Regularization

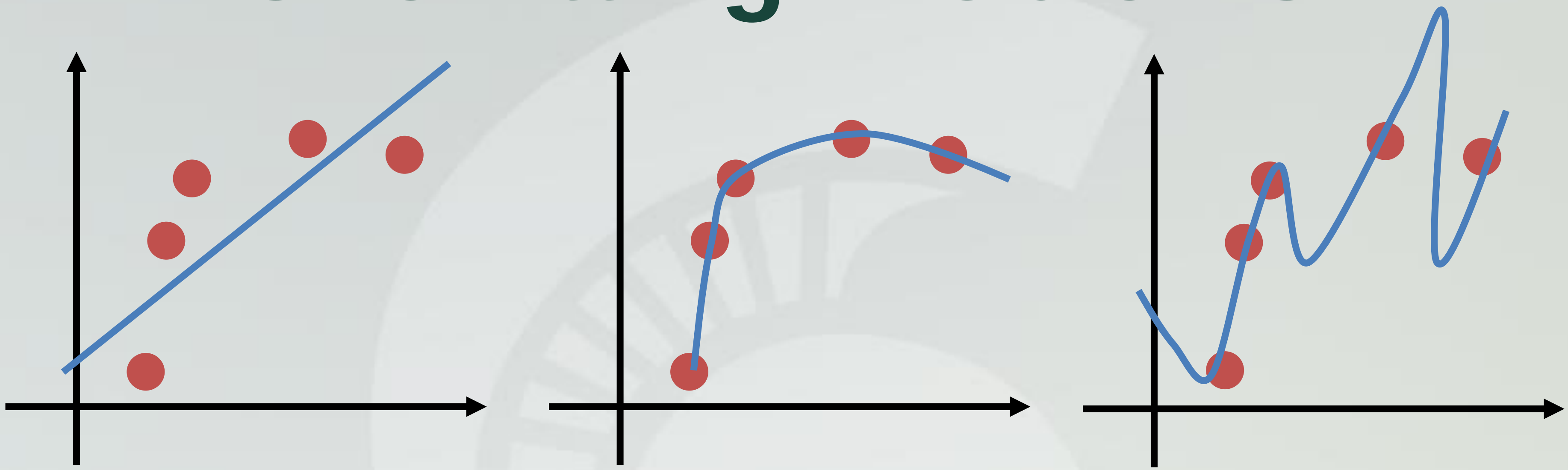
Guowei Wei
Department of Mathematics
Michigan State University

References:
Duc D. Nguyen's lecture notes
Andrew Ng's notes
Wikipedia

Introduction

- Minimize the magnitude of parameters
- Eliminate the overfitting problems

Overfitting Problems



$p(x) = c_0 + c_1 x$
(Underfit, high bias)

$p(x) = c_0 + c_1 x + c_2 x^2$
(Just right)

$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$
(Overfit, high variance)

- Overfitting: The *predictor* may perfectly fit the training set but fail to *predict* new examples

Avoiding Overfitting

- Reduce the number of features
  - Manually select which features to keep
  - Model selection algorithm
- Regularization
  - Keep all the features, but reduce the magnitude of parameters

$$\mathbf{c} = (c_0, c_{\mathbf{1}}, \ldots)$$

Regularized Loss Function

- Linear Regression

$$L(\mathbf{c}) = \sum_{i=1}^{m} \left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2$$

- Regularized Linear Regression

$$L(\mathbf{c}) = \sum_{i=1}^{m} \left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n} c_j^2$$

Regularized Loss Function

- Logistic Regression

$$L(\mathbf{c}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right)\right)\right]$$

- Regularized Logistic Regression

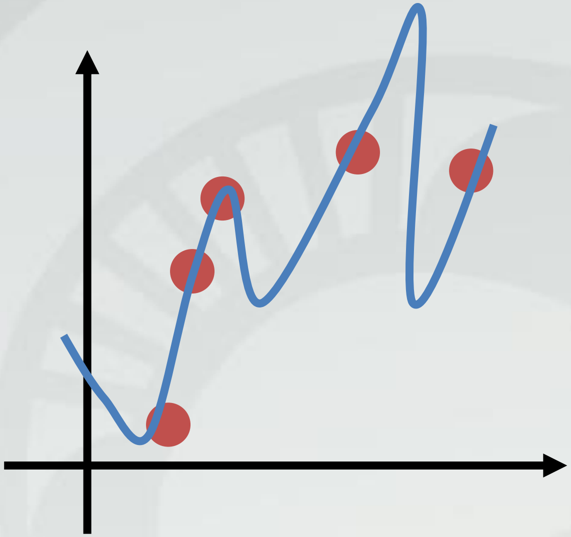
$$L(\mathbf{c}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right)\right)\right]$$

