Machine learning

Probabilistic Generative Classifiers

Hamid Beigy

Sharif University of Technology

April 3, 2023

Table of contents

1. Introduction

2. Bayes decision theory

3. Supervised learning of the Bayesian classifiers

4. Naive Bayes classifier

5. Reading

Introduction

Introduction

1. In classification, the goal is to find a mapping from inputs X to outputs t given a labeled set of input-output pairs
$$S = \{(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N)\}.$$
S is called training set.
2. In the simplest setting, each training input x is a $D-$dimensional vector of numbers.
3. Each component of x is called feature, attribute, or variable and x is called feature vector.
4. The goal is to find a mapping from inputs X to outputs t, where $t \in \{1, 2, \dots, C\}$ with C being the number of classes.
5. When $C = 2$, the problem is called binary classification. In this case, we often assume that $t \in \{-1, +1\}$ or $t \in \{0, 1\}$.
6. When $C > 2$, the problem is called multi-class classification.

Introduction (cont.)

1. Bayes theorem

$$\begin{aligned} p(C_k|X) &= \frac{P(X|C_k)P(C_k)}{P(X)} \\ &= \frac{P(X|C_k)P(C_k)}{\sum_{C_k} p(X|C_k)p(C_k)} \end{aligned}$$

   - $p(C_k)$ is called prior of C_k.
   - $p(X|C_k)$ is called likelihood of data .
   - $p(C_k|X)$ is called posterior probability.

2. Since $p(X)$ is the same for all classes, we can write as

$$p(C_k|X) \propto P(X|C_k)P(C_k)$$

3. Approaches for building a classifier.
   - Generative approach: This approach first creates a joint model of the form of $p(x, C_k)$ and then to condition on x, deriving $p(C_k|x)$.
   - Discriminative approach: This approach creates a model of the form of $p(C_k|x)$ directly.

Bayes decision theory

Bayes decision theory

1. Given a classification task of M classes, $C_1, C_2, \ldots, C_M$, and an input vector x, we can form M conditional probabilities

$$p(C_k|x) \qquad \forall k = 1, 2, \ldots, M$$

2. Without loss of generality, consider two class classification problem. From the Bayes theorem, we have

$$p(C_k|x) \quad \propto \quad P(x|C_k)P(C_k)$$

3. The Bayes classification rule is

$$\begin{array}{lll} \text{if } p(C_1|x) > p(C_2|x) & \text{then} & x \text{ is classified to } C_1 \\ \text{if } p(C_1|x) < p(C_2|x) & \text{then} & x \text{ is classified to } C_2 \\ \text{if } p(C_1|x) = p(C_2|x) & \text{then} & x \text{ is classified to either } C_1 \text{ or } C_2 \end{array}$$

Since $p(x)$ is same for all classes, then it can be removed. Hence

$$p(x|C_1)p(C_1) \lesseqgtr p(x|C_2)p(C_2)$$

4. If $p(C_1) = p(C_2) = \frac{1}{2}$, then we have

$$p(x|C_1) \lesseqgtr p(x|C_2)$$

Simple Example (Mitchell's book)

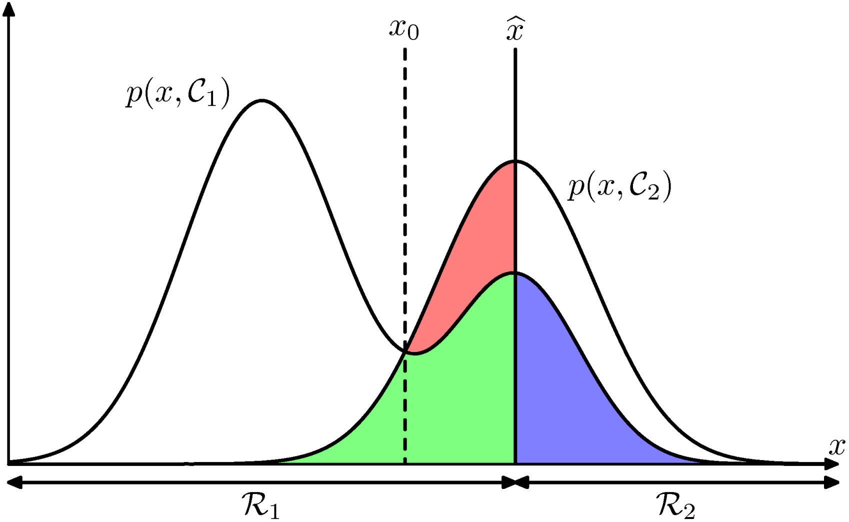
1. Let $C = \{C_1, C_2\}$, with
   - C_1 = the patient has a certain disease,
   - C_2 = the patient has not a certain disease,
2. **Prior :** Over the entire population
   - $P(C_1) = 0.008$,
   - $P(C_2) = 0.992$,
3. **Likelihoods (lab test result $x \in \{0, 1\}$)**
   - $P(x = 1|C_1) = 0.98$ (true positive),
   - $P(x = 0|C_1) = 0.02$ (false negative),
   - $P(x = 1|C_2) = 0.03$ (false positive),
   - $P(x = 0|C_2) = 0.97$ (true negative).
4. After observing positive test outcome $x = 1$, compute posteriors:

$$P(C_1|x = 1) \propto P(x = 1|C_1)P(C_1) = (0.98)0.008 = 0.0078,$$
$$P(C_2|x = 1) \propto P(x = 1|C_2)P(C_2) = (0.03)0.992 = 0.0298$$

5. Normalize to 1 to get the actual probabilities (divide by $0.0078 + 0.0298$).

Probability of error

1. If $p(C_1) = p(C_2) = \frac{1}{2}$, then we have



2. The coloured region may produce error. The probability of error equals to

$$\begin{aligned} P_e = p(error) &= p(x \in \mathcal{R}_1, C_2) + p(x \in \mathcal{R}_2, C_1) \\ &= \frac{1}{2}\int_{\mathcal{R}_1} p(x|C_2)dx + \frac{1}{2}\int_{\mathcal{R}_2} p(x|C_1)dx \end{aligned}$$

Bayes decision theory

Minimizing the classification error probability

Minimizing the classification error probability

1. We now show that the Bayesian classifier is optimal with respect to minimizing the classification probability.
2. Let $\mathcal{R}_1(\mathcal{R}_2)$ be the region in the feature space in which we decide in favor of C_1 (C_2).
3. Error is made if $x \in \mathcal{R}_1$ although it belongs to C_2, or if $x \in \mathcal{R}_2$ but it belongs to C_1. That is

$$\begin{aligned} P_e &= p(x \in \mathcal{R}_2, C_1) + p(x \in \mathcal{R}_1, C_2) \\ &= p(x \in \mathcal{R}_2 | C_1) p(C_1) + p(x \in \mathcal{R}_1 | C_2) p(C_2) \\ &= p(C_1) \int_{\mathcal{R}_2} p(x|C_1) + p(C_2) \int_{\mathcal{R}_1} p(x|C_2) \end{aligned}$$

4. Using the Bayes rule

$$P_e = \int_{\mathcal{R}_2} p(C_1|x) p(x) dx + \int_{\mathcal{R}_1} p(C_2|x) p(x) dx$$

5. Since $\mathcal{R}_1 \cup \mathcal{R}_2$ covers the space, from probability density function, we have

$$p(C_1) = \int_{\mathcal{R}_1} p(C_1|x) p(x) dx + \int_{\mathcal{R}_2} p(C_1|x) p(x) dx$$

6. By combining these two equation, we obtain

$$P_e = p(C_1) - \int_{\mathcal{R}_1} \left[p(C_1|x) - p(C_2|x) \right] p(x) dx$$

Minimizing the classification error probability (cont.)

1. The probability of error equals to

$$P_e = p(C_1) - \int_{\mathcal{R}_1} \left[p(C_1|x) - p(C_2|x)\right] p(x)dx$$

2. The probability of error is minimized if $\mathcal{R}_1$ is the region of the space in which

$$\left[p(C_1|x) - p(C_2|x)\right] > 0$$

3. Then $\mathcal{R}_2$ becomes the region where the reverse is true

$$\left[p(C_1|x) - p(C_2|x)\right] < 0$$

4. For classification task with M classes, x is assigned to class C_k with the following rule

$$\text{if } p(C_k|x) > p(C_j|x) \qquad \forall j \neq k$$

5. Show that this rule also minimizes the classification error probability for classification task with M classes.

Bayes decision theory

Minimizing the average risk

Minimizing the average risk

1. The classification error probability is not always the best criterion to be adopted for minimization. Why?
2. This because it assigns the same importance to all errors.
3. In some applications such as IDS, patient classification, and spam filtering some wrong decisions may have more serious implications than others.
4. In some cases, it is more appropriate to assign a penalty term to weight each error.
5. In such case, we try to minimize the following risk.

$$r = \lambda_{12} p(C_1) \int_{\mathcal{R}_2} p(x|C_1) dx + \lambda_{21} p(C_2) \int_{\mathcal{R}_1} p(x|C_2) dx$$

6. In general, the risk/loss associated to class C_k is defined as

$$r_k = \sum_{i=1}^{M} \lambda_{ki} \int_{\mathcal{R}_i} p(x|C_k) dx$$

7. The goal is to partition the feature space so that the average risk is minimized.

$$\begin{aligned} r &= \sum_{k=1}^{M} r_k p(C_k) \\ &= \sum_{i=1}^{M} \int_{\mathcal{R}_i} \left(\sum_{k=1}^{M} \lambda_{ki} p(x|C_k) p(C_k) \right) dx \end{aligned}$$

Minimizing the average risk (cont.)

1. The average risk is equal to

$$r = \sum_{i=1}^{M} \int_{\mathcal{R}_i} \left(\sum_{k=1}^{M} \lambda_{ki} p(x|C_k) p(C_k) \right) dx$$

2. This is achieved if each integral is minimized, so that

$$x \in \mathcal{R} \text{ if } r_i = \sum_{k=1}^{M} \lambda_{ki} p(x|C_k) p(C_k) < r_j = \sum_{k=1}^{M} \lambda_{kj} p(x|C_k) p(C_k) \qquad \forall j \neq i$$

3. When $\lambda_{ki} = 1$ (for $k \neq i$), minimizing the average risk is equivalent to minimizing the classification error probability.

