Lecture 2
Supervised Learning Overview

Zhihua Jiang

Content

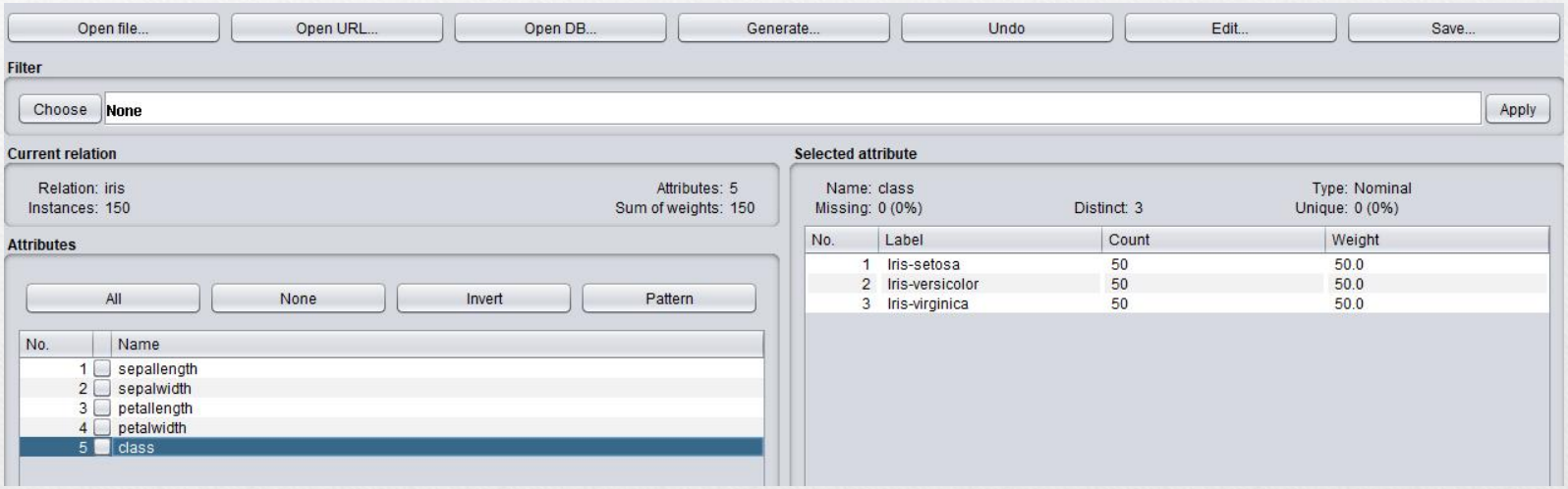
Basic terminology (Recap)

- Supervised learning: learn with labeled training data
- Unsupervised learning: learn with unlabeled training data
- Semi–supervised: a small amount of labeled data with a large amount of unlabeled data.

Ex. of Labeled data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Ex. of Labeled data



Ex. of Labeled data

Analysis Result: 28.747 seconds cost. 47 pdf ruins. 105\64 pdf have result. All quotations number:86 , positive:17 , objective:69 , negative:0 .

Pola...	Quotation	Citation
o	In this paper, weka [7] machine learning tool is used for performing evaluation using clustering and classification algorithm.	A Comparative Stud...
p	Chapter 4 shows some analytical results using popular open source data mining tool WEKA [5][6] to examine the efficiency of our pro...	A Fine Grained Tec...
p	Decision trees are a classic way to represent information from a machine learning algorithm, and offer a fast and powerful way to exp...	A Fine Grained Tec...
o	The framework implementation is based on the Kieker [11] framework for monitoring and dynamic analysis of software systems, and ...	A framework for sys...
o	The machine learning library used in our evaluation is Weka [1].	A framework for sys...
p	One of these packages is WEKA [17], which includes many different machine learning algorithms In general, the use of machine learni...	A Method for Gener...
p	Weka[17] is a free open source machine learning toolbox that is widely used among the developers in this área.	A Method for Gener...

