Naive Bayes and Gaussian models for classification

Michael Collins

February 22, 2012

Today's Lecture

- Probabilistic models:
  - Naive bayes
  - Gaussian models

Classification using Perceptron, SVMs

- Input: training examples $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$, where $\underline{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$

- Output: a parameter vector $\underline{\theta}$ that defines a function

$$f(\underline{x}; \underline{\theta})$$

  that maps points $\underline{x}$ to labels $y \in \{-1, +1\}$

Naive Bayes

- Input: a training sample $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$, where $\underline{x}_i \in \{-1, 1\}^d$ and $y_i \in \{-1, +1\}$
- Output: a parameter vector $\underline{\theta}$ that defines a distribution (i.e., a probability mass function (PMF))

$$p(\underline{x}, y; \underline{\theta})$$

- p is a well-defined PMF, i.e.,

$$\sum_{\underline{x}, y} p(\underline{x}, y; \underline{\theta}) = 1 \text{ and for all } \underline{x}, y, \;\; p(\underline{x}, y; \underline{\theta}) \geq 0$$

Using the Model for Classification

- The output of a naive bayes classifier is

$$
\begin{aligned}
f(\underline{x}) &= \arg\max_{y} p(y|\underline{x};\underline{\theta}) \\
&= \arg\max_{y} \frac{p(\underline{x},y;\underline{\theta})}{\sum_{y} p(\underline{x},y;\underline{\theta})} \\
&= \arg\max_{y} p(\underline{x},y;\underline{\theta})
\end{aligned}
$$

How do we Define $p(\underline{x}, y; \underline{\theta})$?

- There are 2^d possible values for $\underline{x}$, and 2 possible values for y, giving 2^{d+1} possibilities in total

The Naive Bayes Assumption

- Define random variables $Y, X_1, X_2, \ldots, X_d$. Each sample point is an input vector, with a label, defining values for Y and $X_1 \ldots X_d$.

- We'll make the following assumption:

$$
\begin{aligned}
& P(Y = y, X_1 = x_1, X_2 = x_2, \ldots X_d = x_d) \\
= \;& P(Y = y) P(X_1 = x_1, X_2 = x_2, \ldots X_d = x_d \,|\, Y = y) \\
= \;& P(Y = y) \prod_{j=1}^{d} P(X_j = x_j \,|\, Y = y)
\end{aligned}
$$

Note: the first step is exact (by the chain rule). The second step is an assumption, **the naive bayes assumption**

Parameters in a Naive Bayes Model

- The model form is as follows:

$$p(\underline{x}, y; \underline{\theta}) = q(y) \prod_{j=1}^{d} q_j(x_j|y)$$

- The parameter vector $\underline{\theta}$ contains the following parameters:
  - $q(y)$ for $y \in \{-1, +1\}$
  - $q_j(x|y)$ for $j = 1 \ldots d$, $y \in \{-1, +1\}$, and $x \in \{-1, 1\}$
- Constraints on these parameters:

$$q(+1) + q(-1) = 1$$

and for $y \in \{-1, +1\}$, for $j = 1 \ldots d$,

$$q_j(+1|y) + q_j(-1|y) = 1$$

Maximum Likelihood Estimates

- Given a training sample $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$, parameter estimates can be defined as

$$q(y) = \frac{\sum_{i=1}^{n} [[y_i = y]]}{n}$$

and

$$q_j(x|y) = \frac{\sum_{i=1}^{n} [[x_{i,j} = x \wedge y_i = y]]}{\sum_{i=1}^{n} [[y_i = y]]}$$

- Notation: $[[\pi]] = 1$ if the statement π is true, 0 otherwise. For example, $\sum_{i=1}^{n} [[y_i = y]]$ is the number of times $y_i = y$ in the training sample.

The Log-Likelihood Function, and ML Estimation

- The model form is as follows: $p(\underline{x}, y; \underline{\theta}) = q(y) \prod_{j=1}^{d} q_j(x_j|y)$. Our training data is $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$

- The **likelihood** of the training data under parameters $\underline{\theta}$ is

$$L'(\underline{\theta}) = \prod_{i=1}^{n} p(\underline{x}_i, y_i; \underline{\theta})$$

- The **log-likelihood** is

$$L(\underline{\theta}) = \log L'(\underline{\theta}) = \sum_{i=1}^{n} \log p(\underline{x}_i, y_i; \underline{\theta})$$

- The maximum-likelihood estimates are

$$\arg\max_{\underline{\theta}} L(\underline{\theta}) = \arg\max_{\underline{\theta}} L'(\underline{\theta})$$

Laplace Smoothing

- Define the smoothed estimates to be

$$q_j(x|y) = \frac{\alpha + \sum_{i=1}^{n} [[x_{i,j} = x \wedge y_i = y]]}{2\alpha + \sum_{i=1}^{n} [[y_i = y]]}$$

  where $\alpha > 0$ is some (typically small) constant, e.g., $\alpha = 1$

- In practice, this can give a big improvement over maximum-likelihood estimates.