4. In two–class case, we have

$$\begin{aligned} r_1 &= \lambda_{11} p(x|C_1) p(C_1) + \lambda_{21} p(x|C_2) p(C_2) \\ r_2 &= \lambda_{12} p(x|C_1) p(C_1) + \lambda_{22} p(x|C_2) p(C_2) \end{aligned}$$

5. We assign x to C_1 if $r_1 < r_2$, that is

$$(\lambda_{21} - \lambda_{22})\, p(x|C_2) p(C_2) < (\lambda_{12} - \lambda_{11})\, p(x|C_1) p(C_1)$$

Minimizing the average risk (cont.)

1. In other words,
$$x \in C_1(C_2) \text{ if } \frac{p(x|C_1)}{p(x|C_2)} > (<) \frac{p(C_2)}{p(C_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}$$
2. Assume that the loss matrix is in the form of
$$\Lambda = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}.$$
3. Then, we have
$$x \in C_1(C_2) \text{ if } \frac{p(x|C_1)}{p(x|C_2)} > (<) \frac{p(C_2)}{p(C_1)} \frac{\lambda_{21}}{\lambda_{12}}$$
4. When $p(C_1) = p(C_2) = \frac{1}{2}$, we have
$$x \in C_1(C_2) \qquad \text{if } p(x|C_1) > (<) p(x|C_2) \frac{\lambda_{21}}{\lambda_{12}}$$
5. If $\lambda_{21} > \lambda_{12}$, then x is assigned to C_2 if
$$p(x|C_2) > p(x|C_1) \frac{\lambda_{12}}{\lambda_{21}}$$
6. That is, $p(x|C_1)$ is multiplied by a factor less than one and the effect is the movement of the threshold to left of x_0.

Minimizing the average risk (example)

Example (Average error vs. average risk)

1. In a two class problem with a single feature x, distributions of two classes are

$$p(x|C_1) = \frac{1}{\sqrt{\pi}} exp\left(-x^2\right) \qquad p(C_1) = \frac{1}{2}$$

$$p(x|C_2) = \frac{1}{\sqrt{\pi}} exp\left(-(x-1)^2\right) \qquad p(C_2) = \frac{1}{2}$$

2. Compute x_0 for minimum error probability classifier. x_0 is the solution of

$$\frac{1}{\sqrt{\pi}} exp\left(-x_0^2\right) = \frac{1}{\sqrt{\pi}} exp\left(-(x_0-1)^2\right)$$

   $x_0 = \frac{1}{2}$ is the solution of the above equation.

3. If the following loss matrix is given, compute x_0 for the minimum average risk classifier.

$$\Lambda = \begin{bmatrix} 0 & 0.5 \\ 1.0 & 0 \end{bmatrix}.$$

4. x_0 must satisfy the following equation.

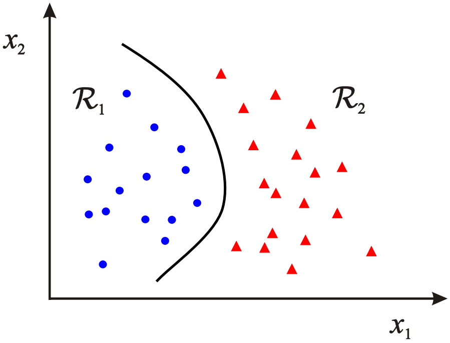
$$\frac{1}{\sqrt{\pi}} exp\left(-x_0^2\right) = \frac{2}{\sqrt{\pi}} exp\left(-(x_0-1)^2\right) \qquad x_0 = \frac{1-\ln 2}{2} < \frac{1}{2}$$

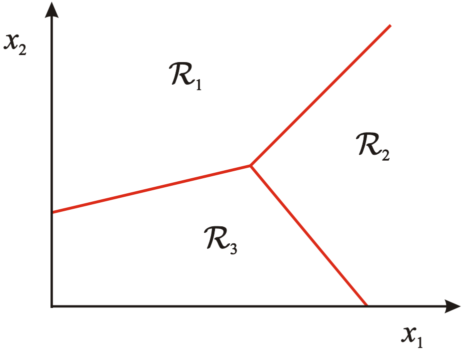
Bayes decision theory

Discriminant function and decision surface

Discriminant function and decision surface

1. As discussed, minimizing either the risk or the error probability is equivalent to partitioning the feature space into M regions for M classes.
2. If two regions $\mathcal{R}_i$ and $\mathcal{R}_j$ happen to be continuous, then they are separated by a decision surface in the multi-dimensional feature space.





Discriminant function and decision surface(cont.)

1. For the minimum error probability case, this surface described by

$$p(C_i|x) - p(C_j|x) = 0.$$

2. This difference from one side of the surface is positive and from other side, it is negative.
3. Sometimes, instead of working directly with probabilities (or risks), it is more convenient to work with an equivalent function of them such as

$$g_i(x) = f(p(C_i|x))$$

   Function $f(.)$ is monotonically increasing. (why?)
4. Function $g_i(x)$ is known as a discriminant function.
5. Now, the decision test is stated as

$$\text{Classify } x \text{ in } C_i \qquad \text{if } g_i(x) > g_j(x) \qquad \forall j \neq i$$

6. The decision surfaces, separating continuous regions are stated as

$$g_{ij}(x) = g_i(x) - g_j(x) \qquad \forall i, j = 1, 2, \ldots, M, \text{ and } j \neq i$$

Bayes decision theory

Bayesian classifiers for Normally distributed classes

Discriminant function for Normally distributed classes

1. The one dimensional Gaussian distribution with mean of μ and variance σ^2 is given by

$$p(x) = \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The $D-$dimensional Gaussian distribution with mean of μ and covariance matrix Σ is

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

2. What is the optimal classifier when the involved pdfs are $\mathcal{N}(\mu, \Sigma)$?
3. For Gaussian densities, it is preferable to work with the following discriminant functions.

$$\begin{aligned} g_i(x) &= \ln[p(x|C_i)p(C_i)] \\ &= \ln p(x|C_i) + \ln p(C_i) \end{aligned}$$

4. Or

$$\begin{aligned} g_i(x) &= -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) + w_{i0} \\ w_{i0} &= -\frac{D}{2}\ln(2\pi) - \frac{D}{2}\ln|\Sigma_i| + \ln p(C_i) \end{aligned}$$

5. By expanding the above equation, we obtain the following quadratic form.

$$g_i(x) = -\frac{1}{2}x^T\Sigma_i^{-1}x + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x + w_{i0}$$

Discriminant function for Normally distributed classes(example)

1. For Normally distributed classes, we have the following quadratic form classifier.

$$g_i(x) \quad = \quad -\frac{1}{2}x^T\Sigma_i^{-1}x + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x + w_{i0}$$

2. Assume

$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$$

   Thus we have

$$g_i(x) \quad = \quad -\frac{1}{2\sigma_i^2}\left(x_1^2 + x_2^2\right) + \frac{1}{2\sigma_i^2}\left(\mu_{i1}x_1 + \mu_{i2}x_2\right) - \frac{1}{2\sigma_i^2}\left(\mu_{i1}^2 + \mu_{i2}^2\right) + w_{i0}$$

3. Obviously the associated decision curves $g_i(x) - g_j(x) = 0$ are quadratics.
4. In this case the Bayesian classifier is a quadratic classifier, i.e. the partition of the feature space is performed via quadratic decision surfaces.

Discriminant function for Normally distributed classes (cont.)

1. The discriminant functions for optimal classifier when the involved pdfs are $\mathcal{N}(\mu, \Sigma)$ have the following form

$$\begin{aligned} g_i(x) &= -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) + w_{i0} \\ w_{i0} &= -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i| + \ln p(C_i) \end{aligned}$$

2. By expanding the above equation, we obtain the following quadratic form.

$$g_i(x) = -\frac{1}{2}x^T\Sigma_i^{-1}x + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x + w_{i0}$$

3. Based on the above equations, We distinguish three distinct cases:
   - When $\Sigma_i = \sigma^2 I$, where σ^2 is a scalar and I is the identity matrix;
   - $\Sigma_i = \Sigma$, i.e. all classes have equal covariance matrices;
   - Σ_i is arbitrary.

Discriminant function for Normally distributed classes ($\Sigma_i = \sigma^2 I$)

1. The discriminant functions for optimal classifier when pdfs are $\mathcal{N}(\mu, \Sigma)$ have form

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + w_{i0}$$

2. By replacing $\Sigma_i = \sigma^2 I$ in the above equation, we obtain

$$\begin{aligned} g_i(x) &= -\frac{1}{2}(x-\mu_i)^T(\sigma^2)^{-1}(x-\mu_i) + w_{i0} \\ &= -\frac{||x-\mu_i||^2}{2\sigma^2} + w_{i0} \\ &= -\frac{1}{2\sigma^2}\left(x^T x - 2\mu_i^T x + \mu_i^T \mu_i\right) + w_{i0} \end{aligned}$$

3. Terms $x^T x$ and other constants are equal for all classes so they can be dropped.

$$g_i(x) = \frac{1}{\sigma^2}\left(\mu_i^T x - \frac{1}{2}\mu_i^T \mu_i\right) + w_{i0}$$

4. This is a linear discriminant function $g_i(x) = w_i^T x + w_{i0}$ with

$$w_i = \frac{\mu_i}{\sigma^2}$$

$$w_{i0} = -\frac{\mu_i^T \mu_i}{2\sigma^2} + \ln p(C_i)$$

Discriminant function for Normally distributed classes ($\Sigma_i = \sigma^2 I$)

1. For this case, the discriminant functions are equal to

$$g_i(x) = \frac{1}{\sigma^2}\mu_i^T x + w_{i0}$$

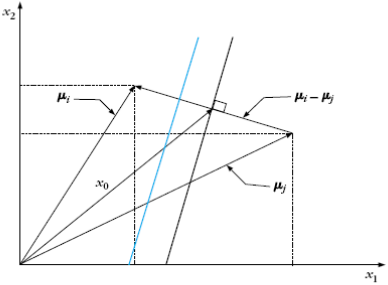
2. The corresponding hyperplanes can be written as

$$\begin{aligned}
g_{ij}(x) = g_i(x) - g_j(x) &= \frac{1}{\sigma^2}\mu_i^T x + w_{i0} - \frac{1}{\sigma^2}\mu_j^T x + w_{j0} \\
&= \frac{1}{\sigma^2}\left(\mu_i - \mu_j\right)^T x + w_{i0} - w_{j0} \\
&= w^T(x - x_0) = 0 \\
w &= \mu_i - \mu_j \\
x_0 &= \frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \ln\left(\frac{p(C_i)}{p(C_j)}\right)\frac{\mu_i - \mu_j}{||\mu_i - \mu_j||^2}
\end{aligned}$$

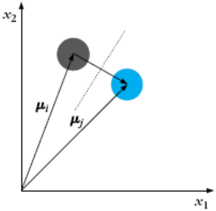
3. This implies that the decision surface is a hyperplane passing through the point x_0.
4. For any x on the decision hyperplane, vector $(x - x_0)$ also lies on the hyperplane and hence $(\mu_i - \mu_j)$ is orthogonal to the decision hyperplane.

