به نام یگانه معبود بخشنده مهربان

مبانی یادگیری ماشین

Machine Learning Foundations

گروه هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

ترم اول سال تحصیلی ۰۲-۰۳

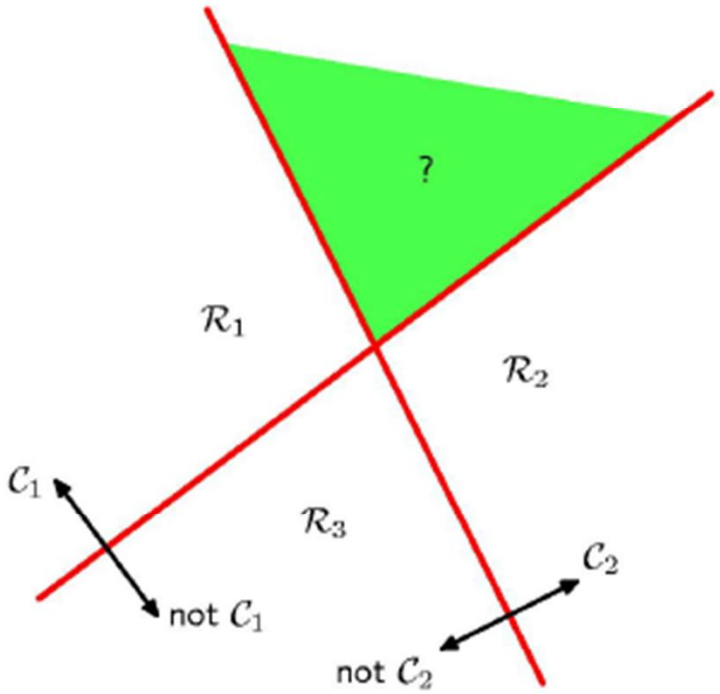
ارائه دهنده : پیمان ادیبی

دسته بندی چند دسته ای

Multi-class Classification

روش یکی در مقابل بقیه برای $K>2$ دسته

- First idea: Use $K-1$ classifiers, each solving a two class problem of separating point in a class C_k from points not in the class.
- Known as **1 vs all** or **1 vs the rest** classifier

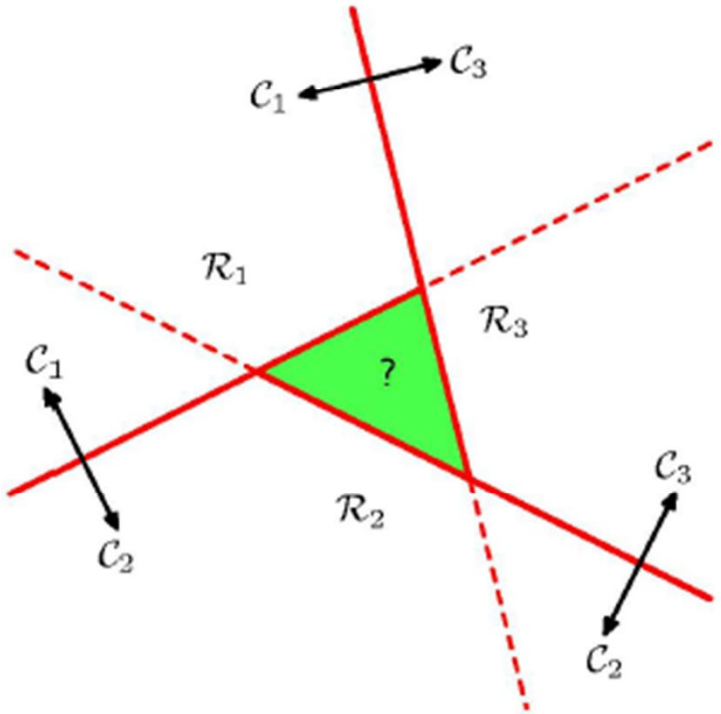


- PROBLEM: More than one good answer for green region!

✓414

روش یکی در مقابل یکی برای $K>2$ دسته

- Another simple idea: Introduce $K(K-1)/2$ two-way classifiers, one for each possible pair of classes
- Each point is classified according to majority vote amongst the disc. func.
- Known as the **1 vs 1 classifier**



- PROBLEM: Two-way preferences need not be transitive

✓4

توابع تمایز خطی برای هر دسته

- We can avoid these problems by considering a single K-class discriminant comprising K functions of the form

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k,0}$$

and then assigning a point $\mathbf{x}$ to class C_k if

$$\forall j \neq k \qquad y_k(\mathbf{x}) > y_j(\mathbf{x})$$

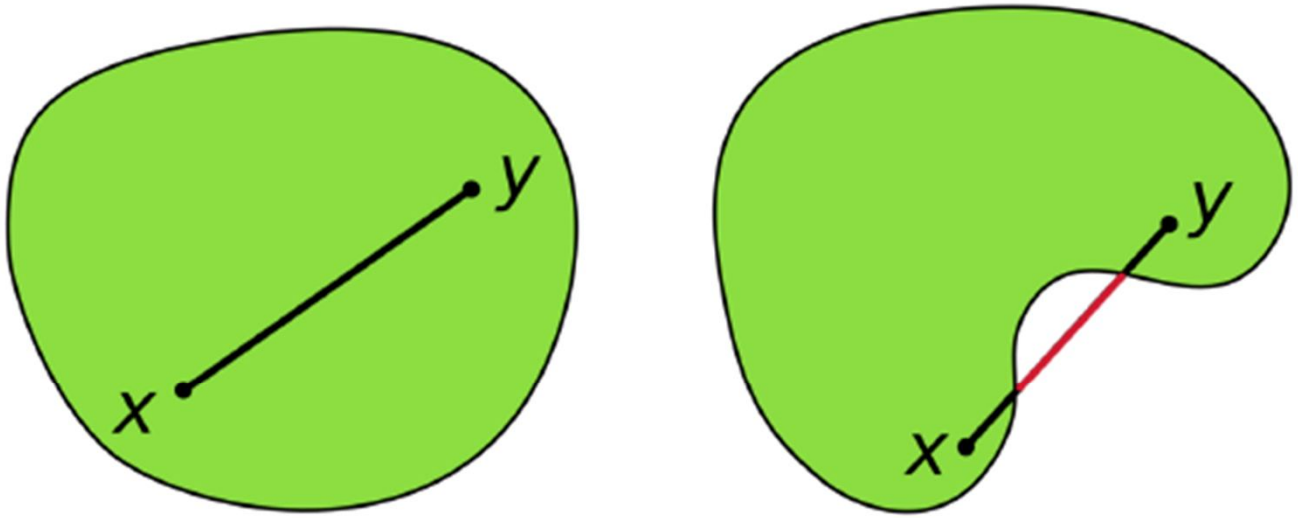
- Note that $\mathbf{w}_k^T$ is now a vector, not the k-th coordinate
- The decision boundary between class C_j and class C_k is given by $y_j(\mathbf{x}) = y_k(\mathbf{x})$, and thus it's a $(D-1)$ dimensional hyperplane defined as

