CS489/698
Lecture 12: February 13, 2017

Support Vector Machines

[B] Sec. 7.1 [D] Sec. 11.5-11.6
[HTF] Chap. 12 [M] Sec. 14.5 [RN] 18.9
[MRT] Chap. 4

Sparse kernel techniques

- Kernel based approaches: complexity depends on the amount of data, not the dimensionality of the space. But for large datasets, this is not practical.
  - Kernel matrix is square in # of data points
  - Prediction requires inversion of the kernel matrix, which is cubic in # of data points

- Can we use a **sparse representation**?
  - i.e., kernel that depends on a subset of the data

Support Vector Machines

- Kernel depends on subset of data
- Picture

Max-Margin Classifier

- Find linear separator that maximizes the distance (or margin) to closest data points
- Picture

Margin

- Linear separator: $\boldsymbol{w}^T \phi(\boldsymbol{x}) = 0$

- Distance to linear separator:

$$\frac{y\boldsymbol{w}^T \phi(\boldsymbol{x})}{||\boldsymbol{w}||} \quad \text{where } y \in \{-1,1\}$$

- Maximum margin:

$$max_{\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||} \left\{ \min_n y_n \, \boldsymbol{w}^{\boldsymbol{T}} \phi(\boldsymbol{x}_{\boldsymbol{n}}) \right\}$$

Comparison

Perceptron

Support Vector Machine

Maximum Margin

- Unique max margin linear separator

$$max_{\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||} \left\{ \min_n y_n \, \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \right\}$$

- Alternatively, we can fix the minimal distance to 1 and minimize $||\boldsymbol{w}||$

$$\min_{\boldsymbol{w}} \ \frac{1}{2} ||\boldsymbol{w}||^2$$
$$\text{s.t. } y_n \, \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \geq 1 \quad \forall n$$

- This is a convex quadratic optimization problem that can easily be solved by many optimization packages

Derivation

$$argmax_{\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||}\left\{\min_n y_n\, \boldsymbol{w}^T\phi(\boldsymbol{x}_n)\right\}$$

$$= argmax_{\boldsymbol{w},\delta} \frac{1}{||\boldsymbol{w}||}\delta \quad \text{s.t. } y_n\, \boldsymbol{w}^T\phi(\boldsymbol{x}_n) \geq \delta \;\; \forall n$$

$$= argmax_{\boldsymbol{w},\delta} \frac{1}{\left|\left|\frac{\boldsymbol{w}}{\delta}\right|\right|} \quad \text{s.t. } y_n \frac{\boldsymbol{w}^T}{\delta}\phi(\boldsymbol{x}_n) \geq 1 \;\; \forall n$$

replace $\frac{\boldsymbol{w}}{\delta}$ by $\boldsymbol{w}'$

$$= argmax_{\boldsymbol{w}'} \frac{1}{||\boldsymbol{w}'||} \quad \text{s.t. } y_n \boldsymbol{w}'^T\phi(\boldsymbol{x}_n) \geq 1 \;\; \forall n$$

$$= argmin_{\boldsymbol{w}'} ||\boldsymbol{w}'|| \quad \text{s.t. } y_n \boldsymbol{w}'^T\phi(\boldsymbol{x}_n) \geq 1 \;\; \forall n$$

$$= argmin_{\boldsymbol{w}'} \frac{1}{2}||\boldsymbol{w}'||^2 \quad \text{s.t. } y_n \boldsymbol{w}'^T\phi(\boldsymbol{x}_n) \geq 1 \;\; \forall n$$

Support Vectors

- Quadratic optimization problem

$$\min_{\boldsymbol{w}} \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2$$
$$\text{s.t. } y_n\, \boldsymbol{w}^{\boldsymbol{T}} \phi(\boldsymbol{x}_{\boldsymbol{n}}) \geq 1 \quad \forall n$$

- Only the points where $y_n\, \boldsymbol{w}^{\boldsymbol{T}} \phi(\boldsymbol{x}_n) = 1$ are necessary. These points define the active constraints and are known as the **support vectors**

Dual representation

- Idea: reformulation where $\phi(\boldsymbol{x})$ appears only in a kernel
- Approach: find the dual of the optimization problem
- Result: (sparse) kernel support vector machines

