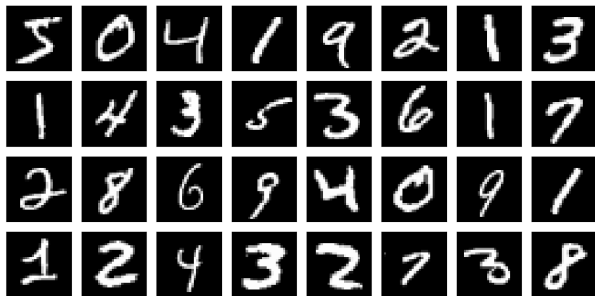


Classification

Nan Ye

School of Mathematics and Physics
The University of Queensland

MNIST



<http://yann.lecun.com/exdb/mnist/>

ImageNet

Jigsaw puzzle

A puzzle that requires you to reassemble a picture that has been mounted on a stiff base and cut into interlocking pieces




































1145
pictures

64.77%
Popularity
Percentile

Wordnet
IDs

- instrumentality, instrumentation (0)
- device (2760)
- implement (726)
- container (744)
- hardware, ironware (0)
- equipment (479)
 - automation (0)
 - radiotherapy equipment (0)
 - recorder, recording equipment (0)
 - naval equipment (11)
 - teaching aid (1)
 - sports equipment (99)
 - stock-in-trade (0)
 - electrical system (0)
 - game equipment (80)
 - pool table, billiard table (0)
 - paintball gun (0)
 - backboard, basketball (0)
 - crossbar (0)
 - net (1)
 - goal (3)
 - game (5)
 - puzzle (4)
 - crossword puzzle (0)
 - Chinese puzzle (0)
 - jigsaw puzzle (0)
 - tangram (0)
 - counter (2)
 - bowling equipment (6)
 - man, piece (12)
 - jack, jackstones (0)
 - horseshoe (0)

Treemap Visualization
Images of the Synset
Downloads


*Images of children synsets are not included. All images shown are thumbnails. Images may be subject to copyright.

Prev
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
...
[46](#)
[47](#)
Next

<http://www.image-net.org/>

Classification

- Classification involves determining the category of an input.
- A supervised classification method learns a classifier using a set of labelled examples.

$(\mathbb{5}, 5) \dots (\mathbb{4}, 4)$
 $(\mathbb{1}, 1) \dots (\mathbb{2}, 2)$  classifier

- We are often interested in the accuracy of a classifier, but there are a few other commonly used performance measures (e.g. precision, recall, F-measure).

Decision boundary

- When the inputs are numerical vectors in \mathbf{R}^d , it is convenient to think of a classifier as the boundary dividing the classes.
- When $d \leq 3$, we can visualize the classifier by explicitly drawing the decision boundary.
- When $d > 3$, we cannot plot the decision boundary, but people often embed the data in a 2/3 dimensional space, and visualize the decision boundary there.

Your turn

Consider the following classifier for 2D points

$$f(x_1, x_2) = \begin{cases} \text{red,} & \text{if } x_1 > 5, \\ \text{green,} & \text{if } x_1 \leq 5 \text{ and } x_2 > 1, \\ \text{blue,} & \text{if } x_1 \leq 5 \text{ and } x_2 \leq 1. \end{cases}$$

Draw the decision boundary for the classifier, and label each region with its class.

Classification Algorithms

- Nearest neighbor classifier
- Naive Bayes classifier
- Logistic regression
- Support vector machines

Remark on notations

- As in the case of regression, we use $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ to denote the training set.
- A bold font \mathbf{x} denotes a vector of d attributes, which may or may not be numerical.
- We often write \mathbf{x} as (x_1, \dots, x_d) , and \mathbf{x}_i as (x_{i1}, \dots, x_{id}) .
 - That is, the i -th component of \mathbf{x} is written as x_i , and the j -th component of \mathbf{x}_i is written as x_{ij} .

Nearest Neighbor Classifier

- As in the case of regression, we can also use the nearest neighbors to help us to determine the class of an input.
- For a given input \mathbf{x} , we predict the majority label of its k nearest neighbors in the training set, that is, a k NN classifier is given by

$$h_n(\mathbf{x}) = \text{majority}\{y_i : \mathbf{x}_i \in N_k(\mathbf{x})\},$$

where $N_k(\mathbf{x})$ consists of the k nearest examples of \mathbf{x} (wrt some distance measure).

Distance measures

- Various distance measures are used in practice.
- The Minkowski distance is given by

$$d_p(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_p = \left(\sum_i |x_i - x'_i|^p \right)^{1/p},$$

with d_1 being the Manhattan distance, and d_2 being the Euclidean distance.

