Supervised Learning

LIN Juntong

Oct 24, 2016

Agenda

- Linear Models for Regression
  - Linear Regression
  - Probabilistic Interpretation
  - Generalized Linear Regression
- Discriminative Classification
  - Logistic Regression
- Generative Classification
  - Gaussian Discriminative Analysis
  - Naive Bayes

Linear Regression

Linear Regression

- big picture of machine learning
- training set
  $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\},\ \mathcal{Y} = \{y^{(1)}, \ldots, y^{(N)}\}$
- hypothesis

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + \ldots + w_D x_D = \sum_{j=0}^{D} w_j x_j = \mathbf{w}$$

- traing method
  - cost function

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)})^2$$

  - gradient descent

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w})$$

BGD vs SGD

- Batch gradient descent

  Repeat until convergence {

  $$w_j := w_j + \alpha \sum_{i=1}^{N} (y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)}))x_j^{(i)}$$

  }

- Stochastic gradient descent

  Loop {

  For $i = 1$ to N {

  $$w_j := w_j + \alpha(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)}))x_j^{(i)}$$

  }

  }

Probabilistic Interpretation

why choose Euclidean distance as cost function

1. inevitable error s.t. Gaussian Distribution
$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(\epsilon^{(i)})^2}{2\sigma^2})$$
2. y^i become a random variable
$$y^{(i)} = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$$
$$p(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2\sigma^2})$$
3. MLE(Max Likelihood Estimation)
$$\mathcal{L}(\mathbf{w}) = N\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\cdot\frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2$$
   - Maximizing $\mathcal{L}(\mathbf{w})$ gives the same answer as minimizing
$$\frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2$$

Generalized Linear Regression

Locally Weighted Linear Regression (LWR)

- LWR algorithm
  - Fit $\mathbf{w}$ to minimize $\sum_i \theta^{(i)}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2$
  - Output $\mathbf{w}^T\mathbf{x}$
  - where

    $$\theta^{(i)} = \exp(-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}\|^2}{2\tau^2})$$
- local vs global
- non-parametric

Linear Regression with Nonlinear Basis

- model nonlinear functions

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

  - where $\phi(x) = (1, x, x^2, \ldots, x^{M-1})$
  - still use least squares method to estimate

Geometry of Least Squares

- least-square vs orthogonal projection

Logistic Regression

Logistic Regression

- hypothesis

  $$P(y=1|\mathbf{x}) = f_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$$

- training method
  - MLE

    $$\mathbf{L}(\mathbf{w}) = \log L(\mathbf{w}) = \sum_{i=1}^{N} y^{(i)} \log f_{\mathbf{w}}(\mathbf{x}^{(i)}) + (1 -$$

  - gradient ascent

    $$w_j := w_j + \alpha(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)}))x_j^{(i)}$$

name of logistic regression

- odds of t
  $$\frac{t}{1-t}$$
- log odds / logit function of t
  $$\log \frac{t}{1-t}$$

Discriminative vs Generative Classification

Discriminative vs Generative Classification

- Discriminative
  - model $P(y \mid x)$
  - e.g. Logistic Regression, perception, SVM
- Generative
  - model $P(x \mid y)$ and $P(y)$
  - Bayesian Formula to get $P(y \mid x)$
  - e.g. GDA, NB
- Summary
  - only in the case of classification
  - diffent in process of modeling

Gaussian Discriminative Analysis

Assumption of $p(y)$

- Bernoulli
  $$Bern(x|\beta) = \beta^x (1-\beta)^{1-x}$$
- Binomial
  $$Bin(m|N,\beta) = \binom{N}{m} \beta^m (1-\beta)^{N-m}$$
- Multinomial
  $$Mult(m_1, \ldots, m_K|N,\beta) = \binom{N}{m_1 m_2 \ldots m_K} \prod_{k=1}^{K} \beta_k^{m_k}$$
  - m_k and β_k
  - $C_n^m = \frac{n!}{m!(n-m)!}$

Gaussian Discriminant Analysis(GDA)

