CS485/685 Machine Learning
Lecture 4: Jan 12, 2012

Linear Regression

[B] Section 3.1

Linear model for regression

- Simplest form of regression
- Picture:

Problem

- Data: $\{(\boldsymbol{x_1}, t_1), (\boldsymbol{x_2}, t_2), \dots, (\boldsymbol{x_N}, t_N)\}$
  - $\boldsymbol{x} = < x_1, x_2, \dots, x_D >$: input vector
  - t: target (continuous value)
- Problem: find hypothesis h that maps $\boldsymbol{x}$ to t
  - Assume that h is linear:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D = \boldsymbol{w}^{\boldsymbol{T}} \binom{1}{\boldsymbol{x}}$$

- Objective: minimize some loss function
  - Euclidean loss: $L_2(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(\boldsymbol{x_n}, \boldsymbol{w}) - t_n)^2$

Optimization

- Find best w that minimizes Euclidean loss

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}} \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^{\boldsymbol{T}} \binom{1}{\boldsymbol{x_n}} \right)^2$$

- Convex optimization problem
  $\Longrightarrow$ unique optimum (global)

Solution

- Let $\overline{\boldsymbol{x}} = \binom{1}{\boldsymbol{x}}$ then $\min_{\boldsymbol{w}} \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right)^2$
- Find $\boldsymbol{w}^*$ by setting the derivative to 0

  $\frac{\partial L_2}{\partial w_j} = \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right) \bar{x}_{nj} = 0 \quad \forall j$

  $\Longrightarrow \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right) \overline{\boldsymbol{x}}_{\boldsymbol{n}} = 0$
- This is a linear system in $\boldsymbol{w}$, therefore we rewrite it as $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$

  where $\boldsymbol{A} = \sum_{n=1}^{N} \overline{\boldsymbol{x}}_{\boldsymbol{n}} \overline{\boldsymbol{x}}_{\boldsymbol{n}}^{\boldsymbol{T}}$ and $\boldsymbol{b} = \sum_{n=1}^{N} t_n \overline{\boldsymbol{x}}_{\boldsymbol{n}}$

Solution

- If training instances span $\Re^{D+1}$ then $\boldsymbol{A}$ is invertible:

$$\boldsymbol{w} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

- In practice it is faster to solve the linear system $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$ directly instead of inverting $\boldsymbol{A}$
  - Gaussian elimination
  - Conjugate gradient
  - Iterative methods

Picture

Regularization

- Least square solution may not be stable
  - i.e., slight perturbation of the input may cause a dramatic change in the output
  - Form of **overfitting**

Example 1

- Training data: $\bar{\boldsymbol{x}}_1 = \begin{pmatrix}1\\0\end{pmatrix}$ $\bar{\boldsymbol{x}}_2 = \begin{pmatrix}1\\\epsilon\end{pmatrix}$
  $t_1 = 1$ $t_2 = 1$
- $\boldsymbol{A} =$
- $\boldsymbol{A}^{-1} =$ $\boldsymbol{b} =$
- $\boldsymbol{w} =$

CS485/685 (c) 2012 P. Poupart 9

Example 2

- Training data: $\bar{\boldsymbol{x}}_1 = \begin{pmatrix}1\\0\end{pmatrix}$ $\bar{\boldsymbol{x}}_2 = \begin{pmatrix}1\\\epsilon\end{pmatrix}$
  $t_1 = 1 + \epsilon$ $t_2 = 1$
- $\boldsymbol{A} =$
- $\boldsymbol{A}^{-1} =$ $\boldsymbol{b} =$
- $\boldsymbol{w} =$

CS485/685 (c) 2012 P. Poupart 10

Picture

Regularization

- Idea: favor smaller values
- Tikhonov regularization: add $||\boldsymbol{w}||_2^2$ as a penalty term
- Ridge regression:

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}} \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^{\boldsymbol{T}} \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right)^2 + \frac{\lambda}{2} ||\boldsymbol{w}||_2^2$$

where λ is a weight to adjust the importance of the penalty

Regularization

- Solution: $(\lambda \boldsymbol{I} + \boldsymbol{A})\boldsymbol{w} = \boldsymbol{b}$

- Notes
  - Without regularization: eigenvalues of linear system may be arbitrarily close to 0 and the inverse may have arbitrarily large eigenvalues.
  - With Tikhonov regularization, eigenvalues of linear system are $\geq \lambda$ and therefore bounded away from 0. Similarly, eigenvalues of inverse are bounded above by $1/\lambda$.

Regularized Examples

Example 1

Example 2

Generalized Linear Regression

- How can we do non-linear regression while using the same machinery?
- Idea: map inputs to a different space and do linear regression in that space

Example

- Suppose the underlying function is quadratic

Basis functions

- Use non-linear basis functions:
  - Let ϕ_i denote a basis function

$$\phi_0(x) = 1$$
$$\phi_1(x) = x$$
$$\phi_2(x) = x^2$$

  - Let the hypothesis space H be

$$H = \{x \to w_0\phi_0(x) + w_1\phi_1(x) + w_2\phi_2(x) | w_i \in \Re\}$$

- If the basis functions are non-linear in x, then a non-linear hypothesis can still be found by linear regression

Common basis functions

- Polynomial: $\phi_j(x) = x^j$
- Gaussian: $\phi_j(x) = e^{-\frac{(x-\mu_j)^2}{2s^2}}$
- Sigmoid: $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$
  where $\sigma(a) = \frac{1}{1+e^{-a}}$
- Also Fourier basis functions, wavelets, etc.

Next class

- Linear regression by
  - Maximum likelihood estimation (ML)
  - Maximum a posteriori estimation (MAP)
  - Bayesian learning