Naive Bayes and Gaussian Bayes Classifier

Ladislav Rampasek
slides by Mengye Ren and others

February 22, 2016

Naive Bayes

Bayes' Rule:

$$p(t|x) = \frac{p(x|t)p(t)}{p(x)}$$

Naive Bayes Assumption:

$$p(x|t) = \prod_{j=1}^{D} p(x_j|t)$$

Likelihood function:

$$L(\theta) = p(x, t|\theta) = p(x|t, \theta)p(t|\theta)$$

Example: Spam Classification

- Each vocabulary is one feature dimension.
- We encode each email as a feature vector $x \in \{0, 1\}^{|V|}$
- $x_j = 1$ iff the vocabulary x_j appears in the email.

- We want to model the probability of any word x_j appearing in an email given the email is spam or not.
- Example: \$10,000, Toronto, Piazza, etc.

- Idea: Use Bernoulli distribution to model $p(x_j|t)$
- Example: $p(\text{“\$10,000”}|\text{spam}) = 0.3$

Bernoulli Naive Bayes

Assuming all data points $x^{(i)}$ are i.i.d. samples, and $p(x_j|t)$ follows a Bernoulli distribution with parameter μ_{jt}

$$p(x^{(i)}|t^{(i)}) = \prod_{j=1}^{D} \mu_{jt^{(i)}}^{x_j^{(i)}} (1 - \mu_{jt^{(i)}})^{(1-x_j^{(i)})}$$

$$p(t|x) \propto \prod_{i=1}^{N} p(t^{(i)}) p(x^{(i)}|t^{(i)}) = \prod_{i=1}^{N} p(t^{(i)}) \prod_{j=1}^{D} \mu_{jt^{(i)}}^{x_j^{(i)}} (1 - \mu_{jt^{(i)}})^{(1-x_j^{(i)})}$$

where $p(t) = \pi_t$. Parameters π_t, μ_{jt} can be learnt using maximum likelihood.

Derivation of maximum likelihood estimator (MLE)

$\theta = [\mu, \pi]$

$$\log L(\theta) = \log p(x, t|\theta)$$

$$= \sum_{i=1}^{N} \left(\log \pi_{t^{(i)}} + \sum_{j=1}^{D} x_j^{(i)} \log \mu_{jt^{(i)}} + (1 - x_j^{(i)}) \log(1 - \mu_{jt^{(i)}}) \right)$$

Want: $\arg\max_\theta \log L(\theta)$ subject to $\sum_k \pi_k = 1$

Derivation of maximum likelihood estimator (MLE)

Take derivative w.r.t. μ

$$\frac{\partial \log L(\theta)}{\partial \mu_{jk}} = 0 \Rightarrow \sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) \left(\frac{x_j^{(i)}}{\mu_{jk}} - \frac{1 - x_j^{(i)}}{1 - \mu_{jk}} \right) = 0$$

$$\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) \left[x_j^{(i)}(1 - \mu_{jk}) - \left(1 - x_j^{(i)}\right) \mu_{jk} \right] = 0$$

$$\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) \mu_{jk} = \sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) x_j^{(i)}$$

$$\mu_{jk} = \frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) x_j^{(i)}}{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}$$

Derivation of maximum likelihood estimator (MLE)

Use Lagrange multiplier to derive π

$$\frac{\partial L(\theta)}{\partial \pi_k} + \lambda \frac{\partial \sum_\kappa \pi_\kappa}{\partial \pi_k} = 0 \Rightarrow \lambda = -\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) \frac{1}{\pi_k}$$

$$\pi_k = -\frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}{\lambda}$$

Apply constraint: $\sum_k \pi_k = 1 \Rightarrow \lambda = -N$

$$\boxed{\pi_k = \frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}{N}}$$

Spam Classification Demo

Gaussian Bayes Classifier

Instead of assuming conditional independence of x_j, we model $p(x|t)$ as a Gaussian distribution and the dependence relation of x_j is encoded in the covariance matrix.

Multivariate Gaussian distribution:

$$f(x) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

μ: mean, Σ: covariance matrix, D: $\dim(x)$

Derivation of maximum likelihood estimator (MLE)

$$\theta = [\mu, \Sigma, \pi], Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

$$p(x|t) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

$$\log L(\theta) = \log p(x, t|\theta) = \log p(t|\theta) + \log p(x|t, \theta)$$

$$= \sum_{i=1}^{N} \log \pi_{t^{(i)}} - \log Z - \frac{1}{2}\left(x^{(i)} - \mu_{t^{(i)}}\right)^T \Sigma_{t^{(i)}}^{-1}\left(x^{(i)} - \mu_{t^{(i)}}\right)$$

Want: $\arg\max_\theta \log L(\theta)$ subject to $\sum_k \pi_k = 1$

Derivation of maximum likelihood estimator (MLE)

Take derivative w.r.t. μ

$$\frac{\partial \log L}{\partial \mu_k} = -\sum_{i=0}^{N} \mathbb{1}\left(t^{(i)} = k\right) \Sigma^{-1}(x^{(i)} - \mu_k) = 0$$

$$\boxed{\mu_k = \frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) x^{(i)}}{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}}$$

Derivation of maximum likelihood estimator (MLE)