Ex. of Labeled data

ImageNet

From Wikipedia, the free encyclopedia

The **ImageNet** project is a large visual database designed for use in visual object recognition software research. Over 14 million[1][2] URLs of images have been hand-annotated by ImageNet to indicate what objects are pictured; in at least one million of the images, bounding boxes are also provided.[3] ImageNet contains over 20 thousand categories;[2] a typical category, such as "balloon" or "strawberry", contains several hundred images.[4] The database of annotations of third-party image URL's is freely available directly from ImageNet; however, the actual images are not owned by ImageNet.[5] Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes. The ImageNet Challenge uses a "trimmed" list of one thousand non-overlapping classes.[6]

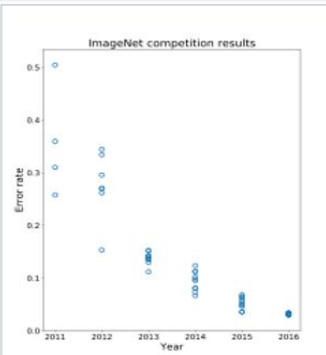
A dramatic 2012 breakthrough in solving the ImageNet Challenge is widely considered to be the beginning of the deep learning revolution of the 2010s: "Suddenly people started to pay attention, not just within the AI community but across the technology industry as a whole."[4][7]

History [edit]

The database was presented for the first time as a poster at the 2009 Conference on Computer Vision and Pattern Recognition (CVPR) in Florida by researchers from the Computer Science department at Princeton University.[8][9]

Dataset [edit]

ImageNet crowdsources its annotation process. Image-level annotations indicate the presence or absence of an object class in an image, such as "there are tigers in this image" or "there are no tigers in this image". Object-level annotations provide a bounding box around the (visible part of the) indicated object. ImageNet uses a variant of the broad WordNet schema to categorize objects, augmented with 120 categories of dog breeds to showcase fine-grained classification.[6] One downside of WordNet use is the categories may be more "elevated" than would be optimal for ImageNet: "Most people are more interested in Lady Gaga or the iPod Mini than in this rare kind of diplodocus." In 2012 ImageNet was the world's largest academic user of Mechanical Turk. The average worker identified 50 images per minute.[2]



ImageNet Challenge [edit]

imagenet_百度搜索 | ImageNet_百度百科 | ImageNet - Wikipedia | ImageNet Tree View

不安全 | www.image-net.org/synset?wnid=n04385799#

IMAGENET

SEARCH

14,197,122 images, 21841 synsets indexed

Home | Explore | About | Download

Not logged in. Login | Signup

Rock, stone

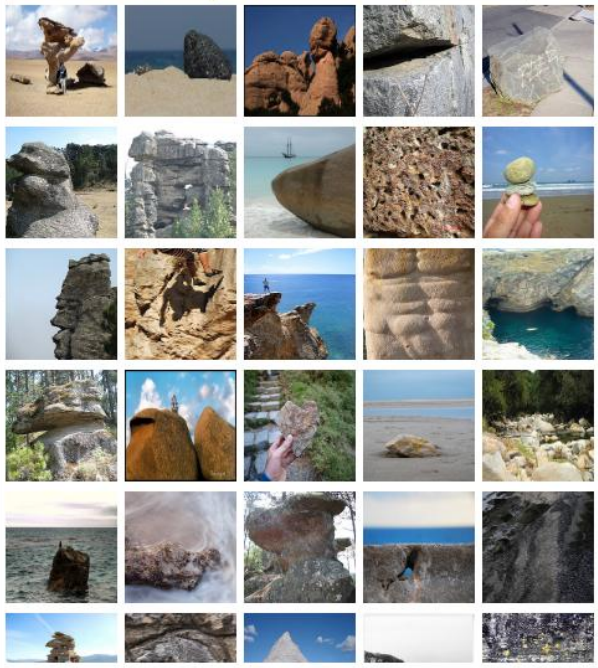
A lump or mass of hard consolidated mineral matter; "he threw a rock at me"