Discriminant function for Normally distributed classes($\Sigma_i = \sigma^2 I$)

1. When $p(C_i) = p(C_j)$, then $x_0 = \frac{1}{2}(\mu_i + \mu_j)$ and the hyperplane passes through the average of μ_i and μ_j.
2. When $p(C_i) < p(C_j)$, the hyperplane located closer to μ_i.
3. When $p(C_i) > p(C_j)$, the hyperplane located closer to μ_j.



4. If σ^2 is small with respect to $||\mu_i - \mu_j||$, the location of the hyperplane is insensitive to the values of $p(C_i)$ and $p(C_j)$.



Discriminant function for Normally distributed classes ($\Sigma_i = \Sigma$)

1. The discriminant functions for optimal classifier when the pdfs are $\mathcal{N}(\mu, \Sigma)$ have form

$$g_i(x) \quad = \quad -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) + w_{i0}$$

2. By replacing $\Sigma_i = \Sigma$ in the above equation, we obtain

$$\begin{aligned} g_i(x) &= -\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i) + w_{i0} \\ &= -\frac{1}{2}x^T\Sigma^{-1}x + \frac{1}{2}x^T\Sigma^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma^{-1}x + w_{i0} \\ &= -\frac{1}{2}x^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}x - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + w_{i0} \end{aligned}$$

3. Terms x^Tx and other constants are equal for all classes and can be dropped. This gives

$$g_i(x) \quad = \quad \frac{1}{2}\left(2\mu_i^T\Sigma^{-1}x - \mu_i^T\Sigma^{-1}\mu_i\right) + \ln p(C_i)$$

4. This is a linear discriminant function with

$$g_i(x) \quad = \quad w_i^Tx + w'_{i0}$$

5. with the following parameters

$$\begin{aligned} w_i &= \mu_i\Sigma^{-1} \\ w'_{i0} &= -\frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln p(C_i) \end{aligned}$$

Discriminant function for Normally distributed classes ($\Sigma_i = \Sigma$)

1. For this case, the discriminant functions are equal to

$$g_i(x) = \frac{1}{2}\left(2\mu_i^T\Sigma^{-1}x - \mu_i^T\Sigma^{-1}\mu_i\right) + \ln p(C_i)$$

2. The corresponding hyperplanes can be written as

$$\begin{aligned} g_{ij}(x) = g_i(x) - g_j(x) &= w^T(x - x_0) = 0 \\ w &= \Sigma^{-1}(\mu_i - \mu_j) \\ x_0 &= \frac{1}{2}(\mu_i + \mu_j) - \ln\frac{p(C_i)}{p(C_j)}\frac{\mu_i - \mu_j}{||\mu_i - \mu_j||^2_{\Sigma^{-1}}} \\ &= \frac{1}{2}(\mu_i + \mu_j) - \ln\frac{p(C_i)}{p(C_j)}\frac{\mu_i - \mu_j}{(\mu_i - \mu_j)^T\Sigma^{-1}(\mu_i - \mu_j)} \end{aligned}$$

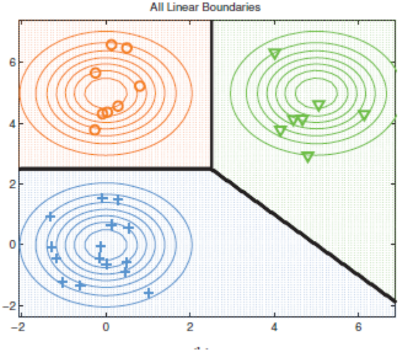
3. The decision function is no longer orthogonal to vector $(\mu_i - \mu_j)$ but to its linear transformation $\Sigma^{-1}(\mu_i - \mu_j)$.

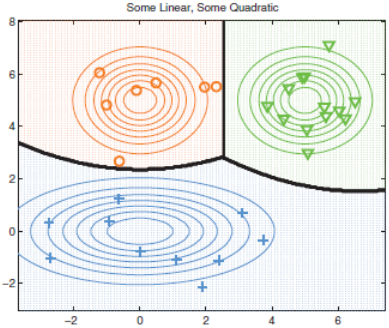
Discriminant function for Normally distributed classes (any Σ_i)

1. The discriminant functions for optimal classifier when the pdfs are $\mathcal{N}(\mu, \Sigma)$ have form of

$$g_i(x) \quad = \quad -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + \ln p(C_i) - \frac{D}{2}\ln(2\pi) - \frac{D}{2}\ln|\Sigma_i|$$

2. The discriminant functions cannot be simplified much further. Only the constant term $\frac{D}{2}\ln(2\pi)$ can be dropped.
3. Discriminant functions are not linear but quadratic.
4. They have much more complicated decision regions than the linear classifiers of the two previous cases.



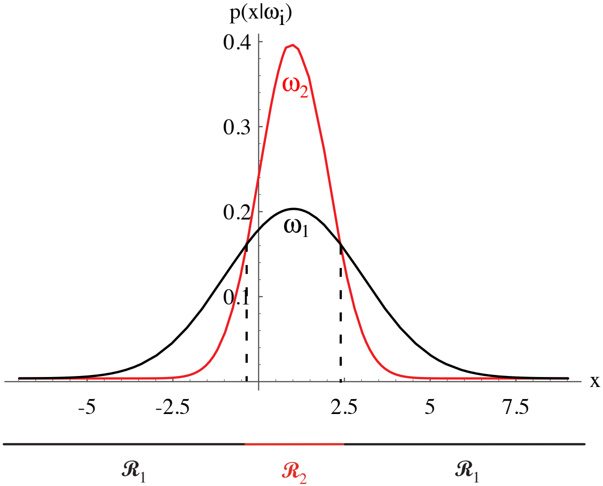


Discriminant function for Normally distributed classes (any Σ_i)

1. The discriminant functions for optimal classifier when the pdfs are $\mathcal{N}(\mu, \Sigma)$ have form of

$$g_i(x) \quad = \quad -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) + \ln p(C_i) - \frac{D}{2}\ln(2\pi) - \frac{D}{2}\ln|\Sigma_i|$$

2. Now, decision surfaces are also quadratic and the decision regions do not have to be even connected sets.



Bayes decision theory

Minimum distance classifier

Minimum distance classifier

1. Assume that we have $p(C_i) = p(C_j)$ with the same covariance matrix, then $g_i(x)$ equals to

$$g_i(x) \quad = \quad -\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i)$$

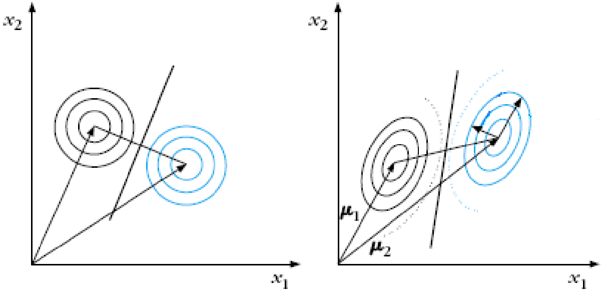
2. For diagonal covariance matrix $(\Sigma = \sigma^2 I)$, the maximum $g_i(x)$ implies minimum Euclidean distance.

$$d_\epsilon \quad = \quad ||x-\mu_i||$$

   Feature vectors are assigned to classes according to their Euclidean distance from their respective mean points.

3. For non-diagonal covariance matrix, the maximum $g_i(x)$ is equivalent to minimizing Mahalanobis distance (Σ^{-1}–norm).

$$d_m \quad = \quad (x-\mu_i)^T\Sigma^{-1}(x-\mu_i)$$



Bayes decision theory

Bayesian classifiers for independent binary features

Bayesian classifiers for independent binary features

1. Let features be binary-valued and independent.
2. Let the class conditional density for each feature be the Bernoulli distribution. This yields

$$p(x|C_i) = \prod_{j=1}^{D} q_{ij}^{x_j} (1 - q_{ij})^{(1-x_j)}$$

   q_{ij} (for $j = 1, 2, \ldots, D$) are parameters for the class conditional density of the class C_i.

3. The discriminant function is

$$\begin{aligned} g_i(x) = \ln p(C_i|x) = \ln p(x|C_i)p(C_i) &= \ln \prod_{j=1}^{D} q_{ij}^{x_j} (1 - q_{ij})^{(1-x_j)} p(C_i) \\ &= \sum_{j=1}^{D} [x_j \ln q_{ij} + \ln (1 - q_{ij}) - x_j \ln (1 - q_{ij})] + \ln p(C_i) \end{aligned}$$

4. These are linear discriminant functions

$$g_i(x) = W_i^T x + w_{i0} = \sum_{j=1}^{D} w_{ij} x_j + w_{i0}$$

$$w_{i0} = \ln p(C_i) + \ln(1 - q_{ij})$$

$$w_{ij} = \sum_{j=1}^{D} \ln q_{ij} - \ln(1 - q_{ij})$$

Supervised learning of the Bayesian classifiers

Supervised learning of the Bayesian classifiers

1. We assumed that the class conditional pdfs $p(x|C_i)$ and the prior probabilities $p(C_i)$ were known. In practice, this is never the case and we study supervised learning of class conditional pdfs.
2. For supervised learning we need training samples. In the training set there are feature vectors from each class and we re-arrange training samples based on their classes.

   $$S_i = \{(x_{i1}, t_{i1}), (x_{i2}, t_{i2}), \ldots, (x_{iN_i}, t_{iN_i})\}$$

   N_i is the number of training samples from the class C_i.
3. We assume that the training samples in the sets S_i are occurrences of the independent random variables.

Supervised learning of the Bayesian classifiers (cont.)

