

CS485/685

Lecture 15: March 1, 2016

Support Vector Machines

[B] Sec. 7.1 [D] Sec. 6.7 [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: $\mathbf{w}^T \phi(\mathbf{x}) = 0$
- Distance to linear separator:

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

- Maximum margin:

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

Comparison

Perceptron

Support Vector Machine

Maximum Margin

- Unique max margin linear separator

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

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

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_n \mathbf{w}^T \phi(\mathbf{x}_n) \geq 1 \quad \forall n \end{aligned}$$

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

Derivation

$$\begin{aligned} & \operatorname{argmax}_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|} \left\{ \min_n y_n \mathbf{w}^T \phi(\mathbf{x}_n) \right\} \\ &= \operatorname{argmax}_{\mathbf{w}, \delta} \frac{1}{\|\mathbf{w}\|} \delta \quad \text{s.t. } y_n \mathbf{w}^T \phi(\mathbf{x}_n) \geq \delta \quad \forall n \\ &= \operatorname{argmax}_{\mathbf{w}, \delta} \frac{1}{\left\| \frac{\mathbf{w}}{\delta} \right\|} \quad \text{s.t. } y_n \frac{\mathbf{w}^T}{\delta} \phi(\mathbf{x}_n) \geq 1 \quad \forall n \end{aligned}$$

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

$$\begin{aligned} &= \operatorname{argmax}_{\mathbf{w}'} \frac{1}{\|\mathbf{w}'\|} \quad \text{s.t. } y_n \mathbf{w}'^T \phi(\mathbf{x}_n) \geq 1 \quad \forall n \\ &= \operatorname{argmin}_{\mathbf{w}'} \|\mathbf{w}'\| \quad \text{s.t. } y_n \mathbf{w}'^T \phi(\mathbf{x}_n) \geq 1 \quad \forall n \\ &= \operatorname{argmin}_{\mathbf{w}'} \frac{1}{2} \|\mathbf{w}'\|^2 \quad \text{s.t. } y_n \mathbf{w}'^T \phi(\mathbf{x}_n) \geq 1 \quad \forall n \end{aligned}$$

Support Vectors

- Quadratic optimization problem

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_n \mathbf{w}^T \phi(\mathbf{x}_n) \geq 1 \quad \forall n \end{aligned}$$

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

Dual representation

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

Dual derivation

- Transform constrained optimization

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

into an unconstrained optimization problem

- Lagrangian

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

$$\text{where } L(\mathbf{w}, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_n a_n \underbrace{[y_n \mathbf{w}^T \phi(\mathbf{x}_n) - 1]}$$

penalty for violating
the n^{th} constraint

Dual derivation

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

$$\min_{\mathbf{w}} \quad \frac{1}{2} \|\mathbf{w}\|^2 - \sum_n a_n [y_n \mathbf{w}^T \phi(\mathbf{x}_n) - 1]$$

- Set derivative to 0

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

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

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

Dual Problem

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

$$\begin{aligned} \max_{\mathbf{a}} L(\mathbf{a}) \\ \text{s.t. } a_n \geq 0 \end{aligned}$$

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

Classification

- Primal problem

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

- Dual problem

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

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

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

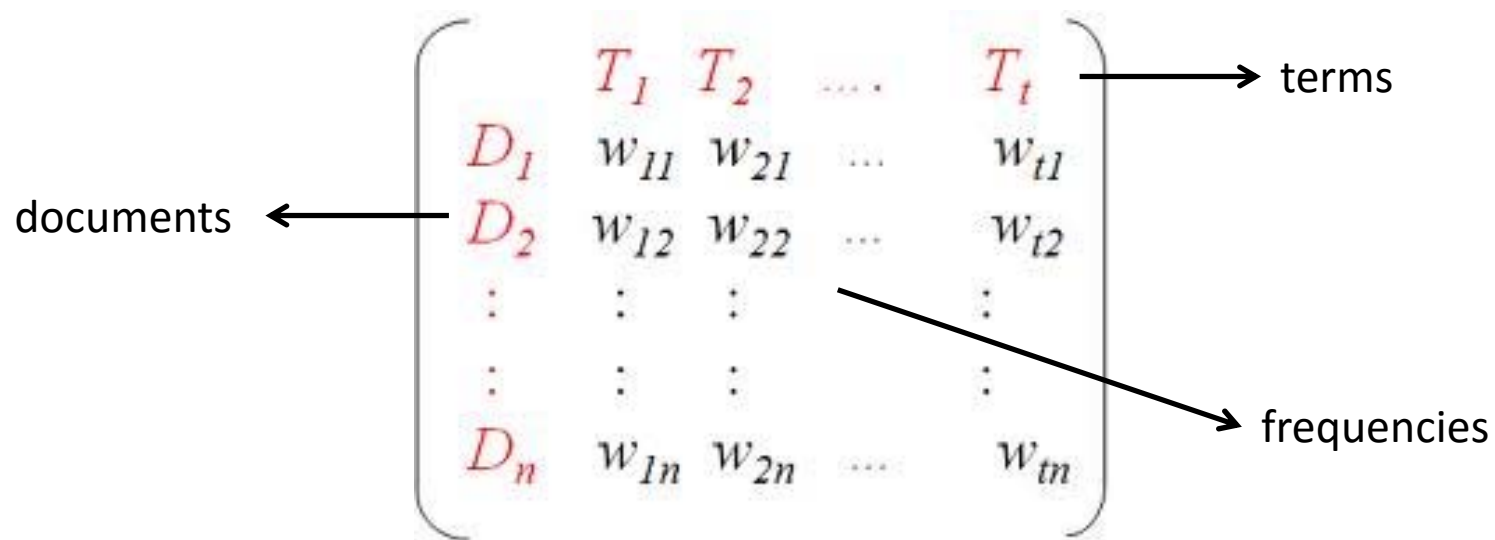
- T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998.
- Early success that helped SVMs become popular

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)



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: $\frac{|\{\text{relevant docs}\} \cap \{\text{retrieved docs}\}|}{|\{\text{retrieved docs}\}|}$
 - Recall: $\frac{|\{\text{relevant docs}\} \cap \{\text{retrieved docs}\}|}{|\{\text{relevant docs}\}|}$
 - 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?