1275 pictures | 98.95% Popularity Percentile | Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
  - plant, flora, plant life (4486)
  - geological formation, formation (175)
  - natural object (1112)
    - rock, stone (30)
      - outcrop, outcropping, rock outcrop (2)
      - whinstone, whin (0)
      - xenolith (0)
      - tor (0)
      - pebble (0)
      - chondrite (0)
      - stepping stone (0)
      - petrifaction (0)
      - sill (0)
      - wall rock (0)
      - boulder, bowlder (3)
      - bedrock (0)
      - achondrite (0)
      - crystal, crystallization (1)
      - calculus, concretion (7)
      - clastic rock (0)
      - intrusion (0)
    - asterism (0)
    - carpet (0)
    - black body, blackbody, full radiator (0)
    - radiator (1)
    - consolidation (0)
    - mechanism (12)
    - body, organic structure, physical structure (12)
    - nest (7)

Treemap Visualization | Images of the Synset | Downloads



ImageNet Object Localization Challenge | Kaggle

Competition Overview

The validation and test data will consist of 150,000 photographs, collected from Flickr and other search engines, hand labeled with the presence or absence of 1000 object categories. The 1000 object categories contain both internal nodes and leaf nodes of ImageNet, but do not overlap with each other.

A random subset of 50,000 of the images with labels will be released as the training set along with a list of the 1000 categories. The remaining images will be used as the test set.

The validation and test data for this competition are not contained in the ImageNet training data.

Evaluation

In this competition, the error for each image is defined as

$$e = min_i(min_j(max(d_{ij}, f_{ij})))$$

where

$d = 0$ if the labels of the two boxes are the same, and $d = 1$ otherwise;

$f = 0$ if the overlap of the two boxes >= 50%, and $f = 1$ otherwise;

i is the predicted labels/bounding boxes, and j is the ground truth labels/bounding boxes.

For example, let's assume for a given image, there are 2 boxes as ground truth (`g_0` , `g_1`), and you predict 3 boxes in your prediction (`p_0` , `p_1` , `p_2`). We iterate through your prediction boxes, and see if they can find a "match" with any of the ground truth boxes. If there's a match, then the min error for this image is 0, otherwise, the min error is 1.

A match is defined as

1. the prediction box has a class label that is the same as the ground truth box, and

2. the prediction bounding box has over 50% match in the area with the ground truth bounding box.

Note the min error is either 0 or 1 for each image.

The total error is then computed as the average of all min errors of all the images in the test dataset.

Submission File

For each image in the test dataset, you will predict a list of label and bounding boxes.

It contains two columns:

- `ImageId` : the id of the test image, for example `ILSVRC2012_test_00000001`
- `PredictionString` : the prediction string should be a space delimited of 5 integers. For example, `1000 240 170 260 240` means it's label 1000, with a bounding box of coordinates (x_min, y_min, x_max, y_max). We accept up to 5 predictions. For example, if you submit `862 42 24 170 186 862 292 28 430 198 862 168 24 292 190 862 299 238 443 374 862 160 195 294 357 862 3 214 135 356` which contains 6 bounding boxes, we will only take the first 5 into consideration.

```
ImageId,PredictionString
ILSVRC2012_test_00000001,1000 240 170 260 240
ILSVRC2012_test_00000002,825 240 170 260 240 829 152 331 246 415
ILSVRC2012_test_00000003,862 42 24 170 186 862 292 28 430 198 862 168 24 292 190 862 299
238 443 374 862 160 195 294 357
```

ImageNet: A Large-Scale Hierarchical Image Database

Figure 11: Samples of detected bounding boxes around different objects.

2.3 Two Simple Approaches to Prediction

- Least squares (最小二乘法)
  - Linear model
  - Minimize the residual sum of squares
  - Small amount of training data
- Nearest neighbors (最近邻法)
  - Neighborhood Measurement
  - Majority voting or average
  - Model–free + precise; but lack of interpretability

Least squares

- Linear model: given a vector of inputs $X^T = (X_1, X_2, ..., X_p)$

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j.$$

bias

weight

- Residual sum of squares (平方残差和):

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2.$$

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta),$$

Least squares

where $\mathbf{X}$ is an $N \times p$ matrix with each row an input vector, and $\mathbf{y}$ is an N-vector of the outputs in the training set. Differentiating w.r.t. β we get the *normal equations*

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0. \tag{2.5}$$

If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution is given by

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.6}$$

and the fitted value at the ith input x_i is $\hat{y}_i = \hat{y}(x_i) = x_i^T\hat{\beta}$. At an arbitrary input x_0 the prediction is $\hat{y}(x_0) = x_0^T\hat{\beta}$. The entire fitted surface is characterized by the p parameters $\hat{\beta}$. Intuitively, it seems that we do not need a very large data set to fit such a model.