1. We assume that the training samples in the sets S_i are occurrences of the independent random variables.
2. The training data may be collected in two distinct ways. These are meaningful when we need to learn the prior probabilities.
   - In mixture sampling, a set of objects are randomly selected, their feature vectors extracted and then they hand-classified to the most appropriate classes. The prior probability of each class is estimated as

     $$p(C_i) = \frac{N_i}{N}$$

   - In separate sampling, the training data for each class is collected separately. In this case, the prior probabilities cannot be deduced and it is most reasonable to assume that they are known (If they are unknown, we usually assume that the prior probabilities for all classes are equal.)

Supervised learning of the Bayesian classifiers (Cont.)

1. We assumed that the probability density functions are known. In most cases, these probability density functions are not known and the underlying pdf will be estimated from the available data.
2. There are various ways to estimate the probability density functions.
3. If we know the type of the pdf, we can estimate the parameters of the pdf such as mean and variance from the available data. These methods are known as parametric methods.
   - In the estimative approach to parametric density estimation, we use an estimate of the parameter θ_j in the parametric density.

     $$p(x|C_j) = p(x|C_j; \hat{\theta}_j)$$

     $\hat{\theta}_j$ is an estimate of the parameter θ_j based on the data samples.
   - In the Bayesian/predictive approach, we assume that we don't know the true value of parameter θ_j. This approach treats θ_j as an unknown random variable.
4. In many cases, we may not have the information about the type of the pdf, but we may know certain statistical parameters such as the mean and the variance. These methods are known as nonparametric methods.

Supervised learning of the Bayesian classifiers

Parametric methods for density estimation

Parametric methods for density estimation

1. In parametric methods, we assume that the sample is drawn from some known distribution (for example Gaussian). But the parameters of this distribution are not known and our goal is to estimate these parameters from the data.
2. Parametric methods are advantageous in that the model is defined by a small number of parameters, and when these parameters are estimated, the distribution as a whole is known.
3. The following methods usually are used to estimate the parameters of the distribution
   - Maximum likelihood estimation
   - Bayesian estimation
   - Maximum a posteriori probability estimation
   - Maximum entropy estimation
   - Mixture Models

Maximum likelihood parameter estimation

1. Consider an $M-$class problem with feature vectors distributed according to $p(x|C_i)$ (for $i = 1, 2, \dots, M$).
2. We assume that $p(x|C_i)$ belongs to some family of parametric distributions. For example, we assume that $p(x|C_i)$ is a normal density with unknown parameters $\theta_i = (\mu_i, \Sigma_i)$.
3. To show the dependence on θ_i, we denote $p(x|C_i) = p(x|C_i; \theta_i)$. The class C_i defines the parametric family, and the parameter vector θ_i defines the member of that parametric family.
4. The parametric families do not need to be same for all classes.
5. Our goal is to estimate the unknown parameters using a set of known feature vectors in each class.
6. If we assume that data from one class do not affect the parameter estimation of the others, we can formulate the problem independent of classes and simplify our notation $(p(x; \theta))$. Then solve the problem for each class independently.
7. Let $X = \{x_1, x_2, \dots, x_N\}$ be random samples drawn from pdf $p(x; \theta)$. We form the joint pdf $p(X; \theta)$.
8. Assuming statistical independence between the different samples, we have

$$p(X; \theta) \quad = \quad p(x_1, x_2, \dots, x_N; \theta_i) = \prod_{k=1}^{N} p(x_k; \theta)$$

Maximum likelihood parameter estimation (cont.)

1. $p(X;\theta)$ is a function of θ and is known as likelihood function.
2. The maximum likelihood (ML) method estimates θ so that the likelihood function takes its maximum value, that is,

$$\hat{\theta}_{ML} \quad = \quad \underset{\theta}{argmax} \prod_{k=1}^{N} p(x_k;\theta)$$

3. A necessary condition that $\hat{\theta}_{ML}$ must satisfy in order to be a maximum is the gradient of the likelihood function with respect to θ to be zero.

$$\frac{\partial \prod_{k=1}^{N} p(x_k;\theta)}{\partial\theta} \quad = \quad 0$$

4. It is more convenient to work with the logarithm of the likelihood function than with the likelihood function itself. Hence, we define the log likelihood function as

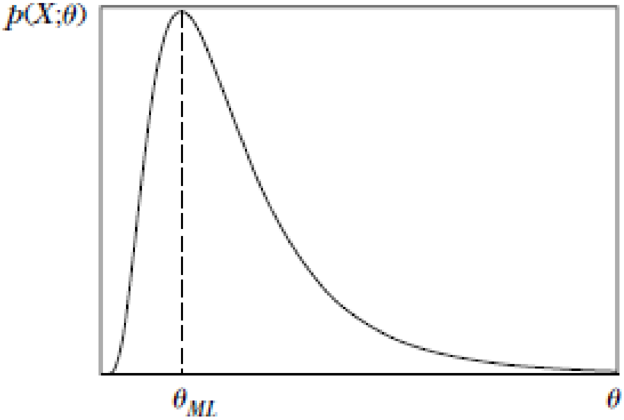
$$\begin{aligned} LL(\theta) &= \ln \prod_{k=1}^{N} p(x_k;\theta) \\ &= \sum_{k=1}^{N} \ln p(x_k;\theta) \end{aligned}$$

Maximum likelihood parameter estimation (cont.)

1. In order to find $\hat{\theta}_{ML}$, it must satisfy

$$\begin{aligned}\frac{\partial LL(\theta)}{\partial \theta} &= \frac{\sum_{k=1}^{N} \partial \ln p(x_k;\theta)}{\partial \theta} \\ &= \sum_{k=1}^{N} \frac{1}{p(x_k;\theta)} \frac{\partial p(x_k;\theta)}{\partial \theta} \\ &= 0\end{aligned}$$

2. The single unknown parameter case



Maximum likelihood estimation for normal distribution

1. Let $x_1, x_2, \dots, x_N$ be vectors sampled from a normal distribution with known covariance matrix and unknown mean, that is,

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

   Obtain ML-estimate of the unknown mean vector.

2. For N available samples, we have

$$LL(\mu) = \ln \prod_{k=1}^{N} p(x_k; \mu) = -\frac{N}{2}\ln[(2\pi)^D|\Sigma|^D] - \frac{1}{2}\sum_{k=1}^{N}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu)$$

3. Taking the gradient with respect to μ, we obtain

$$\frac{\partial LL(\mu)}{\partial \mu} = \begin{bmatrix} \frac{\partial LL(\mu)}{\partial \mu_1} \\ \frac{\partial LL(\mu)}{\partial \mu_2} \\ \vdots \\ \frac{\partial LL(\mu)}{\partial \mu_D} \end{bmatrix} = \sum_{k=1}^{N}\Sigma^{-1}(x_k-\mu) = 0$$

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{k=1}^{N} x_k$$

   That is, the ML estimate of the mean, for Gaussian densities, is the sample average.

Maximum likelihood estimation for normal distribution

1. Assume $x_1, x_2, \dots, x_N$ have been generated by a one-dimensional Gaussian pdf of known mean, μ, but of unknown variance, that is,

$$p(x;\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

   Obtain ML-estimate (MLE) of the unknown variance.

2. For N available samples, we have

$$LL(\sigma^2) = \ln\prod_{k=1}^{N} p(x_k;\sigma^2) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{N}(x_k-\mu)^2$$

3. Taking the derivative of the above with respect to σ^2 and equating to zero, we obtain

$$\frac{dLL(\sigma^2)}{d\sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{k=1}^{N}(x_k-\mu)^2 = 0$$

4. Solving the above equation with respect to σ^2, results in

$$\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{k=1}^{N}(x_k-\mu)^2$$

Evaluating an estimator

1. Let x be a sample from a pdf with parameter θ, and $\hat{\theta}$ be an estimator of θ.
2. To evaluate the quality of this estimator, we can measure how much it is different from θ, that is $(\hat{\theta} - \theta)^2$.
3. But since it is a random variable (it depends on the sample), we need to average mean square error over possible x and consider $r(\hat{\theta}, \theta)$, the mean square error of estimator $\hat{\theta}$ defined as

$$r(\hat{\theta}, \theta) = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right]$$

4. The bias of an estimator is given as

$$bias_{\theta}(\hat{\theta}) = \mathbb{E}\left[\hat{\theta}\right] - \theta$$

5. If $bias_{\theta}(\hat{\theta}) = 0$ for all values of θ, then we say that $\hat{\theta}$ is an unbiased estimator.

Evaluating an estimator: bias and variance (cont.)

1. If $bias_\theta(\hat{\theta}) = 0$ for all values of θ, then we say that $\hat{\theta}$ is an unbiased estimator.

Example (Sample average)

Assume that N samples x_k are drawn from some density with mean μ, the sample average, $\hat{\mu}$, is an unbiased estimator of the mean, μ, because

$$\mathbb{E}\left[\hat{\mu}\right] = \mathbb{E}\left[\frac{\sum_k x_k}{N}\right] = \frac{1}{N}\sum_k \mathbb{E}\left[x_k\right] = \frac{N\mu}{N} = \mu$$

2. An estimator $\hat{\theta}$ is consistent estimator if

$$\lim_{N\to\infty} Var(\hat{\theta}) \to 0$$

Example (Sample average)

Assume that N samples x_k are drawn from some density with mean μ, the sample average, $\hat{\mu}$, is a consistent estimator of the mean, μ, because

$$Var(\hat{\mu}) = Var\left(\frac{\sum_k x_k}{N}\right) = \frac{1}{N^2}\sum_k Var[x_k] = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N}$$

3. As N gets larger, $\hat{\mu}$ deviates less from μ.

Evaluating an estimator (cont.)

Example (Sample variance)

Assume that N samples x_k are drawn from some density with variance σ^2, the sample variance, $\hat{\sigma}^2$, is a biased estimator of the variance, σ^2, because

$$\begin{aligned}\hat{\sigma}^2 &= \frac{\sum_k (x_k - \hat{\mu})^2}{N} = \frac{\sum_k x_k^2 - N\hat{\mu}^2}{N} \\ \mathbb{E}\left[\hat{\sigma}^2\right] &= \frac{\sum_k \mathbb{E}\left[x_k^2\right] - N\,\mathbb{E}\left[\hat{\mu}^2\right]}{N}\end{aligned}$$

Given that $\mathbb{E}\left[x^2\right] = Var(x) + \mathbb{E}\left[x\right]^2$, we can write

$$\begin{aligned}\mathbb{E}\left[x_k^2\right] &= \sigma^2 + \mu^2 \\ \mathbb{E}\left[\hat{\mu}^2\right] &= \frac{\sigma^2}{N} + \mu^2\end{aligned}$$

Replacing back, we obtain

$$\mathbb{E}\left[\hat{\sigma}^2\right] = \frac{N(\sigma^2 + \mu^2) - N(\sigma^2/N + \mu^2)}{N} = \left(\frac{N-1}{N}\right)\sigma^2 = \left(1 - \frac{1}{N}\right)\sigma^2 \neq \sigma^2$$

This is an example of an asymptotically unbiased estimator whose bias goes to 0 as N goes to ∞.