$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0$$

- What about the binary case? Is this different?
- What is the shape of the overall decision boundary?

توابع تمایز خطی برای هر دسته

- The decision regions of such a discriminant are always **singly connected** and **convex**
- In Euclidean space, an object is **convex** if for every pair of points within the object, every point on the straight line segment that joins the pair of points is also within the object

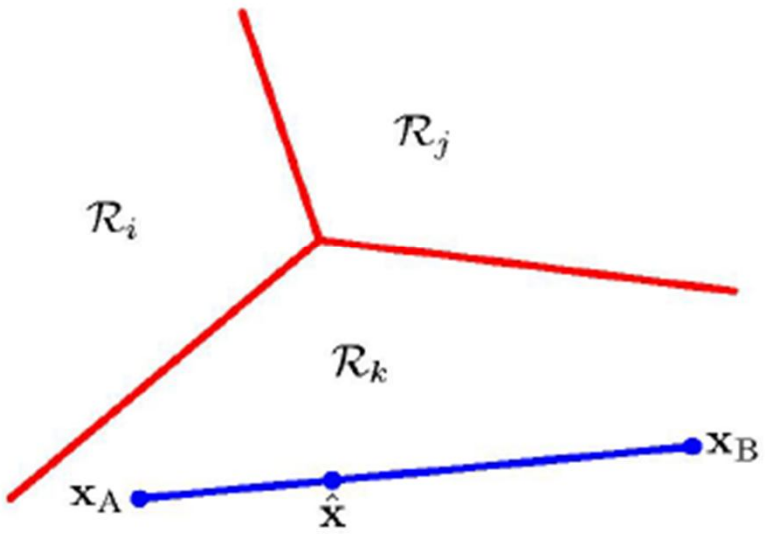


- Which object is convex?

تحدب نواحی در روش توابع تمایز خطی

- The decision regions of such a discriminant are always **singly connected** and **convex**
- Consider 2 points $\mathbf{x}_A$ and $\mathbf{x}_B$ that lie inside decision region R_k
- Any convex combination $\hat{\mathbf{x}}$ of those points also will be in R_k

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1 - \lambda)\mathbf{x}_B$$



تحدب نواحی در روش توابع تمایز خطی - اثبات

- A convex combination point, i.e., $\lambda \in [0, 1]$

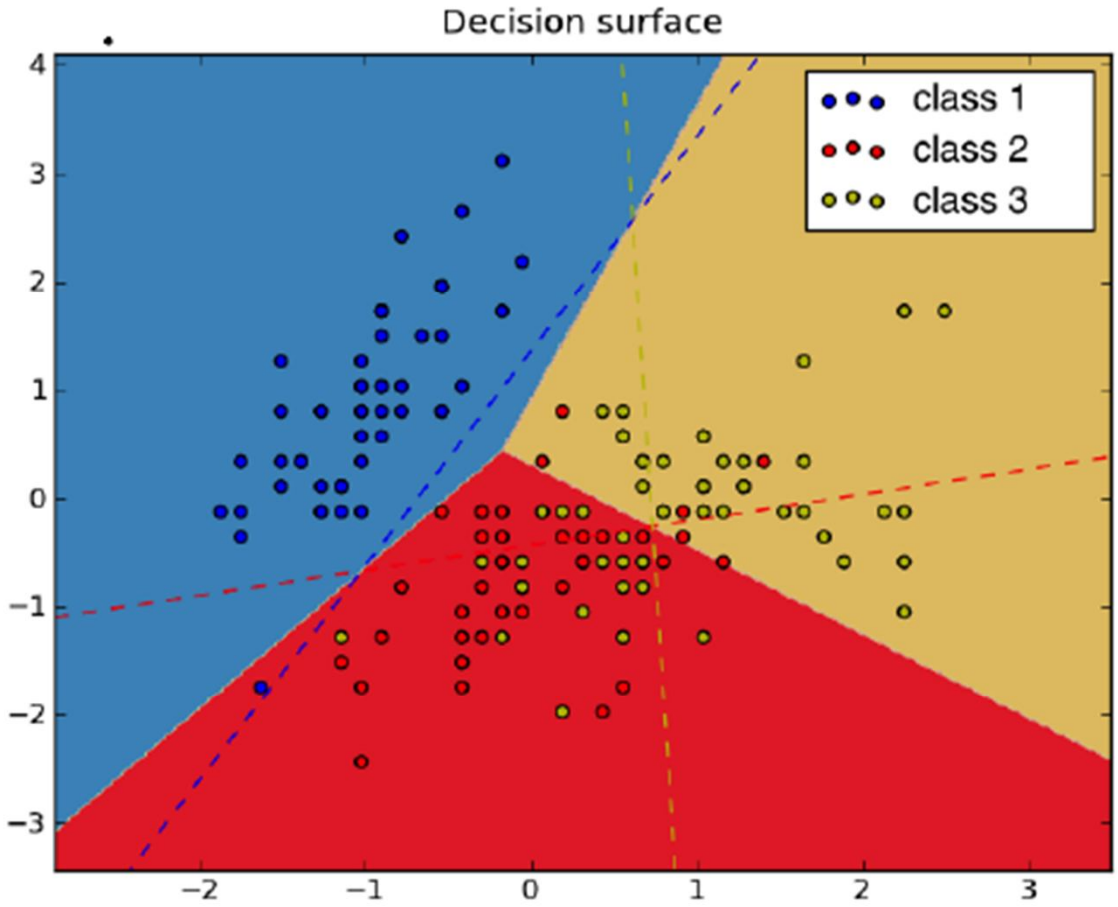
$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1 - \lambda)\mathbf{x}_B$$