$$\hat{G} = \begin{cases} \texttt{ORANGE} & \text{if } \hat{Y} > 0.5, \\ \texttt{BLUE} & \text{if } \hat{Y} \leq 0.5. \end{cases}$$

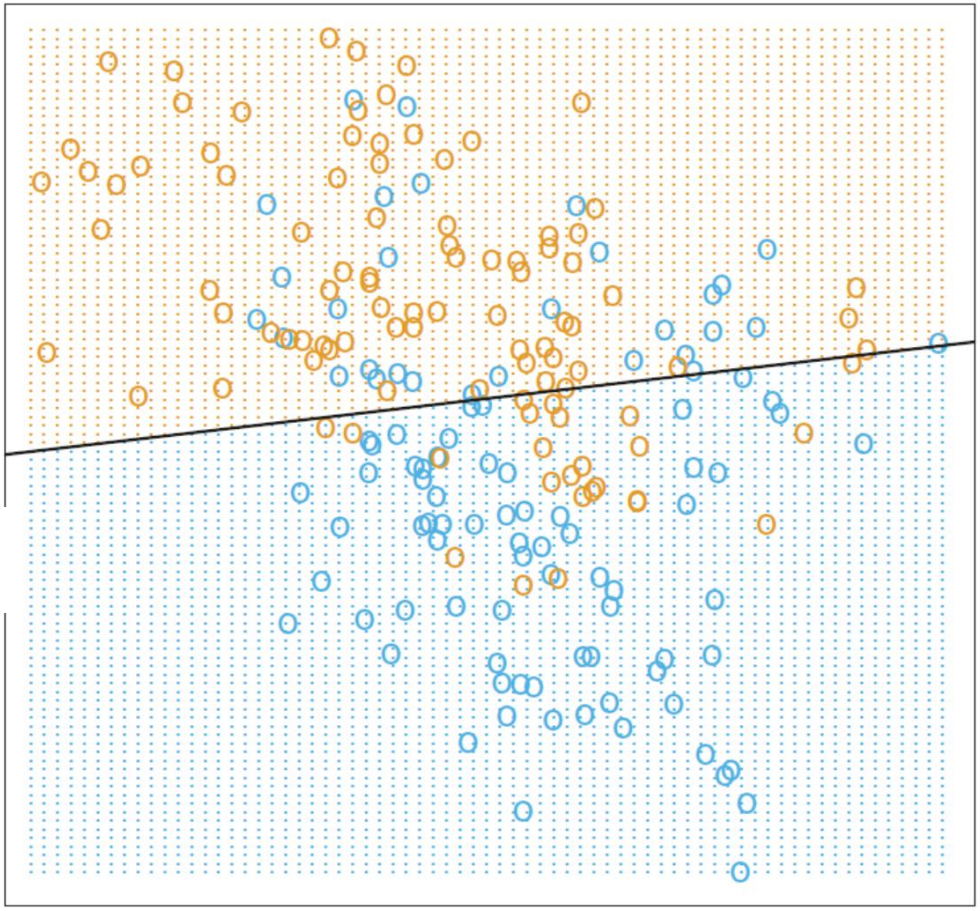


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T \hat{\beta} = 0.5$*. The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE*.*

Nearest neighbors

- $N_k(x)$ is the neighborhood of x defined by the k closest points x_i

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- Common closeness measurement
  - Euclidean distance