Properties of maximum likelihood estimation

1. If θ_0 is the true value of the unknown parameter in $p(x;\theta)$, it can be shown that under generally valid conditions the following are true
   - The ML estimate is asymptotically unbiased, that is
   $$\lim_{N\to\infty} \mathbb{E}\left[\hat{\theta}_{ML}\right] = \theta_0$$
   - The ML estimate is asymptotically consistent, that is, it satisfies
   $$\lim_{N\to\infty} Prob\left[||\hat{\theta}_{ML} - \theta_0||^2 \leq \epsilon\right] = 1$$
   for arbitrarily small ϵ. A stronger condition for consistency is also true
   $$\lim_{N\to\infty} \mathbb{E}\left[\left\|\hat{\theta}_{ML} - \theta_0\right\|^2\right] = 0$$
   - The ML estimate is asymptotically efficient; that is, this achieves the lowest value of variance, which any estimate can achieve (Cramer-–Rao lower bound).
   - The pdf of the ML estimate as $N \to \infty$ approaches the Gaussian distribution with mean θ_0.
2. In summary, the ML estimator is unbiased, is normally distributed, and has the minimum possible variance. However, all properties are valid only for large values of N (Why? check it.).
3. If N is small, little can be said about the ML-estimates in general.

Maximum a posteriori estimation (MAP)

1. In MLE, we considered θ as an unknown parameter.
2. In MAP estimation, we consider θ as a random vector, and we will estimate its value based on sample X.
3. From the Bayes theorem, we have

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

4. The MAP estimation, $\hat{\theta}_{MAP}$ is defined at the point where $p(\theta|X)$ becomes maximum.
5. A necessary condition that $\hat{\theta}_{MAP}$ must satisfy in order to be a maximum is its gradient with respect to θ to be zero.

$$\begin{aligned} \frac{\partial p(\theta|X)}{\partial\theta} &= 0 \\ \frac{\partial p(X|\theta)p(\theta)}{\partial\theta} &= 0 \end{aligned}$$

6. The difference between ML and MAP estimates lies in the involvement of $p(\theta)$ in the MAP.
7. If $p(\theta)$ is uniform, then both estimates yield identical results.

Maximum a posteriori estimation (example)

1. Let $x_1, x_2, \ldots, x_N$ be vectors drawn from a normal distribution with known covariance matrix $\Sigma = \sigma^2 I$ and unknown mean, that is

$$p(x_k; \mu) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{D/2}} exp\left(-\frac{1}{2}(x_k - \mu)^T \sigma^{-1}(x_k - \mu)\right)$$

2. Assume that the unknown mean vector μ is known to be normally distributed as

$$p(\mu) = \frac{1}{(2\pi)^{D/2}\sigma_\mu^D} exp\left(-\frac{1}{2}\frac{||\mu - \mu_0||^2}{\sigma_\mu^2}\right)$$

3. The MAP estimate is given by the solution of

$$\frac{\partial}{\partial\mu} \ln\left(\prod_{k=1}^{N} p(x_k|\mu)p(\mu)\right) = 0$$

4. For $\Sigma = \sigma^2 I$, we obtain

$$\hat{\mu}_{MAP} = \frac{\mu_0 + \frac{\sigma_\mu^2}{\sigma^2}\sum_{k=1}^{N} x_k}{1 + \frac{\sigma_\mu^2}{\sigma^2}N}$$

5. When $\frac{\sigma_\mu^2}{\sigma^2} \gg 1$, then $\hat{\mu}_{MAP} \approx \hat{\mu}_{ML}$.
6. MLE and MAP find a specific point estimate of unkown parameter.

Bayesian estimation

1. Both MLE and MAP compute a specific estimate of the unknown parameter vector θ.
2. Sometimes, before looking at dataset, we may have some prior information on the possible value range that a parameter, θ, may take.
3. This information is quite useful and should be used, especially when the dataset is small.
4. The prior information doesn't tell us exactly what the parameter value is (otherwise we would not need the dataset), and we model this uncertainty by viewing θ as a random variable and by defining a prior density for it, $p(\theta)$.
5. For example, we are told that θ is approximately normal and with 90 percent confidence, θ lies between 5 and 9, symmetrically around 7.

Bayesian estimation

1. The prior density $p(\theta)$ tells the likely values that θ may take before looking at the sample.
2. This is combined with what the sample data tells ($p(X|\theta)$) using Bayes rule and get the posterior density of θ, which tells the θ values after looking at the sample.

$$\begin{aligned} p(\theta|X) &= \frac{p(X|\theta)p(\theta)}{p(X)} \\ &= \frac{p(X|\theta)p(\theta)}{\int p(X|\theta')p(\theta')d\theta'} \end{aligned}$$

3. Given the set of X and the prior information $p(\theta)$, the goal is to compute the conditional pdf $p(x|X)$.

Bayesian estimation (cont.)

1. For estimating the density at x, we have

$$\begin{aligned} p(x|X) &= \int p(x,\theta|X)d\theta \\ &= \int p(x|\theta, X)p(\theta|X)d\theta \\ &= \int p(x|\theta)p(\theta|X)d\theta \end{aligned}$$

   $p(x|\theta, X) = p(x|\theta)$, because once we know θ, the sufficient statics, we know everything about the distribution.

2. Evaluating the integrals may be quite difficult, except in case where the posterior has a nice form.
3. When the full integration is not feasible, we reduce it to a single point.
4. If we can assume that $p(\theta|X)$ has a narrow peak around its mode, then using maximum posteriori (MAP) estimate will make the calculation easier.
5. If $p(\theta|X)$ is known, then $p(x|X)$ is average of $p(x|\theta)$ with respect to θ, that is

$$p(x|X) = \mathop{\mathbb{E}}_{\theta}\left[p(x|\theta)\right]$$

Bayesian estimation (example)

1. Let $x \sim \mathcal{N}(\mu, \sigma^2)$ with unknown mean $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Hence, we have

$$p(\mu|X) = \frac{p(X|\mu)p(\mu)}{p(X)} = \frac{1}{\alpha}\prod_{k=1}^{N} p(x_k|\mu)p(\mu)$$

2. $p(X)$ is a constant denoted as α, or

$$p(\mu|X) = \frac{1}{\alpha}\prod_{k=1}^{N}\frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x_k-\mu)^2}{2\sigma^2}\right)\frac{1}{\sqrt{2\pi}\sigma_0} exp\left(-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right)$$

3. When N samples are given, $p(\mu|X)$ turns out to be a Gaussian (show it), that is

$$\begin{aligned} p(\mu|X) &= \frac{1}{\sqrt{2\pi}\sigma_N} exp\left(-\frac{1}{2}\frac{(\mu-\mu_N)^2}{\sigma_N^2}\right) \\ \mu_N &= \frac{N\sigma_0^2\bar{x}_N + \sigma^2\mu_0}{N\sigma_0^2+\sigma^2} \\ \sigma_N^2 &= \frac{\sigma^2\sigma_0^2}{N\sigma_0^2+\sigma^2} \end{aligned}$$

4. By some algebraic simplification, we obtain the following Gaussian pdf

$$p(x|X) = \frac{1}{\sqrt{2\pi(\sigma^2+\sigma_N^2)}} exp\left(-\frac{1}{2}\frac{(x-\mu_N)^2}{\sigma^2+\sigma_N^2}\right)$$

Example (Coin toss)

1. We have three biased coins.
2. The probability of landing coins at heads is λ, p, q, respectively.
3. The tossing scenario is
   - Toss coin 0;
   - If head, we toss coin 1 another 4 times;
   - Otherwise, we toss coin 2 another 4 times;
   - We can only observe the sequence produced by coins 1 and/or coin 2, which are data y_j for $j \in \{1, 2, 3, 4\}$: HHHT, HTHT, HHHT, HTTH;
   - The goal is to estimate most likely values for $\theta = (\lambda, p, q)^\top$.
4. Thus, we have no idea which data points came from coin 1 and which from coin 2.
   - Let $\theta_n = (\lambda_n, p_n, q_n)^\top$ be the current estimate of parameters, for simplicity, we just write as $(\lambda, p, q)^\top$
   - Let z be the hidden *indicator* variable, which coin was tossed at the beginning of each attempt.
     - If $z = 1$, coin 1 was tossed;
     - If $z \neq 1$, coin 2 was tossed;
   - Suppose there were m coin tosses and h_j heads in the jth coin toss y_j.
   - What is the probability $P(z)$ given $\theta = (\lambda, p, q)^T$ and y?

Mixture models for density estimation

1. An alternative way to model an unknown density function $p(x)$ is via linear combination of M density functions in the form of

$$p(x) \quad = \quad \sum_{m=1}^{M} \pi_m p(x|m)$$

   where

$$\sum_{m=1}^{M} \pi_m = 1$$

$$\int_x p(x|m)dx = 1$$

2. This modeling implicitly assumes that each point x may be drawn from any M model distributions with probability π_m (for $m = 1, 2, \ldots, M$).
3. It can be shown that this modeling can approximate closely any continuous density function for a sufficient number of mixtures M and appropriate model parameters.

Mixture models(cont.)

1. First, we select a set of density components $p(x|m)$ in the parametric form $p(x|m, \theta)$.

$$p(x; \theta) = \sum_{m=1}^{M} \pi_m p(x|\theta_m)$$

2. Then, we compute parameters $\theta_1, \theta_2, \dots, \theta_M$ and $\pi_1, \pi_2, \dots, \pi_M$ based on training data.
3. The parameter set is defined as $\theta = \{\pi_\mathbf{1}, \pi_\mathbf{2}, \dots, \pi_\mathbf{M}, \theta_\mathbf{1}, \theta_\mathbf{2}, \dots, \theta_\mathbf{M}\}$ and $\sum_i \pi_1 = 1$.
4. Given data $X = \{x_1, x_2, \dots, x_N\}$, and assuming mixture model

$$p(x; \theta) = \sum_{m=1}^{M} \pi_m p(x|\theta_m)$$

   we want to estimate parameters.