Naive Bayes: Summary

- Input: a training sample $(\underline{x}_i, y_i)$ for $i = 1 \dots n$, where $\underline{x}_i \in \{0,1\}^d$ and $y_i \in \{-1, +1\}$

- Output: a parameter vector $\underline{\theta}$ that defines a distribution $p(\underline{x}, y; \underline{\theta})$. The vector $\underline{\theta}$ contains the $q(y)$ and $q_j(x|y)$ parameter estimates, which are estimated using maximum-likelihood or laplace smoothing.

- On a new test example, the output of the classifier is

$$\arg\max_y p(\underline{x}, y; \underline{\theta})$$

Naive Bayes: Generalizations

- Generalizations: it's simple to generalize naive bayes to the multi-class case where $y \in \{1, 2, \ldots, k\}$

- Generalizations: it's simple to generalize naive bayes to the case where attributes can take more than 2 values, i.e., for all $j = 1 \ldots d$, $x_j \in \{1, 2, \ldots, m_j\}$

More Notes on Naive Bayes

- One potential advantage: Simplicity, and efficiency
- A second potential advantage: The method is well defined in cases of *missing attributes*: training or test examples where some x_j values are not observed.
- An important thing to realise: naive bayes constructs a linear classifier

Today's Lecture

- Probabilistic models:
  - Naive bayes
  - Gaussian models

Data with Continuous-Valued Attributes

- For naive bayes, we assumed $\underline{x} \in \{-1, +1\}^d$
- What probabilistic models can we use when $\underline{x} \in \mathbb{R}^d$?

The Multivariate Normal Distribution

- The density (pdf) for a multivariate normal distribution where $\underline{x} \in \mathbb{R}^d$ is

$$N(\underline{x}; \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu})\right)$$

- $\underline{\mu} \in \mathbb{R}^d$ specifies the mean of the distribution
- Σ is a $d \times d$ matrix specifying the covariance of the distribution. Σ must be symmetric and positive semi-definite
- $|\Sigma|$ is the determinant of Σ

More about the Gaussian Distribution

- For a random variable $\underline{X}$ with pdf $N(\underline{x}; \underline{\mu}, \Sigma)$, the mean of the distribution is $\underline{\mu}$:
$$\mathbf{E}[\underline{X}] = \underline{\mu}$$
- The covariance of the random variable is Σ: for all i, j
$$\mathbf{E}[(X_i - \mu_i)(X_j - \mu_j)] = \Sigma_{i,j}$$

A Probabilistic Model Based on Normal Distributions

- Define

$$p(\underline{x}, y; \underline{\theta}) = q(y) N(\underline{x}; \underline{\mu}_y, \Sigma)$$

- The parameter vector $\underline{\theta}$ contains the following parameters:
  - $q(y)$ for $y \in \{-1, +1\}$
  - $\underline{\mu}_y \in \mathbb{R}^d$ for $y \in \{-1, +1\}$
  - Σ, a $d \times d$ positive semi-definite matrix

Applying the Model

- For a new test point $\underline{x}$, the output of the classifier is

$$\begin{aligned} f(\underline{x}) &= \arg\max_y p(y|\underline{x};\underline{\theta}) \\ &= \arg\max_y \frac{p(\underline{x},y;\underline{\theta})}{\sum_y p(\underline{x},y;\underline{\theta})} \\ &= \arg\max_y p(\underline{x},y;\underline{\theta}) \\ &= \arg\max_y q(y)N(\underline{x};\underline{\mu}_y,\Sigma) \end{aligned}$$

The Maximum-Likelihood Estimates

- Define our estimates as:

$$q(y) = \frac{\sum_{i=1}^{n} [[y_i = y]]}{n}$$

and

$$\underline{\mu}_y = \frac{\sum_{i:y_i=y} \underline{x}_i}{\sum_{i=1}^{n} [[y_i = y]]}$$

and

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\underline{x}_i - \underline{\mu}_{y_i})(\underline{x}_i - \underline{\mu}_{y_i})^T$$

Linear Decision Boundaries in the Model

- Because we've used a single parameter Σ, for the covariance of both distributions, it can be shown that the *decision boundary is again a linear separator*.
- Note: the decision boundary is the set of points $\underline{x}$ for which

$$p(\underline{x}, +1; \underline{\theta}) = p(\underline{x}, -1; \underline{\theta})$$