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

15-Nearest Neighbor Classifier

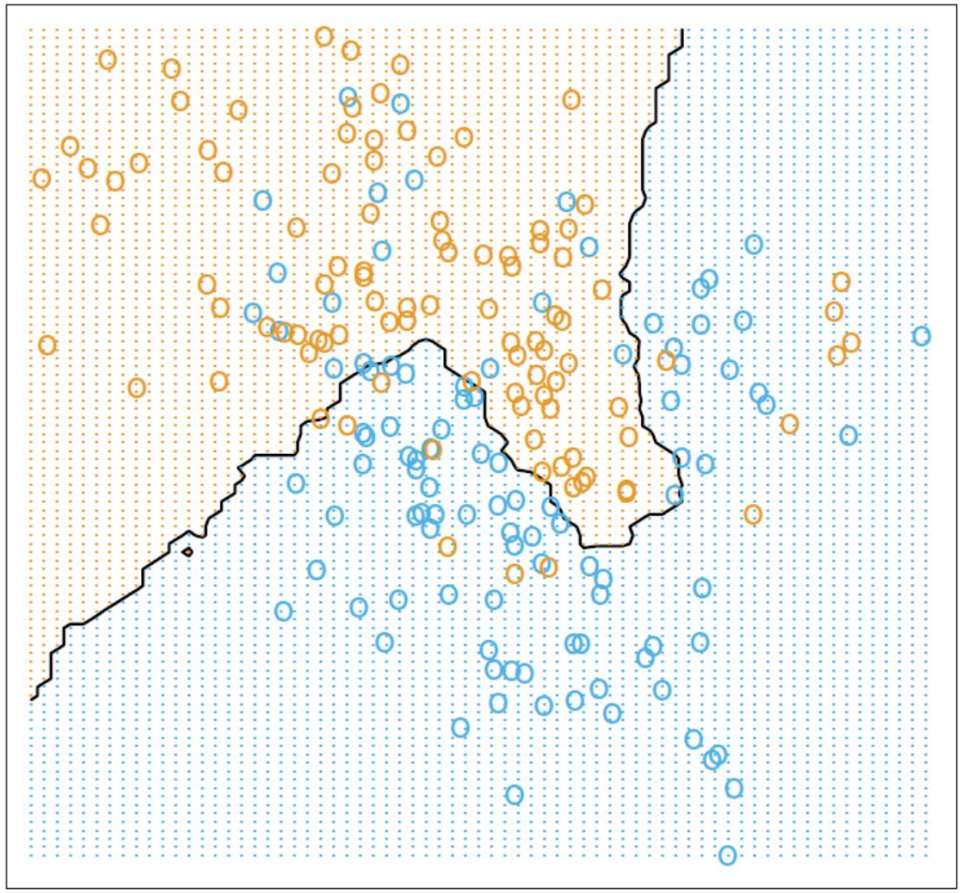


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

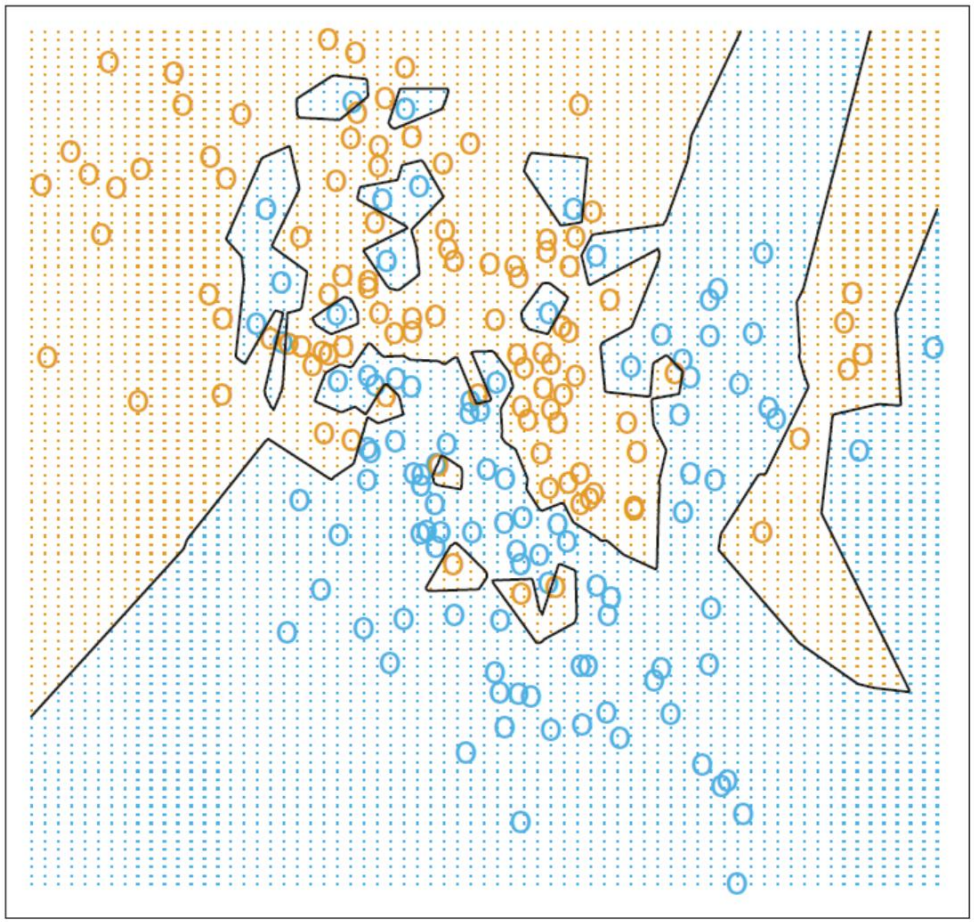


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*

2.4 Statistical Decision Theory

- Loss function (损失函数): penalize errors in prediction
  - Squared loss function for regression (f is continuous)
  - Zero–one loss function for classification (f is discrete)
- Expected prediction error (期望预测误差, EPE): expectation of loss function
- Optimal prediction: to minimize the EPE

Squared loss function

- Squared loss function:

$$L(Y, f(X)) = (Y - f(X))^2$$

$X \in \mathbb{R}^p$, $Y \in \mathbb{R}$, with joint distribution $\Pr(X, Y)$

$$\text{EPE}(f) = E(Y - f(X))^2 = \int [y - f(x)]^2 \Pr(dx, dy)$$

Zero-one loss function

- All misclassifications are charged a single unit.
- An estimate $\hat{G}$ will assume values in $\mathcal{G}$, $K = \mathrm{card}(\mathcal{G})$

$$\mathrm{EPE} = \mathrm{E}[L(G, \hat{G}(X))]$$

$$\mathrm{EPE} = \mathrm{E}_X \sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)]\Pr(\mathcal{G}_k|X)$$

Zero-one loss function

- Minimize EPE pointwise

$$\hat{G}(x) = \text{argmin}_{g \in \mathcal{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k | X = x).$$

- Simplify:

$$\hat{G}(x) = \text{argmin}_{g \in \mathcal{G}} [1 - \Pr(g | X = x)]$$

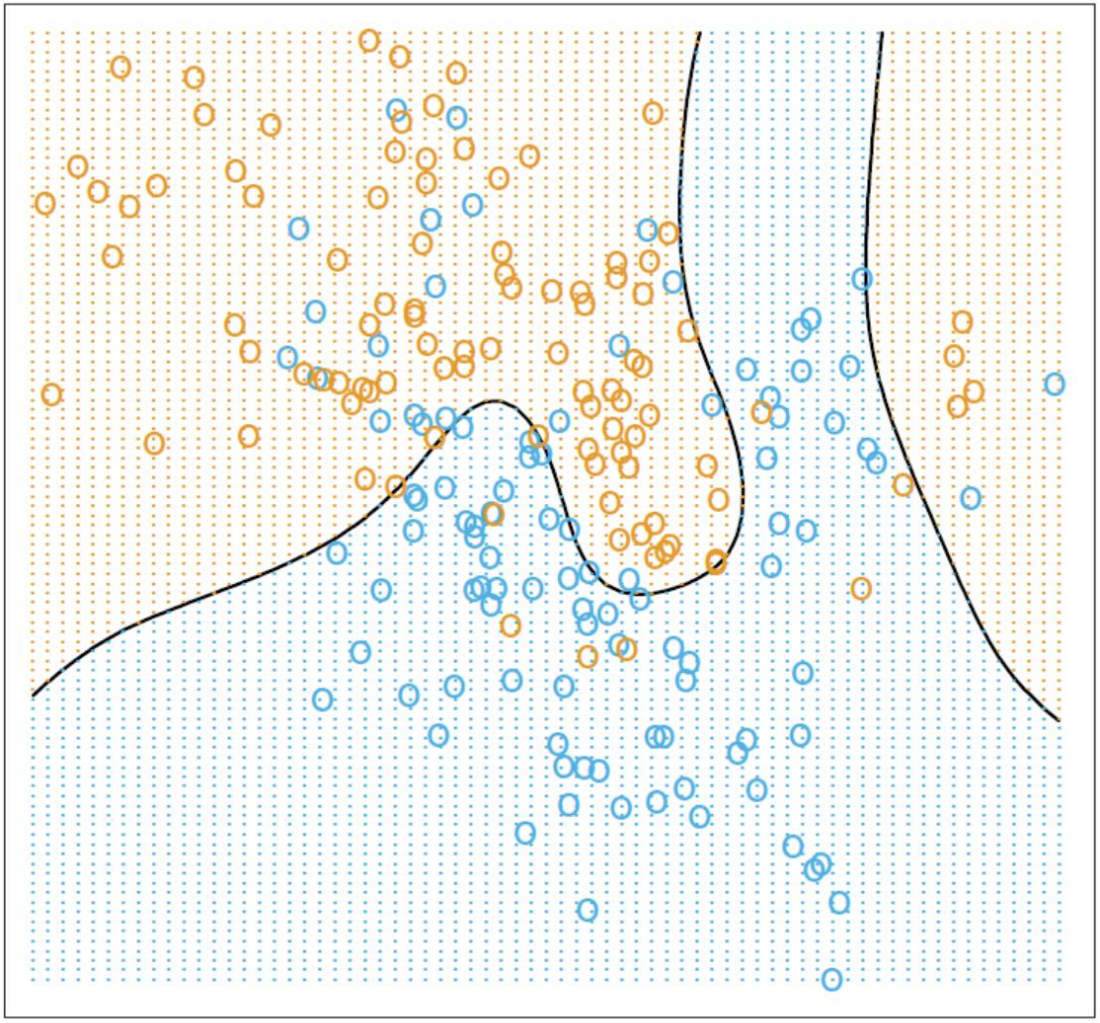
Zero-one loss function

- **Bayes classifier** (贝叶斯分类器)
  - classify to the most probable class, using the conditional (discrete) distribution

$$\hat{G}(x) = \text{argmin}_{g \in \mathcal{G}} [1 - \Pr(g|X = x)]$$

$$\hat{G}(x) = \arg\max_g \Pr(g \mid X = x)$$

Bayes Optimal Classifier



2.5 Local Methods in High Dimensions

- **Curse of dimensionality (维度灾难)**
  - When the dimensionality increases, the volume of the space increases so fast that the available data becomes sparse
  - The amount of data needed to support the result often grows exponentially with the dimensionality
  - Difficult for sampling; local methods inefficient