5. In order to estimate each π_m, we can count how many points from X coming from each of M components then normalize by N.

$$\hat{\pi}_m = \frac{N_m}{N}$$

   Each N_m can be obtained from $N_m = \sum_{n=1}^{N} z_{mn}$ and

$$z_{mn} = \begin{cases} 1 & \text{if the } n^{th} \text{ point was drawn from component } m \\ 0 & \text{otherwise.} \end{cases}$$

Mixture models(cont.)

1. What are the values parameters of each component $\hat{\theta}_m$?
2. We need to estimate $\hat{\theta}_m$ which maximizes the likelihood of the data points that are drawn from component m under the parametric form $p(x|\theta_m)$.
3. If the mixture components were Gaussian, then MLE for the component mean vectors would be

$$\hat{\mu}_m = \frac{\sum_{n=1}^{N} z_{mn} x_n}{\sum_{n=1}^{N} z_{mn}}$$

4. The estimation for covariance matrix for each component would be

$$\hat{\Sigma}_m = \frac{1}{\sum_{n=1}^{N} z_{mn}} \sum_{n=1}^{N} z_{mn}(x_n - \hat{\mu}_m)(x_n - \hat{\mu}_m)^T$$

5. The difficulty is that we do not know z_{mn}. This is a major difficulty because the variables z_{mn} are hidden or latent then our ML estimates cannot follow in the straightforward manner we had anticipated.
6. The problem is that we assumed knowledge of the values for indicator variables z_{mn}.

Expectation maximization (cont.)

1. EM algorithm has two steps: Expectation & Maximization
2. In E-step, the missing data are estimated given the observed data and the current estimate of the model parameters.

$$p(m|x_n) \quad = \quad \frac{p(x_n|\theta_m)p(m)}{\sum_{m'=1}^{M} p(x_n|\theta'_m)p(m')}$$

3. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of missing data from E-step is used instead of the actual missing data.

$$\begin{aligned} \hat{\mu}_m &= \frac{\sum_{n=1}^{N} z_{mn}x_n}{\sum_{n=1}^{N} z_{mn}} \\ \hat{\Sigma}_m &= \frac{\sum_{n=1}^{N} p(m|x_n)(x_n - \hat{\mu}_m)^T(x_n - \hat{\mu}_m)}{\sum_{n=1}^{N} p(m|x_n)} \\ \pi_m &= \frac{1}{N}\sum_{n=1}^{N} p(m|x_n) \end{aligned}$$

Supervised learning of the Bayesian classifiers

Nonparametric methods for density estimation

Nonparametric methods for density estimation

1. Parametric methods assume that samples are drawn from some known distribution (for example Gaussian). But the parameters of this distribution is not known and our goal is to estimate these parameters from the data.
2. The main advantage of the parametric methods is the model is defined up to a small number of parameters and when these parameters are estimated, the whole distribution is known.
3. Methods used to estimate the parameters of the distribution are maximum likelihood estimation and Bayesian estimation.
4. Why nonparametric methods for density estimation?
   - Common parametric forms do not always fit the densities encountered in practice.
   - Most of the classical parametric densities are unimodal, whereas many practical problems involve multi-modal densities.
   - Non-parametric methods can be used with arbitrary distributions without assumption of knowing the forms of the underlying densities.
5. In nonparametric estimation, we assume that similar inputs have similar outputs. This is a reasonable assumption because the world is smooth and functions, whether they are densities, discriminants, or regression functions, change slowly.

Nonparametric methods for density estimation (cont.)

1. Let $X = \{x_1, x_2, \ldots, x_N\}$ be random samples drawn i.i.d. from probability density function $p(x)$.
2. $P_{\mathcal{R}}$ is probability that a vector x will fall in a region $\mathcal{R}$ and is given by

$$P_{\mathcal{R}} = \int_{x \in \mathcal{R}} p(x) dx$$

3. The probability that k of N samples will fall in $\mathcal{R}$ is given by the binomial law.

$$P_{\mathcal{R}}^{(k)} = \left(\begin{array}{c} N \\ k \end{array} \right) P_{\mathcal{R}}^{k} \left(1 - P_{\mathcal{R}}\right)^{N-k}$$

4. The expected value of k is equal to $\mathbb{E}[k] = NP_{\mathcal{R}}$ and MLE for $P_{\mathcal{R}}$ equals to $\frac{k}{N}$.
5. If $p(x)$ is continuous and $\mathcal{R}$ is small enough so that $p(x)$ does not vary significantly in it, then for all $x \in \mathcal{R}$, we can approximate $P_{\mathcal{R}}$ with

$$P_{\mathcal{R}} = \int_{x' \in \mathcal{R}} p(x') dx' \approx p(x)V$$

   V is the volume of $\mathcal{R}$

6. Then the density function can be estimated as

$$p(x) \approx \frac{k/N}{V}$$

Nonparametric methods for density estimation (example)

1. Let x be a univariate feature vector and $\mathcal{R}(x)$ be the region given by

$$\mathcal{R}(x) = \{x' | x' \leq x\}$$

2. The nonparametric estimator for the cumulative distribution function, $P(x)$, at point x is the proportion of sample points that are less than or equal to x.

$$\hat{P}(x) = \frac{|\mathcal{R}(x)|}{N}$$

3. The nonparametric estimate for the density function can be calculated as

$$\hat{p}(x) = \frac{1}{h}\left[\frac{|\mathcal{R}(x+h)| - |\mathcal{R}(x)|}{N}\right]$$

   h is the length of the interval and instances x that fall in this interval are assumed to be close enough.

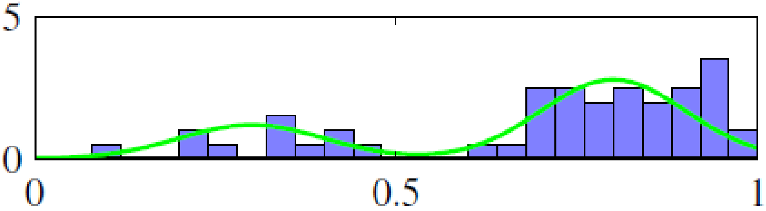
4. Different heuristics are used to determine the instances that are close and their effects on the estimate.

Histogram estimator

1. The oldest and most popular method is the histogram where the input space is divided into equal-sized intervals called bins.
2. Given an origin x_0 and a bin width h, the m^{th} bins denoted by $\mathcal{R}_m(x)$ is the interval $[x_0 + mh, x_0 + (m+1)h)$ for positive and negative integers m and the estimate is given as

$$\hat{p}(x) \quad = \quad \frac{|\mathcal{R}_m(x)|}{Nh}$$

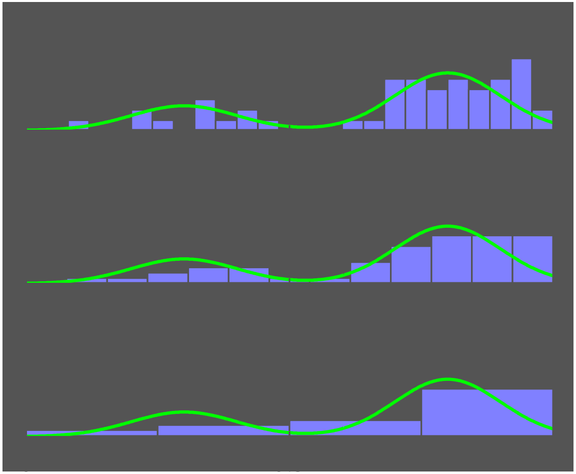
3. In constructing the histogram, we have to choose both an origin and a bin width.



4. The estimate is 0 if no instance falls in a bin and there are discontinuities at bin boundaries.
5. One advantage of the histogram is that once the bin estimates are calculated and stored, we do not need to retain the training set.

Histogram estimator (cont)

1. A histogram density model is dependent on
   - The choice of origin.
   - The choice of bin width.
2. The choice of origin is much less significant than the value of bin width.
3. The choice of origin affects the estimate near boundaries of bins, but it is mainly the bin width that has an effect on the estimate
   - When bins are small, the estimate is spiky.
   - When bins become larger, the estimate becomes smoother.



Histogram estimator (cont)

1. The histogram is useful for visualization of data in one or two dimensions.
2. This technique is unsuited to most density estimation applications because
   - The estimated density has discontinuities due to the bin edge.
   - The histogram approach is not scalable with dimensionality.
3. This density estimation approach teaches us two important lessons:
   - For estimating the probability density at x, we should consider its local neighboring points. The locality defines a natural smoothing parameter. In our example bin width.
   - For obtaining good results, value of the smoothing parameter should be neither too large nor too small.
4. The density estimate has the following form

$$p(x) \approx \frac{K/N}{V}$$

5. We can find density using
   - Fix V and determine K from the data (kernel density).
   - Fix K and determine V from the data (K-nearest-neighbor).

Naive estimator

1. A generalization of histogram (naive estimator), addresses the choice of bin locations.
2. This method adaptively determines the bin locations, thus the origin is eliminated.
3. For bin width h, bin denoted by $\mathcal{R}(x)$ is interval $\left[x - \frac{h}{2}, x + \frac{h}{2}\right)$ and the estimate is

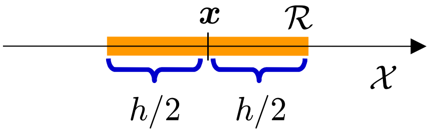
$$\hat{p}(x) = \frac{|\mathcal{R}(x)|}{Nh}$$

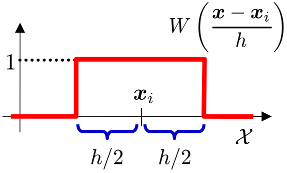
4. This equals to the histogram estimate where x is always at the center of a bin of size h.
5. The estimator can also be written as

$$\hat{p}(x) = \frac{1}{Nh}\sum_{k=1}^{N} w\left(\frac{x - x_k}{h}\right)$$

w is weight function and defined as

$$w(u) = \begin{cases} 1 & \text{if } |u| \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$





Kernel estimator

1. To get a smooth estimate, we use a smooth weight function, called kernel function, and in this context is also called Parzen window.

$$\hat{p}(x) = \frac{1}{Nh}\sum_{i=1}^{N} w\left(\frac{x-x_i}{h}\right)$$

   $w(.)$ is some kernel (window) function and h is the bandwidth (smoothing parameter).

