CS485/685
Lecture 12: Feb 9, 2012

Support Vector Machines (continued)

[B] Section 7.1

Overlapping Class Distributions

- So far we assumed that data is linearly separable
  - High dimensions help for linear separability, but may hurt for generalization
- But what if the data is noisy (mistakes or outliers)
  - Constraints should allow misclassifications
- Picture

Soft margin

- Idea: relax constraints by introducing slack variables $\xi_n \geq 0$

$$y_n(\boldsymbol{w}^{\boldsymbol{T}} \phi(\boldsymbol{x}_{\boldsymbol{n}}) + b) \geq 1 - \xi_n \quad \forall n$$

- Picture:

Soft margin classifier

- New optimization problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad C \sum_{n=1}^{N} \xi_n + \frac{1}{2} ||\boldsymbol{w}||^2$$

s.t. $y_n(\boldsymbol{w}^{\boldsymbol{T}} \phi(\boldsymbol{x}_{\boldsymbol{n}}) + b) \geq 1 - \xi_n$

and $\xi_n \geq 0 \quad \forall n$

- where $C > 0$ controls the trade-off between the slack variable penalty and the margin

Soft margin classifier

- Notes:
  1. Since $\sum_n \xi_n$ is an upper bound on the # of misclassifications, C can also be thought as a regularization coefficient that controls the trade-off between error minimization and model complexity
  2. When $C \to \infty$, then we recover the original hard margin classifier
  3. Soft margins handle minor misclassifications, but the classifier is still very sensitive to outliers

Support Vectors

- As before support vectors correspond to active constraints

$$y_n(\boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x_n}) + b) = 1 - \xi_n$$

  – i.e., all points that are in the margin or misclassified
- Picture:

Dual derivation

- Transform constrained optimization

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\left||\boldsymbol{w}|\right|^2$$
$$\text{s.t. } y_n(\boldsymbol{w}^T\phi(\boldsymbol{x_n}) + b) \geq 1 - \xi_n \text{ and } \xi_n \geq 0 \quad \forall n$$

into an unconstrained optimization problem

- Lagrangian

$$\max_{\boldsymbol{a},\boldsymbol{\mu}} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{\mu}) \text{ s.t. } \boldsymbol{a}, \boldsymbol{\mu} \geq 0$$

where $L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{\mu}) = C\sum_n \xi_n + \frac{1}{2}\left||\boldsymbol{w}|\right|^2$

$$- \sum_n a_n[y_n(\boldsymbol{w}^T\phi(\boldsymbol{x_n}) + b) - 1 + \xi_n] - \sum_n \mu_n\xi_n$$

Dual derivation

- Solve $\min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{\mu})$ by setting derivatives to 0

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_n a_n y_n \phi(\boldsymbol{x}_n)$$
$$\frac{\partial L}{\partial b} = 0 \implies 0 = \sum_n a_n y_n$$
$$\frac{\partial L}{\partial \xi_n} = 0 \implies a_n = C - \mu_n$$

- Eliminate w, b, ξ and μ based on these conditions:

$$L(\boldsymbol{a}) = \sum_n a_n - \frac{1}{2}\sum_n\sum_m a_n a_m y_n y_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

Dual Problem

- The resulting **dual problem** is

$$\max_{\boldsymbol{a}} L(\boldsymbol{a})$$
$$\text{s.t.} \sum_n a_n y_n = 0$$
$$0 \le \underbrace{a_n \le C} \quad \forall n$$

- NB: Same optimization problem as for hard margins, except for the upper bound on a_n

Dual Problem

- Notes:

1. $a_n = 0 \Longrightarrow$ irrelevant point
2. $a_n > 0 \Longrightarrow$ support vector
   a. $a_n < C \Longrightarrow$ point on the margin
   b. $a_n = C \Longrightarrow$ point inside the margin or misclassified

Classification

- Same as for hard margins
- Primal problem

$$y = sign(\boldsymbol{w}^T \phi(\boldsymbol{x}) + b)$$

- Dual problem

$$y = sign\left(\sum_n a_n y_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b\right)$$

Multiclass SVMs

- Three methods:
  1. One-against-all: for R classes, train R SVMs to distinguish each class from the rest
  2. Continuous ranking: single SVM that returns a continuous value to rank all classes
  3. Pairwise comparison: train $O(R^2)$ SVMs to compare each pair of classes

One-Against-All

- For R classes, train R SVMs to distinguish each class from the rest
- Picture:
- Problem: what if different classes are returned by different SVMs?

Continuous Ranking

- Single SVM that returns a continuous value to rank all classes
- Picture:
- Most popular approach today

Pairwise Comparison

- Train $O(R^2)$ SVMs to compare each pair of classes
- Picture:

- Problem: how do we pick the best class?

Continuous Ranking

- Idea: instead of computing the sign of a linear separator, compare the values of linear functions for each class r

$$y = argmax_r\ \boldsymbol{w}_r^T \phi(\boldsymbol{x}) + b_r$$

Multiclass Margin

- For each class r define a linear constraint:

$$\boldsymbol{w}_y^T \phi(\boldsymbol{x}) + b_y + \delta_{yr} - \boldsymbol{w}_r^T \phi(\boldsymbol{x}) - b_r \geq 1 \quad \forall r$$

where $\delta_{yr} = \begin{cases} 1 & y = r \\ 0 & \text{otherwise} \end{cases}$

- This guarantees a margin of at least 1

Multiclass Classification

- Optimization problem:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{2} \sum_r ||\boldsymbol{w}_r||^2$$

s.t. $\boldsymbol{w}_{y_n}^T \phi(\boldsymbol{x}_n) + b_{y_n} + \delta_{y_n r} - \boldsymbol{w}_r^T \phi(\boldsymbol{x}) - b_r \geq 1 \quad \forall n, r$

- Equivalent to binary SVM when we have only two classes

Overlapping classes

- Add slack variables:

$$\min_{\boldsymbol{W},\boldsymbol{b},\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\sum_r \left|\left|\boldsymbol{w}_r\right|\right|^2$$
s.t. $\mathbf{w}_{y_n}^T \phi(\boldsymbol{x}_n) + b_{y_n} + \delta_{y_n r} - \boldsymbol{w}_r^T \phi(\boldsymbol{x}) - b_r \geq 1 - \xi_n \;\; \forall n, r$

- Equivalent to binary SVM when we have only two classes

Dual representation

- Kernelized form

$$\max_{\{\boldsymbol{a}_n\}} \sum_n \boldsymbol{\tau}_n^T \boldsymbol{e}_{y_n} - C\frac{1}{2}\sum_n \sum_m \boldsymbol{\tau}_n^T \boldsymbol{\tau}_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$
s.t. $\boldsymbol{\tau}_n \leq \boldsymbol{e}_{y_n}$ and $\sum_r \tau_{nr} = 0$

where $e_{y_n r} = \begin{cases} 1 & \text{when } y_n = r \\ 0 & \text{otherwise} \end{cases}$

and $\boldsymbol{\tau}_n = \boldsymbol{e}_{y_n} - \boldsymbol{a}_n$

Classification

- Primal

$$y = argmax_r \; \boldsymbol{w}_r^T \phi(\boldsymbol{x}) + b_r$$

- Dual

$$y = argmax_r \; \sum_n \tau_{y_n r} k(\boldsymbol{x}_n, \boldsymbol{x})$$