- Caveat: k NN does not work well if the features are on different scales.

Case study: 1NN for MNIST

Input images



Nearest neighbor (Manhattan distance)



Nearest neighbor (Euclidean distance)



- Each image can be viewed as a feature vector of length 784, with each feature having integer values from 0 to 255.
- The input images are six images from the test set.
- The nearest neighbors are actually very similar to the input images, but there are small differences.

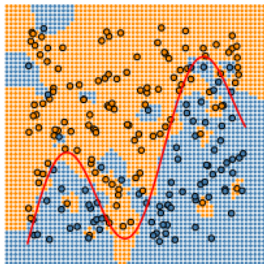
Advantage

- Under mild conditions, as $k \rightarrow \infty$ and $n/k \rightarrow \infty$, the k NN classifier will give you the best possible classification performance on average.

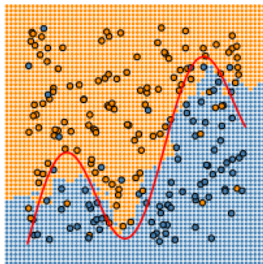
Limitations

- (Curse of dimensionality) The number of samples required for accurate approximation is exponential in the dimension.
- Finding the nearest neighbors is computationally expensive.

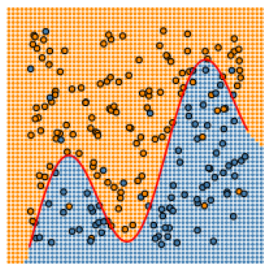
1-NN classifier



10-NN classifier



Bayes optimal classifier



Naive Bayes Classifier (NB)

- Consider an input space $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_d$, where each \mathcal{X}_i is a finite set.
- NB attempts to model the joint distribution of the input X and the output Y assuming that the features are conditionally independent of each other given the output.
- Specifically, NB assumes a joint distribution $p(X, Y)$ satisfying

$$p(x_1, \dots, x_d | y) = p(x_1 | y) \dots p(x_d | y).$$

- Usually, this does not hold, but it can be a good approximation for classification purpose.

Learning (MLE)

- We train a NB model by maximizing the likelihood

$$\arg \max_{\rho} \prod_i p(\mathbf{x}_i, y_i).$$

- The maximum likelihood model is given by

$$\begin{aligned}\hat{p}(y) &= n_y/n, \\ \hat{p}(x_j | y) &= n_{y,x_j}/n_y,\end{aligned}$$

where n_y is the number of times class y appears in the training set, and n_{y,x_j} is the number of times attribute $1 \leq j \leq d$ takes value $x_j \in \mathcal{X}_j$ when the class label is y .

Prediction

- Given an NB model p , an example $\mathbf{x} = (x_1, \dots, x_d)$ is classified as

$$y = \arg \max_{y' \in \mathcal{Y}} p(y' | \mathbf{x}).$$

- This is equivalent to

$$y = \arg \max_{y' \in \mathcal{Y}} p(y', \mathbf{x}) = \arg \max_{y' \in \mathcal{Y}} p(y')p(x_1 | y') \dots p(x_d | y'),$$

by the independence assumption.

Limitations

- Independence assumption unlikely to be satisfied.
- The counts n_y may be 0, making the estimates undefined.
- The counts may be very small, leading to unstable estimates.

Laplace correction

- To deal with small counts, we add pseudo-counts to them

$$\hat{p}(y) = (n_y + c_0) / \sum_{y' \in \mathcal{Y}} (n_{y'} + c_0),$$

$$\hat{p}(x_j | y) = (n_{y,x_j} + c_1) / \sum_{x'_j \in \mathcal{X}_j} (n_{y,x'_j} + c_1),$$

where $c_0 > 0$ and $c_1 > 0$ are user-chosen constants.

- Laplace correction makes NB more stable, but still relies on strong independence assumption.

Extension to real-valued features

- We can assume that $p(x_j | y)$ is a Gaussian distribution.
- It suffices to compute the mean and variance from data.

Logistic Regression (LR)

Model

- $\mathcal{X} = \mathbf{R}^d$.
- Logistic regression estimates conditional distributions of the form

$$p(y | \mathbf{x}, \theta) = \exp(\mathbf{x}^\top \theta_y) / \sum_{y' \in \mathcal{Y}} \exp(\mathbf{x}^\top \theta_{y'}),$$

where $\theta_y = (\theta_{y1}, \dots, \theta_{yd}) \in \mathbf{R}^d$, and θ is the concatenation of θ_y 's.