- From the linearity of the classifier $y(\mathbf{x})$

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1 - \lambda) y_k(\mathbf{x}_B)$$

- Since $\mathbf{x}_A$ and $\mathbf{x}_B$ are in R_k, it follows that $y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A)$, $y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$, $\forall j \neq k$
- Since λ and $1 - \lambda$ are positive, then $\hat{\mathbf{x}}$ is inside R_k
- Thus R_k is singly connected and convex

نواحی در روش توابع تمایز خطی - مثال



رویکردهای دسته بندی:
تمایزی در برابر تولیدی

Discriminative vs Generative Classification

رویکردهای دسته بندی

- Given inputs x and classes y we can do classification in several ways. How?

(features)

$\mathbf{x}$

e.g:
- height
- weight
- color

(class label)

y

elefant

bear

رویکرد تمایزی در دسته بندی

- **Discriminative** classifiers try to either:
  - learn mappings directly from the space of inputs $\mathcal{X}$ to class labels $\{0, 1, 2, \ldots, K\}$

(features)

$\mathbf{x}$

e.g:
- height
- weight
- color

(class label)

y

elefant

bear

For example:

- Linear regression (as a classifier)
- Neural Networks

رویکرد تمایزی در دسته بندی

- **Discriminative** classifiers try to either:
  - or try to learn $p(y|\mathbf{x})$ directly

(features)

$\mathbf{x}$

e.g:
- height
- weight
- color

(class probability)

$p(y|\mathbf{x})$

elefant

bear

For example:

- Logistic Regression

رویکرد تولیدی در دسته بندی

How about this approach: build a model of "how data for a class looks like"

- **Generative** classifiers try to model $p(\mathbf{x}|y)$
- Classification via Bayes rule (thus also called Bayes classifiers)

(prob. of features given label) (class label)

$p(\mathbf{x}|y)$ ← y

e.g:
- height
- weight
- color

elefant

bear

Tries to model:
- How does data look like for a class?

Classification (How?)

رویکرد تولیدی در برابر تمایزی

Two approaches to classification:

- Discriminative classifiers estimate parameters of decision boundary/class separator directly from labeled examples
  - learn $p(y|\mathbf{x})$ directly (logistic regression models)
  - learn mappings from inputs to classes (least-squares, neural nets)

- Generative approach: model the distribution of inputs characteristic of the class (Bayes classifier)
  - Build a model of $p(\mathbf{x}|y)$
  - Apply Bayes Rule

دسته بند بیز

- Aim to diagnose whether patient has diabetes: classify into one of two classes (yes C=1; no C=0)
- Run battery of tests on the patients, get $\mathbf{x}$ for each patient
- Given patient's results: $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T$ we want to compute class probabilities using Bayes Rule:

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)p(C)}{p(\mathbf{x})}$$

- More formally

$$\text{posterior} = \frac{\text{Class likelihood} \times \text{prior}}{\text{Evidence}}$$

- How can we compute $p(\mathbf{x})$ for the two class case?

$$p(\mathbf{x}) = p(\mathbf{x}|C=0)p(C=0) + p(\mathbf{x}|C=1)p(C=1)$$

- To compute $p(C|\mathbf{x})$ we need: $p(\mathbf{x}|C)$ and $p(C)$

مثال: دسته بندی دیابت

- Let's start with the simplest case where the input is only 1-dimensional, for example: white blood cell count (this is our x)
- We need to choose a probability distribution $p(x|C)$ that makes sense

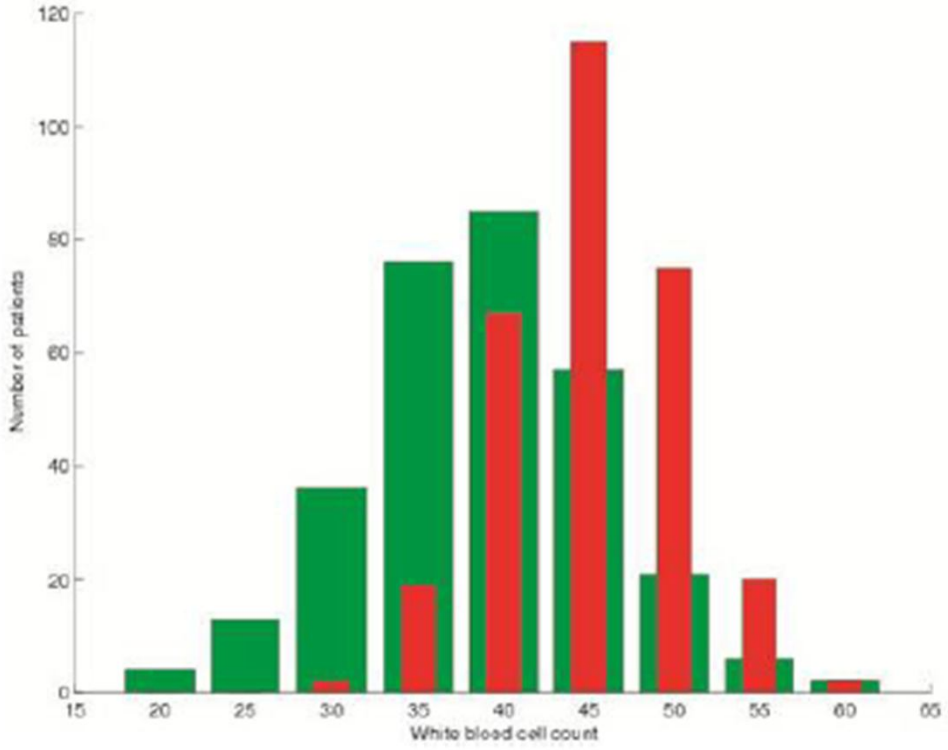


Figure: Our example (showing counts of patients for input value): What distribution to choose?