2. The most popular kernel function is Gaussian kernel function with mean 0 and variance 1.

$$w(u) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{u^2}{2}\right)$$

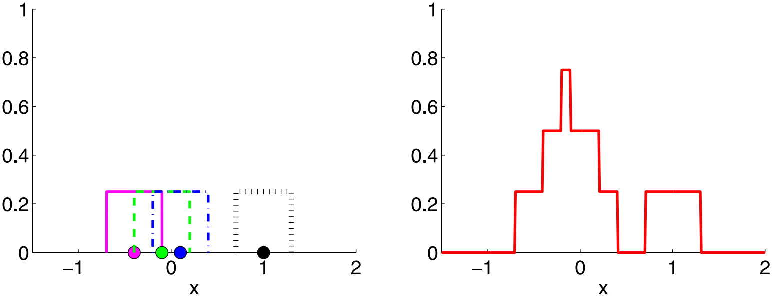
3. Function $w(.)$ determines the shape of influences and h determines the window width.
4. The kernel estimator can be generalized to $D-$dimensional data.

$$\begin{aligned}\hat{p}(x) &= \frac{1}{Nh^D}\sum_{k=1}^{N} w\left(\frac{x-x_k}{h}\right)\\ w(u) &= \left(\frac{1}{\sqrt{2\pi}}\right)^D \exp\left(-\frac{||u||^2}{2}\right)\end{aligned}$$

5. The total number of data points lying in this window (cube) equals to (drive it.)

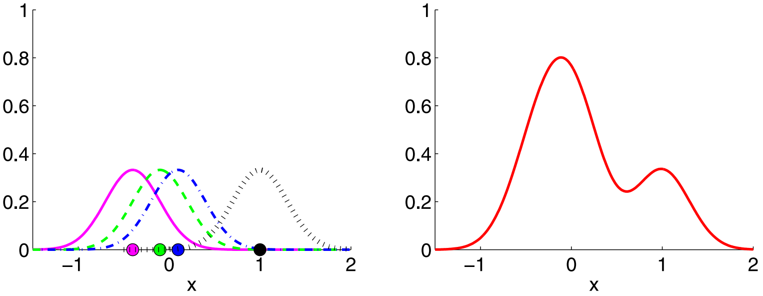
$$k = \sum_{i=1}^{N} w\left(\frac{x-x_i}{h}\right)$$

Kernel estimator



(a) Each Parzen window function

(b) Parzen window estimator

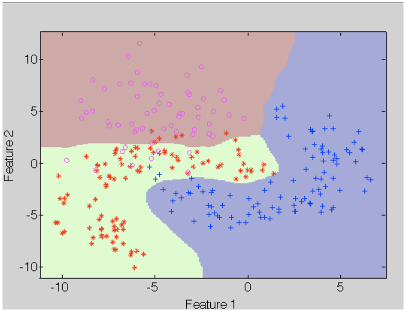


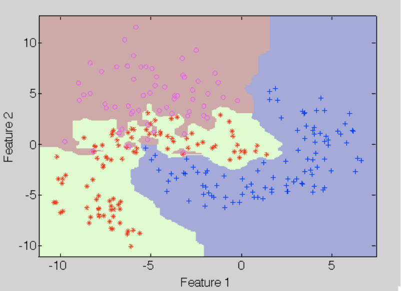
(a) Each Gaussian kernel function

(b) Kernel density estimator

Parzen classifier

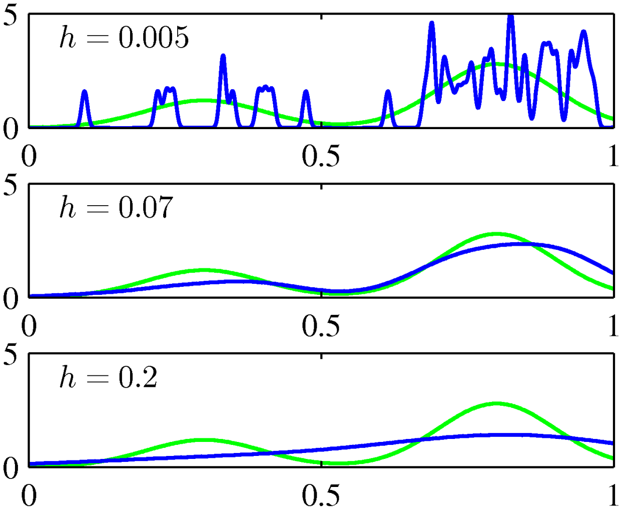
1. In Parzen classifier, the following steps are done.
   - Estimating probability density for each class using a non-parametric approach based on stored training examples.
   - Classifying the test points by the label that corresponds to the maximum density.
2. The decision surface for a parzen classifier depends upon the choice of window function.





$k-$Nearest neighbor estimator

1. One difficulty with the kernel approach is that the parameter h is fixed for all kernels.
2. Large value of h may lead to over-smoothing.
3. Reducing value of h may lead to noisy estimates.
4. The optimal choice of h may be dependent on location within the data space.



$k-$Nearest neighbor estimator (cont.)

1. Instead of fixing h and determining the value of k from the data, we fix the value of k and use the data to find an appropriate value of h.
2. To do this, we consider a small sphere centered on the point x at which we wish to estimate the density $p(x)$ and allow the radius of the sphere to grow until it contains precisely k data points.

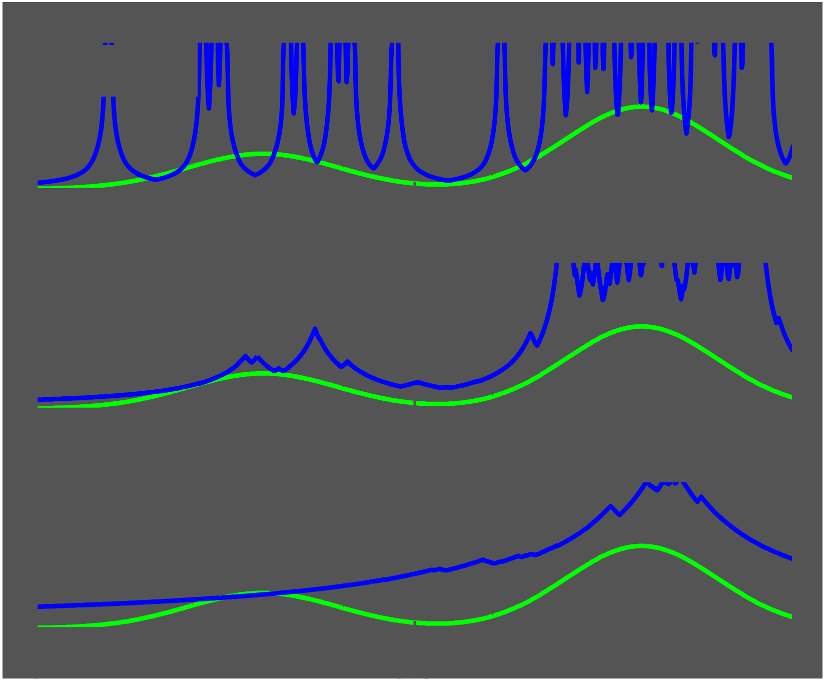
$$\hat{p}(x) = \frac{k}{NV}$$

   V is the volume of the resulting sphere.
3. Value of k determines the degree of smoothing and there is an optimum choice for k that is neither too large nor too small.
4. Note that: The model produced by k nearest neighborhood is not a true density model because the integral over all space diverges.

Theorem

It can be shown that both the K-NN and the kernel density estimators converge to the true probability density in the limit $N \to \infty$ provided V shrinks suitably with N , and K grows with N (Duda and Hart, 1973).

$k-$Nearest neighbor estimator



$k-$Nearest neighbor classifier

1. $k-$NN classifier first uses $k-$NN density estimation for each class and then use Bayes theorem.
2. Let data set with N_i points in class C_i and $\sum_i N_k = N$.
3. To classify a new point x, we draw a sphere centered on x containing precisely k points irrespective of their class.
4. Suppose this sphere has volume V and contains k_i points from class C_i.
5. An estimate of the density associated with each class equals to

$$p(x|C_i) = \frac{k_i}{N_i V}$$

6. The unconditional density is given by

$$p(x) = \frac{k}{NV}$$

7. The class priors equal to

$$p(C_i) = \frac{N_i}{N}$$

8. Combining the above equations using Bayes theorem will results in

$$p(C_i|x) = \frac{p(x|C_i)p(C_i)}{p(x)} = \frac{\frac{k_i}{N_i V}\frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

$k-$Nearest neighbor classifier (cont.)

1. In $k-$NN classifier, the posterior probability of each class equals to

$$p(C_i|x) \quad = \quad \frac{p(x|C_i)p(C_i)}{p(x)} = \frac{\frac{k_i}{N_i V}\frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

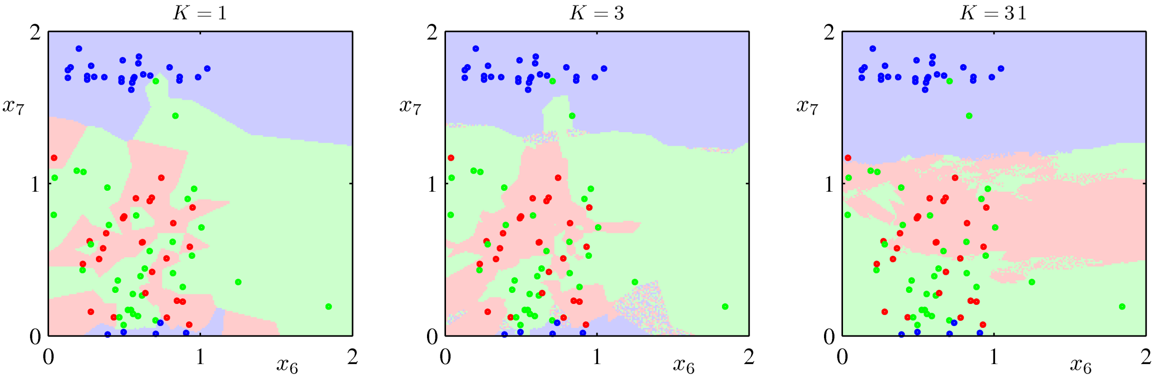
2. For minimizing the misclassification probability, assigning the test point x to the class having the largest $p(C_i|x)$, i.e. the largest value of k_i/k.
3. Thus to classify a new point, we identify the k nearest points from the training data set and then assign the new point to the class having the largest number of representatives amongst this set.
4. The particular case of $k = 1$ is called the nearest-neighbor rule, because a test point is simply assigned to the same class as the nearest point from the training set.

