CSE 6363 - Machine Learning

Naive Bayes

Alex Dillhoff

University of Texas at Arlington

What is covered?

1. Definition of naive Bayes Classifier
2. Parameter Estimates via MLE
3. MNIST Example

Naive Bayes Classifier

We studied classification with a generative model when looking at Linear Discriminant Analysis.

For more complex data, the number of parameters required can be prohibitive.

Naive Bayes Classifier

For more complex data, the number of parameters required can be prohibitive.

Why?

Naive Bayes Classifier

For more complex data, the number of parameters required can be prohibitive.

Why?

The number of parameters increases the size of the covariance matrix, for example.

Naive Bayes Classifier

If we use SVD as the solver for LDA, the covariance matrix does not need to be computed.

Today, we will look at an alternative classifier that does not require such a large number of parameters.

The MNIST Dataset

To motivate naive Bayes classifiers, let's look at slightly more complex data.

The MNIST dataset was one of the standard benchmarks for computer vision classification algorithms for a long time.

It remains useful for educational purposes.

The MNIST Dataset

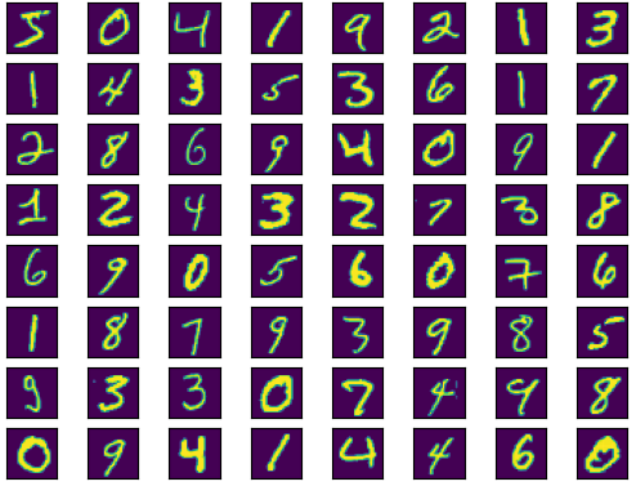

Figure: 64 samples from the MNIST dataset.

The MNIST Dataset

The dataset consists of 60,000 training images and 10,000 testing images of size 28×28.

These images depict handwritten digits.

The original goal of this dataset was to train systems that could correctly classify handwritten digits for the post office.

The MNIST Dataset

For simplicity in our model formulation, we will work with binary version of the images.

This implies that each data sample has 784 binary features.

The MNIST Dataset

We will use the naive Bayes classifier to make an image classification model which predicts the class of digit given a new image.

Each image will be represented by a vector $\mathbf{x} \in \mathbb{R}^{784}$.

The MNIST Dataset

Modeling $p(\mathbf{x}|C_k)$ with a multinomial distribution would require $10^{784} - 1$ parameters since there are 10 classes and 784 features.

With the naive assumption that the features are independent conditioned on the class, the number model parameters becomes 10×784.

Naive Bayes

A naive Bayes classifier makes the assumption that the features of the data are independent.

$$p(\mathbf{x}|C_k, \boldsymbol{\theta}) = \prod_{d=1}^{D} p(x_i|C_k, \theta_{dk}),$$

where $\boldsymbol{\theta}_{dk}$ are the parameters for the class conditional density for class k and feature d.

Naive Bayes

Using the MNIST dataset, $\boldsymbol{\theta}_k \in \mathbb{R}^{784}$.

The posterior distribution is then

$$p(C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(C_k|\boldsymbol{\pi}) \prod_{i=1}^{D} p(x_i|C_k, \theta_{dk})}{\sum_{k'} p(C_{k'}|\boldsymbol{\pi}) \prod_{i=1}^{D} p(x_i|C_{k'}, \theta_{dk'})}.$$

Naive Bayes

If we convert the input images to binary, the class conditional density $p(\mathbf{x}|C_k, \boldsymbol{\theta})$ takes on the Bernoulli pdf.

$$p(\mathbf{x}|C_k, \boldsymbol{\theta}) = \prod_{i=1}^{D} \text{Ber}(x_i|\theta_{dk})$$

Naive Bayes

The parameter θ_{dk} is the probability that the feature $x_i = 1$ given class C_k.

As seen before with MLE, this is very simple to estimate empirically.

Maximum Likelihood Estimation

Fitting a naive Bayes classifier is relatively simple using MLE.

The likelihood is given by

$$p(\mathbf{X}, \mathbf{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{M}(y_n|\boldsymbol{\pi}) \prod_{d=1}^{D} \prod_{k=1}^{K} p(x_{nd}|\theta_{dk})^{\mathbb{1}(y_n=k)}.$$

Maximum Likelihood Estimation

To derive the estimators, we first take the log of the likelihood:

$$\ln p(\mathbf{X}, \mathbf{y}|\boldsymbol{\theta}) = \left[\sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{1}(y_n = k) \ln \pi_k \right] + \sum_{k=1}^{K} \sum_{d=1}^{D} \left[\sum_{n:y_n=k} \ln p(x_{nd}|\theta_{dk}) \right].$$

Maximum Likelihood Estimation

We have a term for the the multinomial and terms for the class-feature parameters.

As with previous models that use a multinomial form, the parameter estimate for the first term is computed as

$$\hat{\pi}_k = \frac{N_k}{N}.$$

Maximum Likelihood Estimation

The features used in our data are binary, so the parameter estimate for each $\hat{\theta}_{dk}$ follows the Bernoulli distribution:

$$\hat{\theta}_{dk} = \frac{N_{dk}}{N_k}.$$

Making a Decision

Given parameters $\boldsymbol{\theta}$, how can we classify a given data sample?

Making a Decision

Given parameters $\boldsymbol{\theta}$, how can we classify a given data sample?

$$\arg\max_k p(y = k) \prod_i p(x_i | y = k)$$

Connection to Logistic Regression

Consider some data with discrete features having one of K states, then $x_{dk} = \mathbb{1}(x_d = k)$.

The class conditional density, in this case, follows a multinomial distribution:

$$p(y = c|\mathbf{x}, \theta) = \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{dck}^{x_{dk}}$$

Connection to Logistic Regression

We can see a connection between naive Bayes and logistic regression when we evaluate the posterior over classes:

$$
\begin{aligned}
p(y = c|\mathbf{x}, \theta) &= \frac{p(y)p(\mathbf{x}|y, \theta)}{p(\mathbf{x})} \\
&= \frac{\pi_c \prod_d \prod_k \theta_{dck}^{x_{dk}}}{\sum_{c'} \pi_{c'} \prod_d \prod_k \theta_{dc'k}^{x_{dk}}} \\
&= \frac{\exp[\log \pi_c + \sum_d \sum_k x_{dk} \log \theta_{dck}]}{\sum_{c'} \exp[\log \pi_{c'} + \sum_d \sum_k x_{dk} \log \theta_{dc'k}]}.
\end{aligned}
$$

Connection to Logistic Regression

This has the same form as the softmax function:

$$p(y = c|\mathbf{x}, \theta) = \frac{e^{\beta_c^T \mathbf{x} + \gamma_c}}{\sum_{c'=1}^{C} e^{\beta_{c'}^T \mathbf{x} + \gamma_{c'}}}$$

Demo

Demo: naive Bayes with MNIST