تحلیل تمایز گاسی (دسته بند بیز گاسی)

- Our first generative classifier assumes that $p(\mathbf{x}|y)$ is distributed according to a multivariate normal (Gaussian) distribution
- This classifier is called Gaussian Discriminant Analysis
- Let's first continue our simple case when inputs are just 1-dim and have a Gaussian distribution:

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu_C)^2}{2\sigma_C^2}\right)$$

with $\mu \in \Re$ and $\sigma^2 \in \Re^+$

- Notice that we have different parameters for different classes
- How can I fit a Gaussian distribution to my data?

تخمین بیشترین شباهت برای توزیع دسته گاسی

- Let's assume that the class-conditional densities are Gaussian

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu_C)^2}{2\sigma_C^2}\right)$$

  with $\mu \in \Re$ and $\sigma^2 \in \Re^+$

- How can I fit a Gaussian distribution to my data?
- We are given a set of training examples $\{x^{(n)}, t^{(n)}\}_{n=1,\cdots,N}$ with $t^{(n)} \in \{0, 1\}$ and we want to estimate the model parameters $\{(\mu_0, \sigma_0), (\mu_1, \sigma_1)\}$
- First divide the training examples into two classes according to $t^{(n)}$, and for each class take all the examples and fit a Gaussian to model $p(x|C)$
- Let's try maximum likelihood estimation (MLE)

تخمین بیشترین شباهت برای توزیع دسته گاسی

(note: we are dropping subscript C for simplicity of notation)

- We assume that the data points that we have are independent and identically distributed

$$p(x^{(1)}, \cdots, x^{(N)}|C) = \prod_{n=1}^{N} p(x^{(n)}|C) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)$$

- Now we want to maximize the likelihood, or minimize its negative (if you think in terms of a loss)

$$\begin{aligned} \ell_{log-loss} &= -\ln p(x^{(1)}, \cdots, x^{(N)}|C) = -\ln\left(\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)\right) \\ &= \sum_{n=1}^{N} \ln(\sqrt{2\pi}\sigma) + \sum_{n=1}^{N} \frac{(x^{(n)}-\mu)^2}{2\sigma^2} = \frac{N}{2}\ln\left(2\pi\sigma^2\right) + \sum_{n=1}^{N} \frac{(x^{(n)}-\mu)^2}{2\sigma^2} \end{aligned}$$

- How do we minimize the function?

✓476

تخمین بیشترین شباهت برای توزیع دسته گاسی

- In summary, we can compute the parameters of a Gaussian distribution in closed form for each class by taking the training points that belong to that class

MLE estimates of parameters for a Gaussian distribution:

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N} (x^{(n)} - \mu)^2$$

احتمال پسین دسته ها

- We now have $p(x|C)$
- In order to compute the posterior probability:

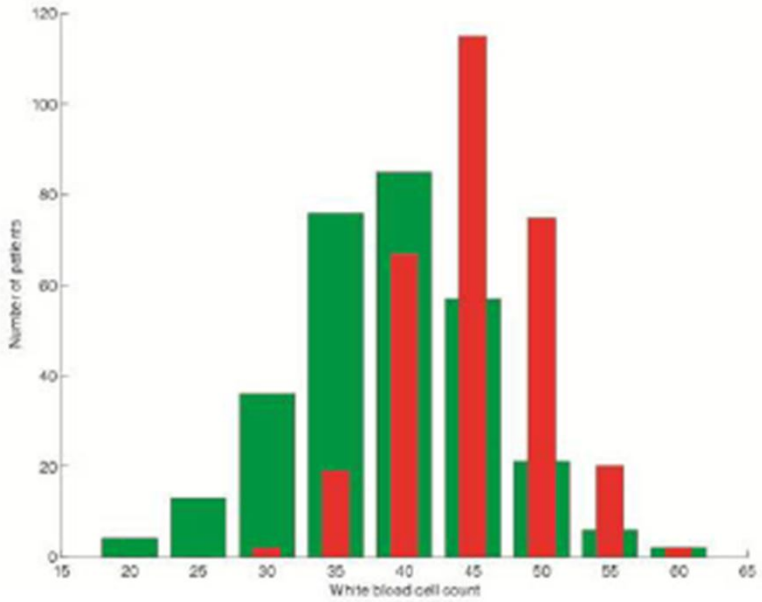
$$p(C|x) = \frac{p(x|C)p(C)}{p(x)}$$

$$= \frac{p(x|C)p(C)}{p(x|C=0)p(C=0) + p(x|C=1)p(C=1)}$$

given a new observation, we still need to compute the **prior**

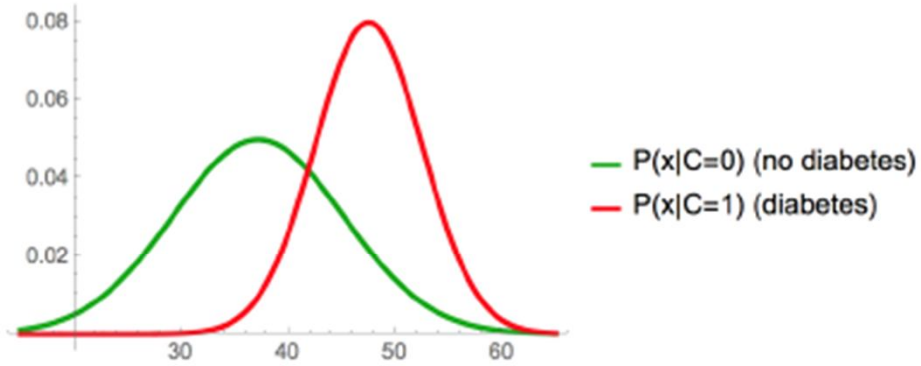
- Prior: In the absence of any observation, what do I know about the problem?