- Assumption
  - $y \sim Bernoulli(\beta)$
  - $\mathbf{x} \mid y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$
  - $\mathbf{x} \mid y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$
- Parameters
  - $\beta, \mu_0, \mu_1, \Sigma$
- Log likelihood
- MLE

$$\beta = \frac{1}{N} \sum_{i=1}^{N} 1\{y^{(i)} = 1\}$$

$$\mu_k = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = k\} \mathbf{x}^{(i)}}{\sum_{i=1}^{N} 1\{y^{(i)} = k\}},\ k = \{0, 1\}$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \mu_{y^{(i)}})(\mathbf{x}^{(i)} - \mu_{y^{(i)}})^T$$

GDA and Logistic Regression

- GDA can be expressed in the form:
  $$P(y = 1 \mid \mathbf{x}; \beta, \mu_0, \mu 1, \Sigma) = \frac{1}{1 + exp(-\mathbf{w}^T \mathbf{x})}$$
- GDA: stronger assumption, more data effcient if assumption is correct
- Logistic Regression: weaker assumption, more robust

Naive Bayes

Email Spam Filter

- INPUT:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \text{a} \\ \text{aardwolf} \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{array}$$

  - $\mathbf{x} \in \{0, 1\}^D$, D is the size of vocabulary
  - a more general form is $x_j \sim multinomial$, which mean counts of word
- OUTPUT: classify emails to spam(y=1) or non-spam(y=0)

Naive Bayes

- Assumption:
  - $p(y) \sim Bernoulli(\phi)$
  - $p(\mathbf{x} \mid y) = p(x_1, \ldots, x_D \mid y) = \prod_{j=1}^{D} p(x_j \mid y)$
- Parameters
  - $p(x_j = 1 \mid y = 0)$
  - $p(x_j = 1 \mid y = 1)$
  - ϕ
- Log likelihood

- MLE

$$p(x_j = 1|y = 1) = \frac{\sum_{i=1}^{N} 1\{x_j^{(i)} = 1 \bigwedge y^{(i)} = 1\}}{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}}$$

$$p(x_j = 1|y = 0) = \frac{\sum_{i=1}^{N} 1\{x_j^{(i)} = 1 \bigwedge y^{(i)} = 0\}}{\sum_{i=1}^{N} 1\{y^{(i)} = 0\}}$$

$$p(y = 1) = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}}{N}$$

- Predict

$$p(y = 1|\mathbf{x}) = \frac{p(\mathbf{x}|y = 1)p(y = 1)}{p(\mathbf{x})}$$

$$= \frac{(\prod_{j=1}^{D} p(x_j|y = 1))p(}{(\prod_{j=1}^{D} p(x_j|y = 1))p(y = 1) + (\prod_{j=1}^{D}}$$

Laplace Smoothing

- Problem
  - no sample doesn't mean 0 probability
- Laplace smoothing

$$p(x=j) = \frac{\sum_{i=1}^{N} 1\{x^{(i)} = j\} + 1}{N+k},\ j = 1, \ldots, k$$

- NB with Laplace smoothing

$$p(x_j = 1 | y = 1) = \frac{\sum_{i=1}^{N} 1\{x_j^{(i)} = 1 \bigwedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{N} 1\{y^{(i)} = 1\} + 2}$$

$$p(x_j = 1 | y = 0) = \frac{\sum_{i=1}^{N} 1\{x_j^{(i)} = 1 \bigwedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{N} 1\{y^{(i)} = 0\} + 2}$$

Event Models for Text Classification

- A different way to represent emails: $\mathbf{x} = (x_1, \ldots, x_M)$ x_j denotes the j^{th} word in the email, taking values in $\{1, \ldots, |V|\}$ V is the vocabulary; M is the length of the email.
- lenth of $\mathbf{x}$ is not fixed

Naive Bayes vs. Logistic Regerssion

- # training set $\rightarrow$ infinite
  - model assumption correct
    - identical
  - model assumption incorrect
    - LR outperforms NB
- finite training set
  - convergence rate of parameter estimation
    - NB order logD   (D = # of attributes in X)
    - LR order D