ f(x_j) = \text{sgn}(w \cdot x_j + b) \]

\[ \xi_j = (1 - (w \cdot x_j + b)y_j)_+ \]

Hinge loss

\[
\min_{w,b} w \cdot w + C \sum_j \xi_j \\
\text{s.t. } (w \cdot x_j + b) y_j \geq 1 - \xi_j \forall j \\
\xi_j \geq 0 \forall j
\]
SVM vs. Logistic Regression

**SVM** : Hinge loss

\[
\text{loss}(f(x_j), y_j) = (1 - (w \cdot x_j + b)y_j))^+ 
\]

**Logistic Regression** : Log loss (−ve log conditional likelihood)

\[
\text{loss}(f(x_j), y_j) = -\log P(y_j \mid x_j, w, b) = \log(1 + e^{-(w \cdot x_j + b)y_j}) 
\]
What about multiple classes?
One against all

Learn 3 classifiers separately:
Class k vs. rest

\[(w_k, b_k)_{k=1,2,3}\]

\[y = \arg\max_k w_k x + b_k\]

But \(w_k\)s may not be based on the same scale.

Note: \((aw)x + (ab)\) is also a solution
Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

\[ w^{(y_j)} \cdot x_j + b^{(y_j)} \geq w^{(y')} \cdot x_j + b^{(y')} + 1, \quad \forall y' \neq y_j, \quad \forall j \]

Margin - gap between correct class and nearest other class

\[ y = \arg \max w^{(k)} \cdot x + b^{(k)} \]
Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

\[
\begin{align*}
\text{minimize}_{w, b} & \quad \sum_y w(y) \cdot w(y) + C \sum_j \sum_{y \neq y_j} \xi_j(y) \\
\text{subject to} & \quad w(y_j) \cdot x_j + b(y_j) \geq w(y) \cdot x_j + b(y) + 1 - \xi_j(y), \quad \forall y \neq y_j, \forall j \\
\xi_j(y) & \geq 0
\end{align*}
\]

\[y = \arg \max w^{(k)} \cdot x + b^{(k)}\]

Joint optimization: \[w_k\]s have the same scale.
What you need to know

• Maximizing margin
• Derivation of SVM formulation
• Slack variables and hinge loss
• Relationship between SVMs and logistic regression
  – 0/1 loss
  – Hinge loss
  – Log loss
• Tackling multiple class
  – One against All
  – Multiclass SVMs
SVMs reminder

Regularization  Hinge loss

\[
\begin{align*}
\min_{w,b} & \quad w \cdot w + C \sum \xi_j \\
\text{s.t.} & \quad (w \cdot x_j + b) y_j \geq 1 - \xi_j \quad \forall j \\
& \quad \xi_j \geq 0 \quad \forall j
\end{align*}
\]

Soft margin approach
Today’s Lecture

• Learn one of the most interesting and exciting recent advancements in machine learning
  – The “kernel trick”
  – High dimensional feature spaces at no extra cost!

• But first, a detour
  – Constrained optimization!
Constrained Optimization

\[ \min_x x^2 \]

s.t. \( x \geq b \)

\[ \min_x x^2 \quad \text{s.t.} \quad x \geq -1 \]

\[ \min_x x^2 \quad \text{s.t.} \quad x \geq 1 \]

\( x^* = 0 \)

\( x^* = 0 \)

\( x^* = 1 \)
Lagrange Multiplier – Dual Variables

\[ \min_x x^2 \]
\[ \text{s.t. } x \geq b \]

Moving the constraint to objective function

Lagrangian:
\[ L(x, \alpha) = x^2 - \alpha(x - b) \]
\[ \text{s.t. } \alpha \geq 0 \]

Solve:
\[ \min_x \max_{\alpha} L(x, \alpha) \]
\[ \text{s.t. } \alpha \geq 0 \]

Constraint is tight when \( \alpha > 0 \)
Duality

Primal problem:
\[ f^* = \min_x f(x) \]
\[ \text{s.t. } x^2 \]

Dual problem:
\[ g^* = \min_x \max_\alpha g(x) \]
\[ \text{s.t. } x^2 - \alpha(x - b) \]
\[ \text{s.t. } \alpha \geq 0 \]

Weak duality \(- g^* \leq f^* \)

For all feasible points \( \tilde{x} \)
\[ g^* \leq g(\tilde{x}) \leq f(\tilde{x}) \]

Strong duality \(- g^* = f^* \) (holds under KKT conditions)
Lagrange Multiplier – Dual Variables

\[ L(x, \alpha) = x^2 - \alpha(x - b) \]

\[ \min_x \max_\alpha \quad \text{s.t.} \quad \alpha \geq 0 \]

Solving:

\[ \frac{\partial L}{\partial x} = 0 \quad \Rightarrow \quad x^* = \frac{\alpha}{2} \]

\[ \frac{\partial L}{\partial \alpha} = 0 \quad \Rightarrow \quad \alpha^* = \max(2b, 0) \]

When \( \alpha > 0 \), constraint is tight