احتمال پسین دسته ها – مثال دیابت



- Doctor has a prior $p(C = 0) = 0.8$, how?
- A new patient comes in, the doctor measures $x = 48$
- Does the patient have diabetes?

احتمال پسین دسته ها – مثال دیابت



- Compute $p(x = 48|C = 0)$ and $p(x = 48|C = 1)$ via our estimated Gaussian distributions
- Compute posterior $p(C = 0|x = 48)$ via Bayes rule using the prior (how can we get $p(C = 1|x = 48)$?)
- How can we decide on diabetes/non-diabetes?

دسته بند بیز – تصمیم گیری

- Use Bayes classifier to classify new patients (unseen test examples)
- Simple Bayes classifier: estimate posterior probability of each class
- What should the decision criterion be?
- The optimal decision is the one that minimizes the expected number of mistakes

دسته بند بیز – تصمیم گیری بنا بر خطر مشروط

- **Conditional risk:**

$$
\begin{aligned}
R(y|\mathbf{x}) &= \sum_{c=1}^{C} L(y(\mathbf{x}), t) p(t = c|\mathbf{x}) \\
&= 0 \cdot p(t = y(\mathbf{x})|\mathbf{x}) + 1 \cdot \sum_{c \neq y} p(t = c|\mathbf{x}) \\
&= \sum_{c \neq y} p(t = c|\mathbf{x}) = 1 - p(t = y(\mathbf{x})|\mathbf{x})
\end{aligned}
$$

- To minimize conditional risk given x, the classifier must decide

$$y(\mathbf{x}) = arg\max_{c} p(t = c|x)$$

- This is the best possible classifier in terms of generalization, i.e. expected misclassification rate on new examples.

✓340

دسته بند بیز ساده

Naïve Bayes Classifier

دسته بند بیز گاسی – ورودی چند بعدی

- Gaussian Discriminant Analysis in its general form assumes that $p(\mathbf{x}|t)$ is distributed according to a multivariate normal (Gaussian) distribution
- Multivariate Gaussian distribution:

$$p(\mathbf{x}|t=k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left[-(\mathbf{x}-\mu_k)^T \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right]$$

  where $|\Sigma_k|$ denotes the determinant of the matrix, and d is dimension of $\mathbf{x}$
- Each class k has associated mean vector μ_k and covariance matrix Σ_k
- Typically the classes share a single covariance matrix Σ ("share" means that they have the same parameters; the covariance matrix in this case): $\Sigma = \Sigma_1 = \cdots = \Sigma_k$

داده چند بعدی

- Multiple measurements (sensors)
- d inputs/features/attributes
- N instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

پارامترهای چند بعدی

- Mean

$$\mathbb{E}[\mathbf{x}] = [\mu_1, \cdots, \mu_d]^T$$

- Covariance

$$\Sigma = Cov(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mu)^T(\mathbf{x} - \mu)] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

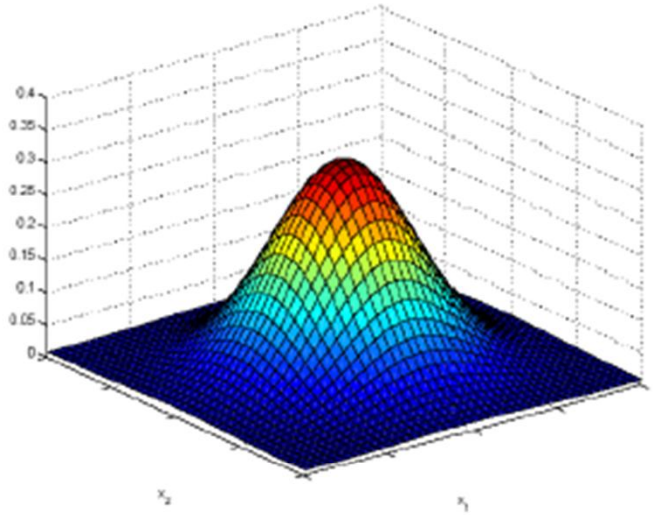
- Correlation $= Corr(\mathbf{x})$ is the covariance divided by the product of standard deviation

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

توزیع گاسی چند بعدی

- $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$, a Gaussian (or normal) distribution defined as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right]$$



- Mahalanobis distance $(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k)$ measures the distance from $\mathbf{x}$ to μ in terms of Σ
- It normalizes for difference in variances and correlations

توزیع نرمال دو بعدی

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Sigma = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Sigma = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