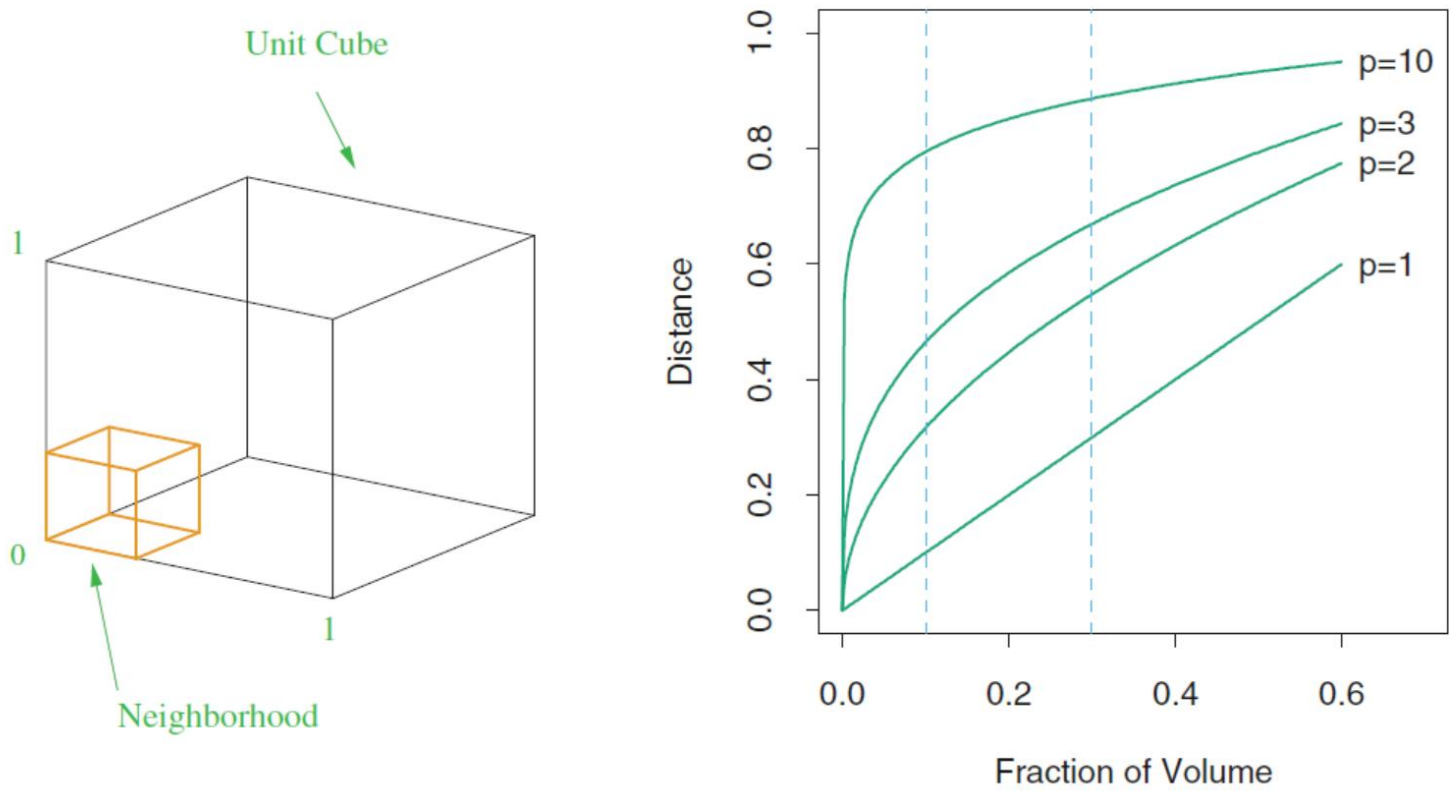


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

2.5 Local Methods in High Dimensions

- **Bias–variance decomposition**

MSE = variance (方差) + squared bias (平方偏差)

$$\begin{aligned}\mathrm{MSE}(x_0) &= \mathrm{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\ &= \mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2 + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\ &= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0).\end{aligned}$$

Variance vs. bias

- Variance: changes in learning performance due to changes in the training set, i.e., impact of data perturbation

$$\mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2$$

- Bias: deviation between the expected and real results of the learning algorithm, i.e., fitting ability of learning algorithm

$$\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)$$

$$
\begin{aligned}
\mathrm{MSE}(x_0) &= \mathrm{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\
&= \mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2 + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0).
\end{aligned}
$$

$$
\begin{aligned}
& E_T[\hat{y}_0 - E_T(\hat{y}_0)]^2 + [E_T(\hat{y}_0) - f(x_0)]^2 \\
&= E_T[\hat{y}_0{}^2 - 2\hat{y}_0 E_T(\hat{y}_0) + (E_T(\hat{y}_0))^2] + E_T(\hat{y}_0)^2 - 2E_T(\hat{y}_0) f(x_0) + f(x_0)^2 \\
&= E_T(\hat{y}_0{}^2) - 2E_T(\hat{y}_0) E_T(\hat{y}_0) + E_T(\hat{y}_0)^2 + E_T(\hat{y}_0)^2 - 2E_T(\hat{y}_0) f(x_0) + f(x_0)^2 \\
&= E_T(\hat{y}_0{}^2) - 2E_T(\hat{y}_0) f(x_0) + f(x_0)^2 \\
&= E_T(\hat{y}_0{}^2) - 2E_T(\hat{y}_0 f(x_0)) + E_T(f(x_0)^2) \\
&= E_T[(\hat{y}_0{}^2) - 2\hat{y}_0 f(x_0) + f(x_0)^2] \\
&= E_T[f(x_0) - \hat{y}_0]^2
\end{aligned}
$$