Theorem

When $N \to \infty$, the error rate NN ($k = 1$) classifier is never more than twice the minimum achievable error rate of an optimal classifier.

$k-$Nearest neighbor classifier (example)

1. The parameter k controls the degree of smoothing, i.e. small k produces many small regions of each class and large k leads to a fewer larger regions.



$k-$Nearest neighbor classifier (cont.)

1. $k-$NN classifier relies on a metric or a distance function, between points
2. For all points , x, y, and z, a metric $d(.,.)$ must satisfy the following properties
   - Non–negativity: $d(x,y) \geq 0$.
   - Reflexivity: $d(x,y) = 0 \iff x = y$.
   - Symmetry: $d(x,y) = d(y,x)$.
   - Triangle inequality : $d(x,y) + d(y,z) \geq d(x,z)$.
3. A general class of metrics for $D-$dimensional feature vectors is Minkowski metric (also referred to as L_p-norm)

$$L_p(x,y) \quad = \quad \left(\sum_{i=1}^{D} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

4. When $p = 1$, the metric called Manhattan or city–block distance and is L_1-norm .
5. When $p = 2$, the metric called Euclidean distance and is L_2-norm .
6. When $p = \infty$, the $L_\infty-$norm is the maximum of distance along individual coordinates axes.

$$L_\infty(x,y) \quad = \quad \max_i |x_i - y_i|$$

Conclusions

1. Both the $k-$NN method, and the kernel density estimator, require the entire training data set to be stored, leading to expensive computation if the data set is large.
2. This effect can be offset, at the expense of some additional one-off computation, by constructing tree-based search structures such as KD-tree to allow(approximate) nearest neighbors to be found efficiently without doing an exhaustive search of the data set.
3. These nonparametric methods are still severely limited.
4. On the other hand, simple parametric models are very restricted in terms of the forms of distribution that they can represent.
5. We therefore need to find density models that are very flexible and yet for which the complexity of the models can be controlled independently of the size of the training set.

Naive Bayes classifier

Naive Bayes classifier

1. Bayesian classifiers estimate posterior probabilities based likelihood, prior, and evidence.
2. These classifiers first estimate $p(x|C_i)$ and $p(C_i)$ and then classify the given instance.
3. How much training data will be required to obtain reliable estimates of these distributions?
4. Consider the number of parameters that must be estimated when $C = 2$ and x is a vector of D boolean features.
5. In this case, we need to estimate a set of parameters

$$\theta_{ij} \quad = \quad p(x_i|C_j)$$

6. Index i takes on 2^D possible values, and j takes on 2 possible values.
7. Therefore, we will need to estimate exactly $2(2^D - 1)$ of such θ_{ij} parameters.
8. Unfortunately, this corresponds to two distinct parameters for each of the distinct instances in the instance space for x.
9. In order to obtain reliable estimates of each of these parameters, we will need to observe each of these distinct instances multiple times! This is clearly unrealistic in most practical learning domains.
10. For example, if x is a vector containing 30 boolean features, then we will need to estimate more than 3 billion parameters.

Naive Bayes classifier (cont.)

1. Given the intractable sample complexity for learning Bayesian classifiers, we must look for ways to reduce this complexity.
2. The Naive Bayes classifier does this by making a conditional independence assumption that dramatically reduces the number of parameters to be estimated when modelling $P(x_i|C_j)$, from our original $2(2^D - 1)$ to just $2D$.

Definition (Conditional Independence)

Given random variables x, y and z, we say x is conditionally independent of y given z, if and only if the probability distribution governing x is independent of the value of y given z; that is

$$p(x_i, y_j|z_k) = p(x_i|z_k)p(y_j|z_k) \qquad \forall i, j, k$$

3. The Naive Bayes algorithm is a classification algorithm based on Bayes rule, that assumes the features $x_1, x_2, \ldots, x_D$ are all conditionally independent of one another, given the class label C_i. Thus we have

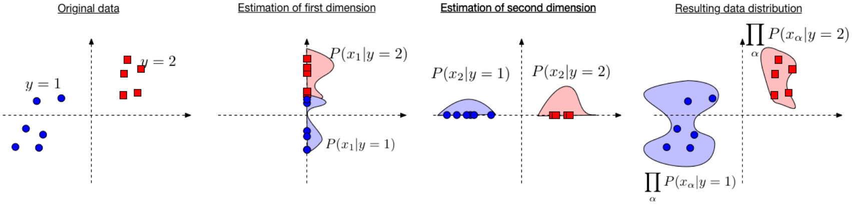
$$p(x_1, x_2, \ldots, x_D|C_j) = \prod_{i=1}^{D} p(x_i|C_j)$$

4. Note that when C and the x_i are boolean variables, we need only $2D$ parameters to define $p(x_{ik}|C_j)$ for the necessary i, j, and k.

Naive Bayes classifier (cont.)

1. The Bayes classifier can be defined as

$$
\begin{aligned}
h(\mathbf{x}) &= \underset{y}{\operatorname{argmax}}\, P(y|\mathbf{x}) \\
&= \underset{y}{\operatorname{argmax}}\; \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\
&= \underset{y}{\operatorname{argmax}}\; P(\mathbf{x}|y)P(y) \\
&= \underset{y}{\operatorname{argmax}}\; \prod_{k=1}^{D} P(x_k|y)P(y) \\
&= \underset{y}{\operatorname{argmax}}\; \sum_{k=1}^{D} \log(P(x_k|y)) + \log(P(y))
\end{aligned}
$$



Original data
$y = 2$
$y = 1$
Estimation of first dimension
$P(x_1|y = 2)$
$P(x_1|y = 1)$
Estimation of second dimension
$P(x_2|y = 1)$
$P(x_2|y = 2)$
Resulting data distribution
$\prod_\alpha P(x_\alpha|y = 2)$
$\prod_\alpha P(x_\alpha|y = 1)$

Credit: K. Weinberger

Naive Bayes for discrete-valued inputs

1. When each D input feature x_i takes on J possible discrete values, and C is a discrete variable taking on M possible values, the learning task is to estimate two sets of parameters.

$$\begin{aligned} \theta_{ijk} &= p(x_i = x'_{ij} | C = C_k) \qquad \text{Feature } x_i \text{ takes value } x_{ij} \\ \pi_k &= p(C = C_k) \end{aligned}$$

2. We can estimate these parameters using either ML estimates or Bayesian/MAP estimates.

$$\theta_{ijk} = \frac{|x_i = x'_{ij} \bigwedge C = C_k|}{|C_k|}$$

3. This maximum likelihood estimate sometimes results in θ estimates of zero, if the data does not happen to contain any training examples satisfying the condition in the numerator. To avoid this, it is common to use a smoothed estimate.

$$\theta_{ijk} = \frac{|x_i = x'_{ij} \bigwedge C = C_k| + l}{|C_k| + lJ}$$

4. Value of l determines the strength of this smoothing.
5. Maximum likelihood estimates for π_k are

$$\pi_k = \frac{|C_k|}{N} = \frac{|C_k| + l}{N + lM}$$

Naive Bayes for continuous inputs

1. When features are continuous, we must choose some other way to represent the distributions $p(x_i|C_k)$.
2. One common approach is to assume that for each possible C_k, the distribution of each feature x_i is Gaussian defined by mean and variance specific to x_i and C_k.
3. In order to train such a Naive Bayes classifier, we must therefore estimate the mean and standard deviation of each of these distributions.

$$\begin{aligned}\mu_{ik} &= E[x_i|C_k] \\ \sigma_{ik}^2 &= E[(x_{ik} - \mu_{ik})^2|C_k]\end{aligned}$$

4. We must also estimate the prior on C.

$$\pi_k = p(C = C_k)$$

5. We can use either maximum likelihood estimates (MLE) or maximum a posteriori (MAP) estimates for these parameters.
6. The maximum likelihood estimator for μ_{ik} is

$$\hat{\mu}_{ik} = \frac{\sum_j x_{ij}\delta(t_j = C_k)}{\sum_j \delta(t_j = C_k)}$$

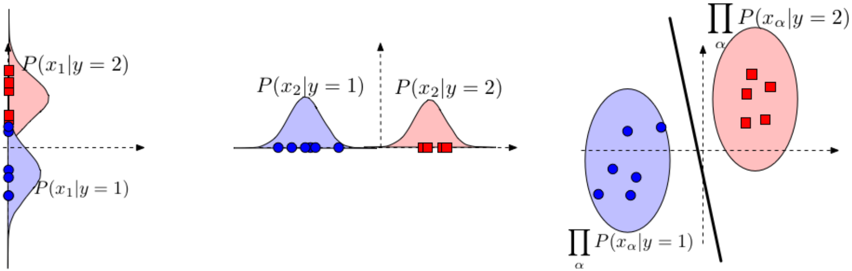
7. The maximum likelihood estimator for σ_{ik}^2 is

$$\hat{\sigma}_{ik}^2 = \frac{\sum_j (x_{ij} - \hat{\mu}_{ik})^2\delta(t_j = C_k)}{\sum_j \delta(t_j = C_k)}$$

Naive Bayes is a linear classifier

1. Let $y_i \in \{-1, +1\}$ and features are multinomial.
2. As an exercise, show that

$$h(\mathbf{x}) = \operatorname*{argmax}_y \; P(y) \prod_{k=1}^{D} P(x_k \mid y) = \operatorname{sign}(\mathbf{w}^\top \mathbf{x} + b)$$


$P(x_1|y=2)$
$P(x_1|y=1)$
$P(x_2|y=1)$
$P(x_2|y=2)$
$\prod_\alpha P(x_\alpha|y=2)$
$\prod_\alpha P(x_\alpha|y=1)$

Credit: K. Weinberger

Reading

Readings

1. Sections 1.5, 2.3, 2.5, & 4.2 of Pattern Recognition and Machine Learning Book (Bishop 2006).
2. Chapter 5 of Probabilistic Machine Learning: An introduction (Murphy 2022).

References i

- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
- Murphy, Kevin P. (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.

Questions?