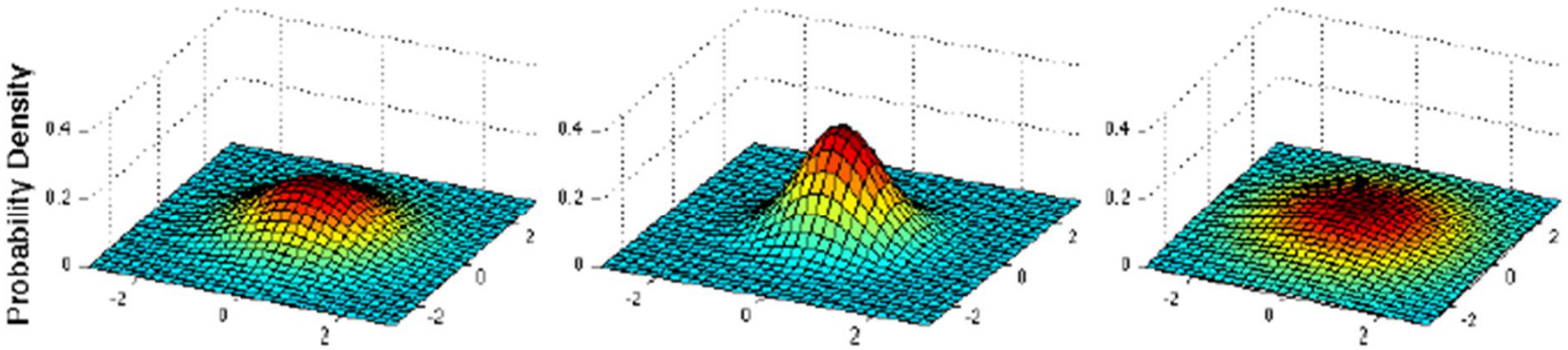


Figure: Probability density function

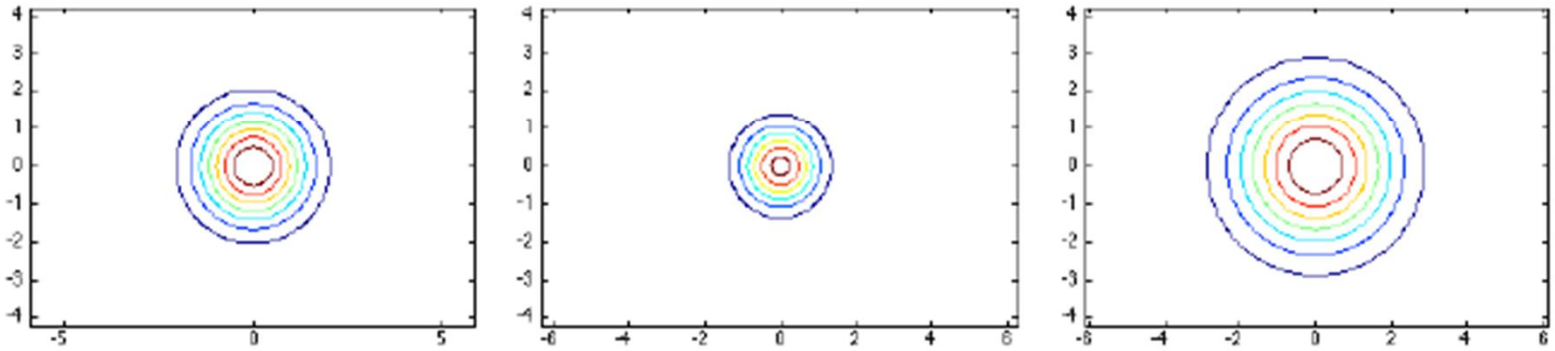


Figure: Contour plot of the pdf

توزیع نرمال دو بعدی

$var(x_1) = var(x_2)$ $var(x_1) > var(x_2)$ $var(x_1) < var(x_2)$

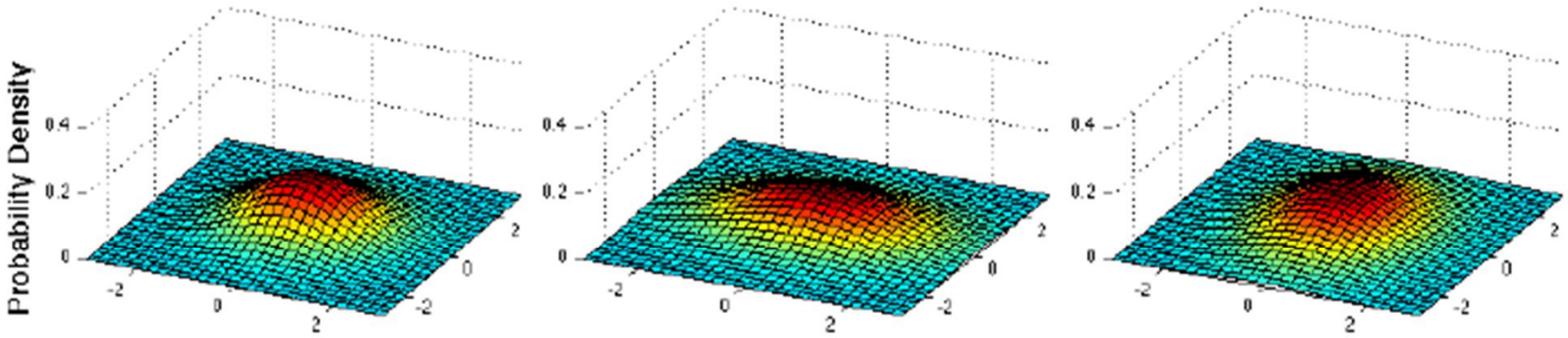