Logistic Regression (LR)

Model

- $\mathcal{X} = \mathbf{R}^d$.
- Logistic regression estimates conditional distributions of the form

$$p(y | \mathbf{x}, \theta) = \exp(\mathbf{x}^\top \theta_y) / \sum_{y' \in \mathcal{Y}} \exp(\mathbf{x}^\top \theta_{y'}),$$

where $\theta_y = (\theta_{y1}, \dots, \theta_{yd}) \in \mathbf{R}^d$, and θ is the concatenation of θ_y 's.

Prediction

An example \mathbf{x} is classified as

$$y = \arg \max_{y' \in \mathcal{Y}} p(y' | \mathbf{x}, \theta).$$

Learning

- Training is often done by maximizing regularized log-likelihood

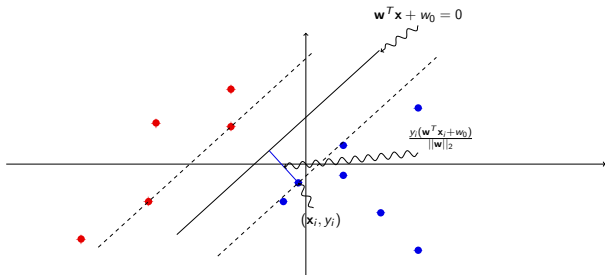
$$L(\theta) = \log \prod_{i=1}^n p(y_i | \mathbf{x}_i, \theta) - \lambda \|\theta\|_2^2.$$

That is, the parameter estimate is

$$\theta_n = \arg \max_{\theta} L(\theta).$$

- $L(\theta)$ is a concave function, and can be optimized using standard numerical methods (such as L-BFGS).

Support Vector Machines (SVMs, optional)



Geometric intuition

- If the data is separable (that is, the two classes lie on the two sides of some hyperplane), then usually there are many separating hyperplanes.
- The intuition of SVM is to find a separating hyperplane with maximal margin (i.e., the minimum distance from the points to it).

Algebraic formulation

- Finding out the maximum margin hyperplane can be directly translated to the following optimization problem

$$\begin{aligned} & \max_{M, \mathbf{w}, w_0} M \\ \text{s.t.} \quad & \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{\|\mathbf{w}\|_2} \geq M, \quad i = 1, \dots, n. \end{aligned}$$

- This can be shown to be equivalent to

$$\begin{aligned} & \min_{\mathbf{w}, w_0} \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad i = 1, \dots, n. \end{aligned}$$

Soft-margin SVMs

- In general, we need to deal with non-separable data (that is, the two classes do not lie on the two sides of any hyperplane).
- The SVM formulation that we have seen is called hard-margin SVM.
- We can adapt hard-margin SVM to the non-separable case, by penalizing examples which do not lie on the side that they belong to.

Algebraic formulation

The soft-margin SVM for non-separable data can be formulated as

$$\min_{\mathbf{w}, w_0, \xi_1, \dots, \xi_n} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_i \xi_i$$

$$\text{subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 - \xi_i, i = 1, \dots, n,$$

$$\xi_i \geq 0, i = 1, \dots, n.$$

where

- $C > 0$ is a user chosen constant.
- Introducing ξ_i allows (\mathbf{x}_i, y_i) to be misclassified with a penalty of $C\xi_i$ in the original objective function $\frac{1}{2} \|\mathbf{w}\|_2^2$.

SVM as minimizing regularized hinge loss

Soft-margin SVMs can be equivalently written as

$$\min_{\mathbf{w}, w_0} \frac{1}{2C} \|\mathbf{w}\|_2^2 + \sum_i \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)),$$

where $\max(0, 1 - y(\mathbf{w}^\top \mathbf{x} + w_0))$ is the hinge loss

$$L_{\text{hinge}}((\mathbf{x}, y), h) = \max(0, 1 - yh(\mathbf{x}))$$

of the classifier $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$, and upper bounds the 0/1 loss

$$L_{0/1}((\mathbf{x}, y), h) = I(y \neq \text{sgn}(h(\mathbf{x})))$$

Your Turn

Which of the following statement is correct? (Multiple choice)

- (a) kNN, naive Bayes, and logistic regression can all be viewed as probabilistic classifiers.
- (b) The scale of features has no influence on kNN's performance.
- (c) Naive Bayes can only be applied to problems with categorical features.
- (d) A logistic regression can be trained by maximizing the log-likelihood.

What You Need to Know...

- Decision boundary
- Probabilistic classifiers: NN, NB and LR.
- SVMs (optional)