Dual derivation

- Transform constrained optimization

$$\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 \quad \text{s.t. } y_n\, \boldsymbol{w}^T \phi(\boldsymbol{x_n}) \geq 1 \quad \forall n$$

into an unconstrained optimization problem

- Lagrangian

$$\max_{\boldsymbol{a}\geq\boldsymbol{0}} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{a})$$

where $L(\boldsymbol{w}, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_n \underbrace{a_n[y_n\, \boldsymbol{w}^T \phi(\boldsymbol{x_n}) - 1]}_{\text{penalty for violating the } n^{th} \text{ constraint}}$

Dual derivation

- Solve inner minimization: $\min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{a})$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\left||\boldsymbol{w}|\right|^2 - \sum_n a_n[y_n\, \boldsymbol{w}^T \phi(\boldsymbol{x_n}) - 1]$$

- Set derivative to 0

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

- Substitute $\boldsymbol{w}$ by $\sum_n a_n y_n \phi(\boldsymbol{x}_n)$ in $L(\boldsymbol{w}, \boldsymbol{a})$ to obtain:

$$L(\boldsymbol{a}) = \sum_n a_n - \frac{1}{2}\sum_n \sum_{n'} a_n a_{n'} y_n y_{n'} k(\boldsymbol{x}_n, \boldsymbol{x}_{n'})$$

Dual Problem

- We are then left with an optimization in $\boldsymbol{a}$ only known as the **dual problem**

$$\max_{\boldsymbol{a}} L(\boldsymbol{a})$$
$$\text{s.t. } a_n \geq 0$$

- **Sparse optimization**: many a_n's are 0

Classification

- Primal problem

$$y_* = sign(\boldsymbol{w}^T \phi(\boldsymbol{x}_*))$$

- Dual problem

$$y_* = sign\left(\sum_n a_n y_n \phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_*)\right)$$

$$y_* = sign\left(\sum_n a_n y_n k(\boldsymbol{x}_n, \boldsymbol{x}_*)\right)$$

Generalization

- Support vector machines generalize quite well
  - i.e., overfitting is rare

- Reason: maximizing the margin is equivalent to minimizing an upper bound on the worst case loss (worst loss for any underlying input distribution).

Case Study: Text Categorization

- T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998.

- Early success that helped SVMs become popular

Text Categorization

- **Problem:** how to categorize a new article as finance, sports, politics, science, health, etc.?

- **Idea:** train a classifier with archives of news articles that have already been classified

Representation

- How should we represent a document?

- Idea: vector of word counts (vector space model)

$$
\begin{pmatrix}
 & T_1 & T_2 & \cdots & T_t \\
D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\
D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \cdots & w_{tn}
\end{pmatrix}
$$

terms

documents

frequencies

Challenges

- **High dimensional input space**:
  – Length of vector is # of words in dictionary (e.g., 10,000)
- **Few irrelevant features**:
  – Most words carry some information that reflect their meaning

- Need an approach that scales well with input dimensionality: **support vector machines**

Experiment

- [Joachim 98]
  - Data: Reuters dataset
  - Compare precision/recall breakeven point
    - i.e., precision = recall
    - Precision: $\frac{|\{relevant\ docs\}| \cap |\{retrieved\ docs\}|}{|\{relevant\ docs\}|}$
    - Recall: $\frac{|\{relevant\ docs\}| \cap |\{retrieved\ docs\}|}{|\{retrieved\ docs\}|}$
  - Algorithms
    - Naïve Bayes: 72.0%
    - Decision trees: 79.4%
    - Rochio: 79.9%
    - K-Nearest Neighbors: 82.3%
    - SVMs: 86.0% (polynomial kernel), 86.4% (Gaussian kernel)

SVM summary

- Find (generalized) linear separator
  - Dual representation (kernel): non-linear separator
- Unique max-margin separator
  - Good generalization
- Convex quadratic optimization
  - Polynomial complexity
  - Global optimality
- Sparse optimization
  - many variables are 0
- Can we do multi-class classification?
- Can we handle data that is not linearly separable?