Figure: Probability density function

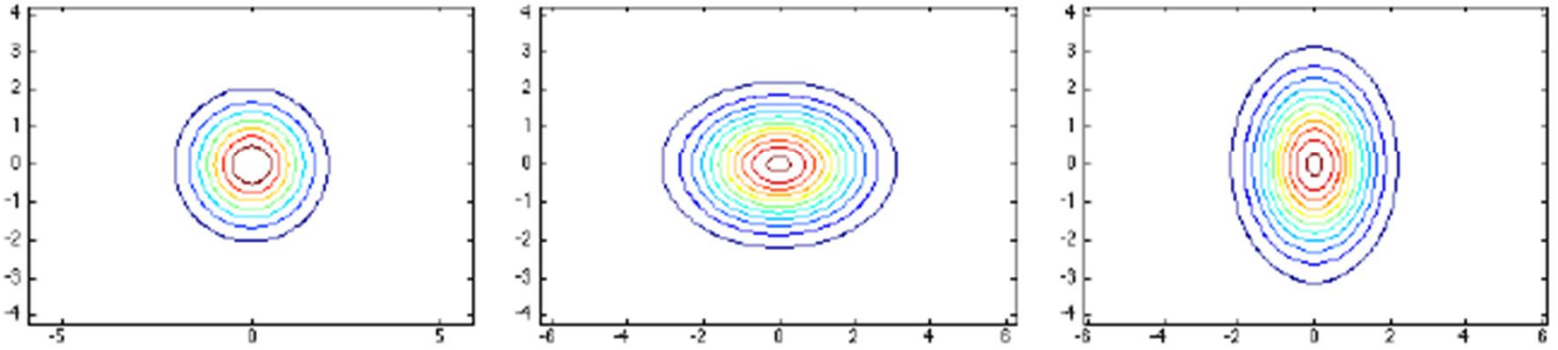


Figure: Contour plot of the pdf

توزیع نرمال دو بعدی

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

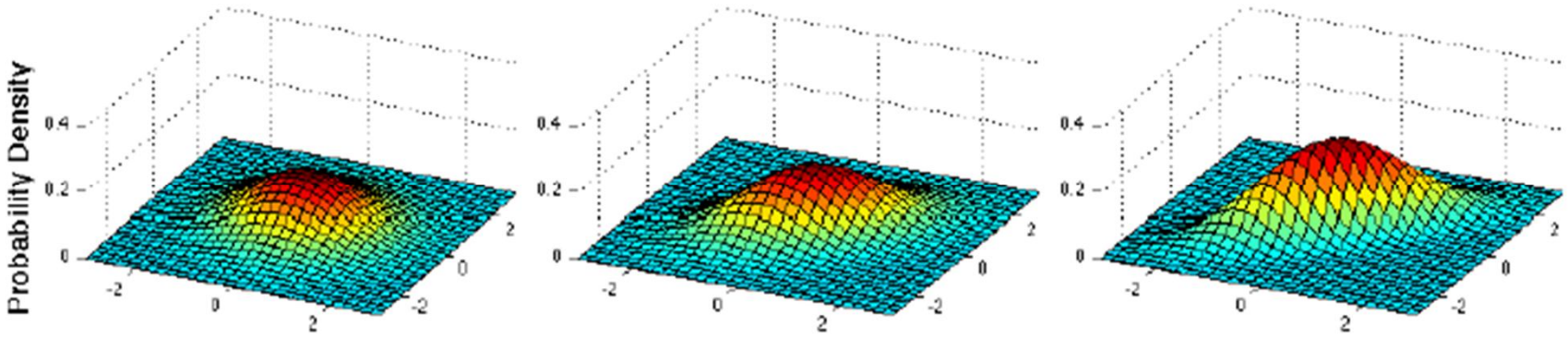


Figure: Probability density function

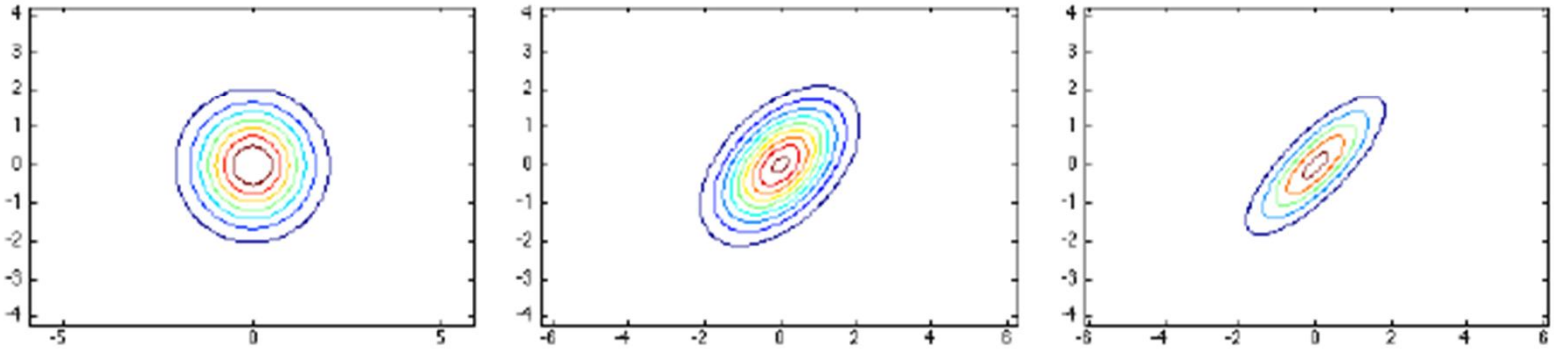