Take derivative w.r.t. Σ^{-1} (not Σ)
Note:

$$\frac{\partial \det(A)}{\partial A} = \det(A) {A^{-1}}^{T}$$

$$\det(A)^{-1} = \det\left(A^{-1}\right)$$

$$\frac{\partial x^T A x}{\partial A} = x x^T$$

$$\Sigma^T = \Sigma$$

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=0}^{N} \mathbb{1}\left(t^{(i)} = k\right)\left[-\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2}(x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T\right] = 0$$

Derivation of maximum likelihood estimator (MLE)

$$Z_k = \sqrt{(2\pi)^D \det(\Sigma_k)}$$

$$\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} = \frac{1}{Z_k} \frac{\partial Z_k}{\partial \Sigma_k^{-1}} = (2\pi)^{-\frac{D}{2}} \det(\Sigma_k)^{-\frac{1}{2}} (2\pi)^{\frac{D}{2}} \frac{\partial \det\left(\Sigma_k^{-1}\right)^{-\frac{1}{2}}}{\partial \Sigma_k^{-1}}$$

$$= \det(\Sigma_k^{-1})^{\frac{1}{2}} \left(-\frac{1}{2}\right) \det\left(\Sigma_k^{-1}\right)^{-\frac{3}{2}} \det\left(\Sigma_k^{-1}\right) \Sigma_k^T = -\frac{1}{2}\Sigma_k$$

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=0}^{N} \mathbb{1}\left(t^{(i)} = k\right) \left[\frac{1}{2}\Sigma_k - \frac{1}{2}(x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T\right] = 0$$

$$\boxed{\Sigma_k = \frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right) \left(x^{(i)} - \mu_k\right) \left(x^{(i)} - \mu_k\right)^T}{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}}$$

Derivation of maximum likelihood estimator (MLE)

$$\pi_k = \frac{\sum_{i=1}^{N} \mathbb{1}\left(t^{(i)} = k\right)}{N}$$

(Same as Bernoulli)

Gaussian Bayes Classifier Demo

Gaussian Bayes Classifier

If we constrain Σ to be diagonal, then we can rewrite $p(x_j|t)$ as a product of $p(x_j|t)$

$$p(x|t) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma_t)}} \exp\left(-\frac{1}{2}(x_j - \mu_{jt})^T \Sigma_t^{-1} (x_k - \mu_{kt})\right)$$

$$= \prod_{j=1}^{D} \frac{1}{\sqrt{(2\pi)^D \Sigma_{t,jj}}} \exp\left(-\frac{1}{2\Sigma_{t,jj}} ||x_j - \mu_{jt}||_2^2\right) = \prod_{j=1}^{D} p(x_j|t)$$

Diagonal covariance matrix satisfies the naive Bayes assumption.

Gaussian Bayes Classifier

Case 1: The covariance matrix is shared among classes

$$p(x|t) = \mathcal{N}(x|\mu_t, \Sigma)$$

Case 2: Each class has its own covariance

$$p(x|t) = \mathcal{N}(x|\mu_t, \Sigma_t)$$

Gaussian Bayes Binary Classifier Decision Boundary

If the covariance is shared between classes,

$$p(x, t = 1) = p(x, t = 0)$$

$$\log \pi_1 - \frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1) = \log \pi_0 - \frac{1}{2}(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)$$

$$C + x^T \Sigma^{-1} x - 2\mu_1^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1 = x^T \Sigma^{-1} x - 2\mu_0^T \Sigma^{-1} x + \mu_0^T \Sigma^{-1} \mu_0$$

$$\left[2(\mu_0 - \mu_1)^T \Sigma^{-1}\right] x - (\mu_0 - \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) = C$$

$$\Rightarrow a^T x - b = 0$$

The decision boundary is a linear function (a hyperplane in general).

Relation to Logistic Regression

We can write the posterior distribution $p(t = 0|x)$ as

$$\frac{p(x, t = 0)}{p(x, t = 0) + p(x, t = 1)} = \frac{\pi_0 \mathcal{N}(x|\mu_0, \Sigma)}{\pi_0 \mathcal{N}(x|\mu_0, \Sigma) + \pi_1 \mathcal{N}(x|\mu_1, \Sigma)}$$

$$= \left\{1 + \frac{\pi_1}{\pi_0} \exp\left[-\frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1) + \frac{1}{2}(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)\right]\right\}^{-1}$$

$$= \left\{1 + \exp\left[\log \frac{\pi_1}{\pi_0} + (\mu_1 - \mu_0)^T \Sigma^{-1} x + \frac{1}{2}\left(\mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0\right)\right]\right\}^{-1}$$

$$= \frac{1}{1 + \exp(-w^T x - b)}$$

Gaussian Bayes Binary Classifier Decision Boundary

If the covariance is not shared between classes,

$$p(x, t = 1) = p(x, t = 0)$$

$$\log \pi_1 - \frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) = \log \pi_0 - \frac{1}{2}(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0)$$

$$x^T \left(\Sigma_1^{-1} - \Sigma_0^{-1}\right) x - 2 \left(\mu_1^T \Sigma_1^{-1} - \mu_0^T \Sigma_0^{-1}\right) x + \left(\mu_0^T \Sigma_0 \mu_0 - \mu_1^T \Sigma_1 \mu_1\right) = C$$

$$\Rightarrow x^T Q x - 2b^T x + c = 0$$

The decision boundary is a quadratic function. In 2-d case, it looks like an ellipse, or a parabola, or a hyperbola.

Thanks!