$$+\frac{\lambda}{2n}\sum_{j=1}^{n} c_j^2$$

Do not include bias c_0

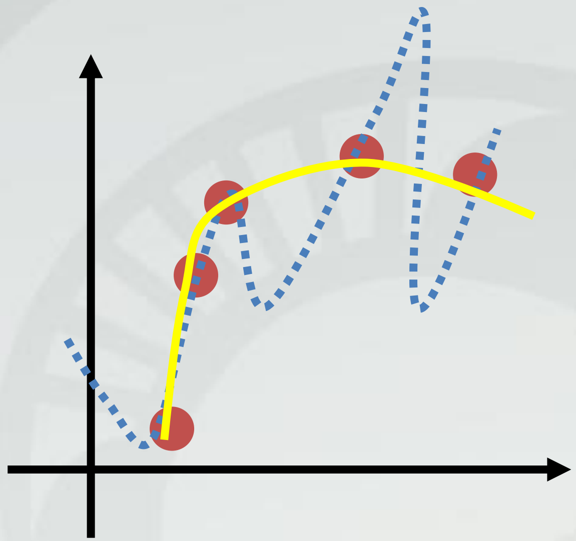
- Use gradient descent for optimization

Regularized Loss Function



$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$$

Regularized Loss Function



$$p(x) = c_0 + c_1 x + c_2 x^2 + 0x^3 + 0x^4$$

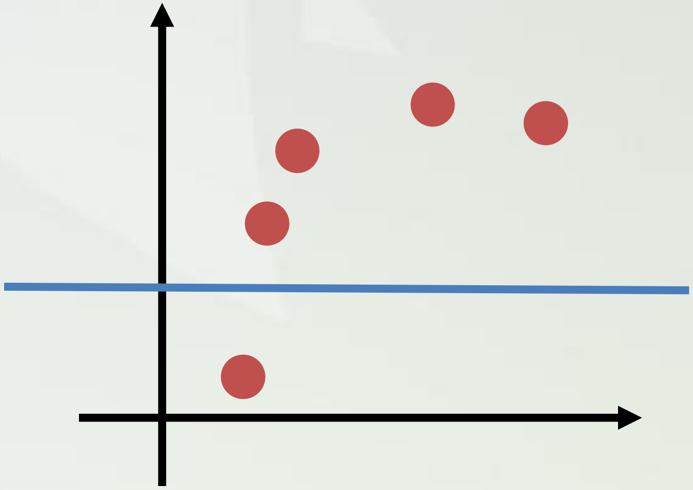
How big is λ?

- Linear Regression

$$L(\mathbf{c}) = \sum_{i=1}^{m}\left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n} c_j^2$$

- What happen when λ is too big, $\lambda = 10^6$
- $c_j = 0,\ j = 1,2,\ldots,n \Rightarrow p_{\mathbf{c}}(\mathbf{x}) = c_0$ (constant)

Therefore underfitting



Discussions

- Square loss function:

$$L(p(\mathbf{x}), y) = \left(1 - yp(\mathbf{x})\right)^2$$

- Hinge loss function (used in SVM):

$$L(p(\mathbf{x}), y) = \max(0, 1 - yp(\mathbf{x}))$$

- Tikhonov regularization:

If $\mathrm{p}(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x}$, $\min \sum_{i=1}^{m} L\left(\mathbf{x}^{(i)} \cdot \boldsymbol{c}, y^{(i)}\right) + \lambda \|\boldsymbol{c}\|_2^2$

In general, $\min \sum_{i=1}^{m} L\left(p(\mathbf{x}^{(i)}), y^{(i)}\right) + \lambda \|p\|_2^2$

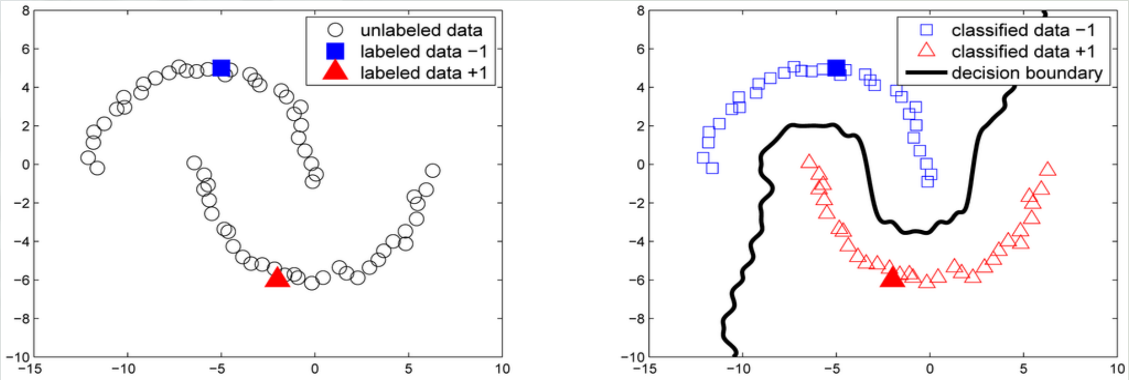
- LASSO: $\min \sum_{i=1}^{m} \frac{1}{m} \left\|\mathbf{x}^{(i)} \cdot \boldsymbol{c} - y^{(i)}\right\| + \lambda \|\boldsymbol{c}\|_1$

(Least absolute shrinkage and selection operator)

Discussions

- Regularizers for Semi-Supervised Learning

$$\min\left[\sum_{i=1}^{m} L\left(p(\mathbf{x}^{(i)}), y^{(i)}\right) + \lambda R\right]$$



The regularizer:

- $R = \sum_{i,j}^{m} w_{ij}\left(p(\mathbf{x}^{(i)}) - p(\mathbf{x}^{(j)})\right)^2 = \mathbf{p}^T L\, \mathbf{p}$
- $L = D - A$: Laplacian matrix.
- D: Degree matrix
- A: Adjacency matrix