Figure: Contour plot of the pdf

توزیع نرمال دو بعدی

$Cov(x_1, x_2) = 0$ $Cov(x_1, x_2) > 0$ $Cov(x_1, x_2) < 0$

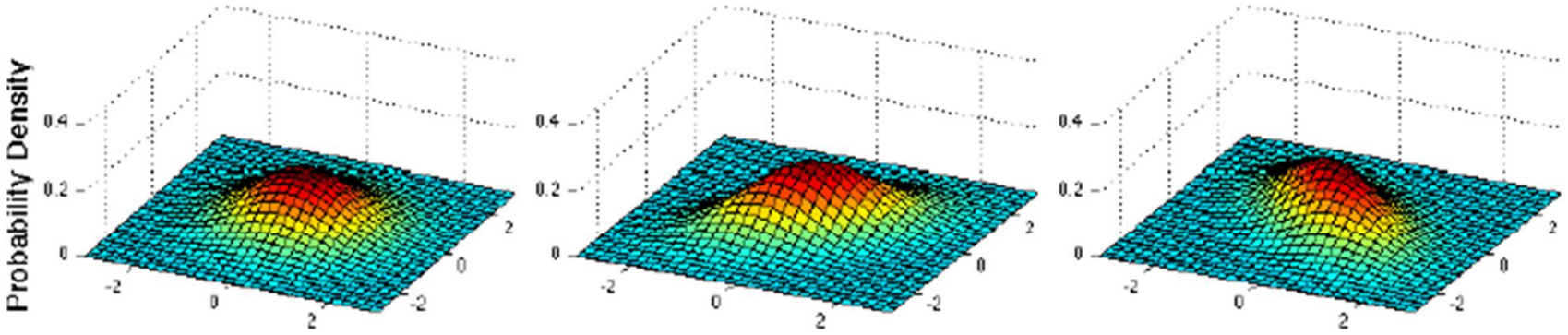


Figure: Probability density function

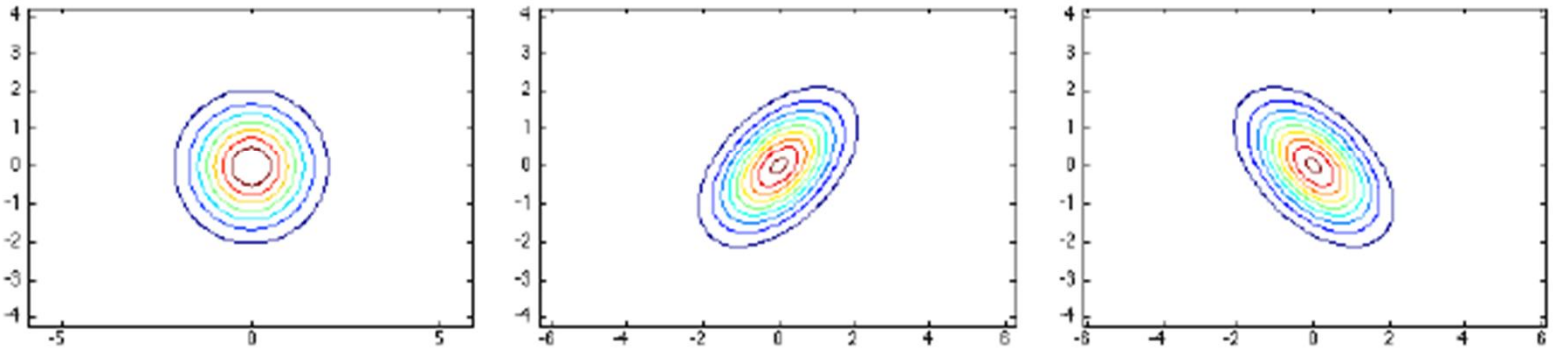
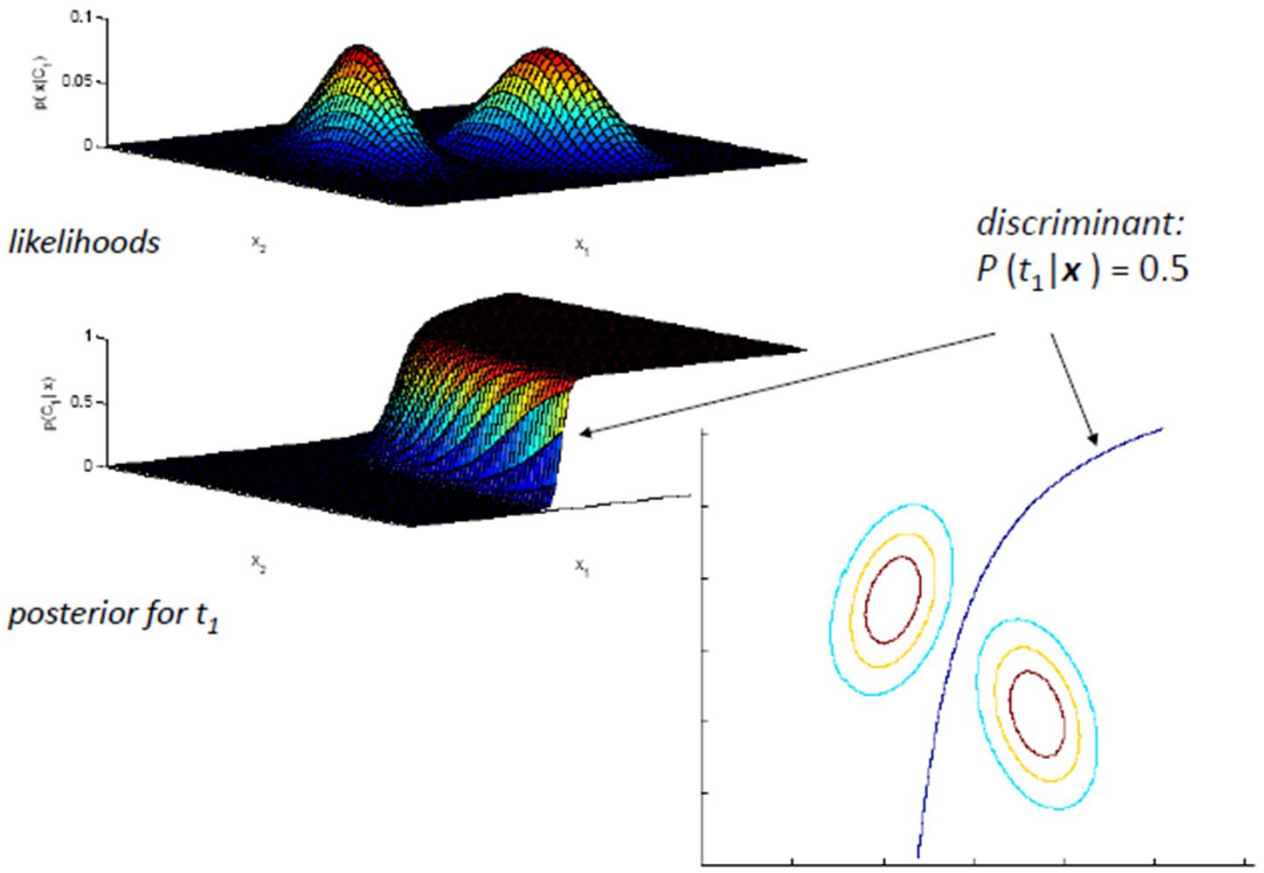


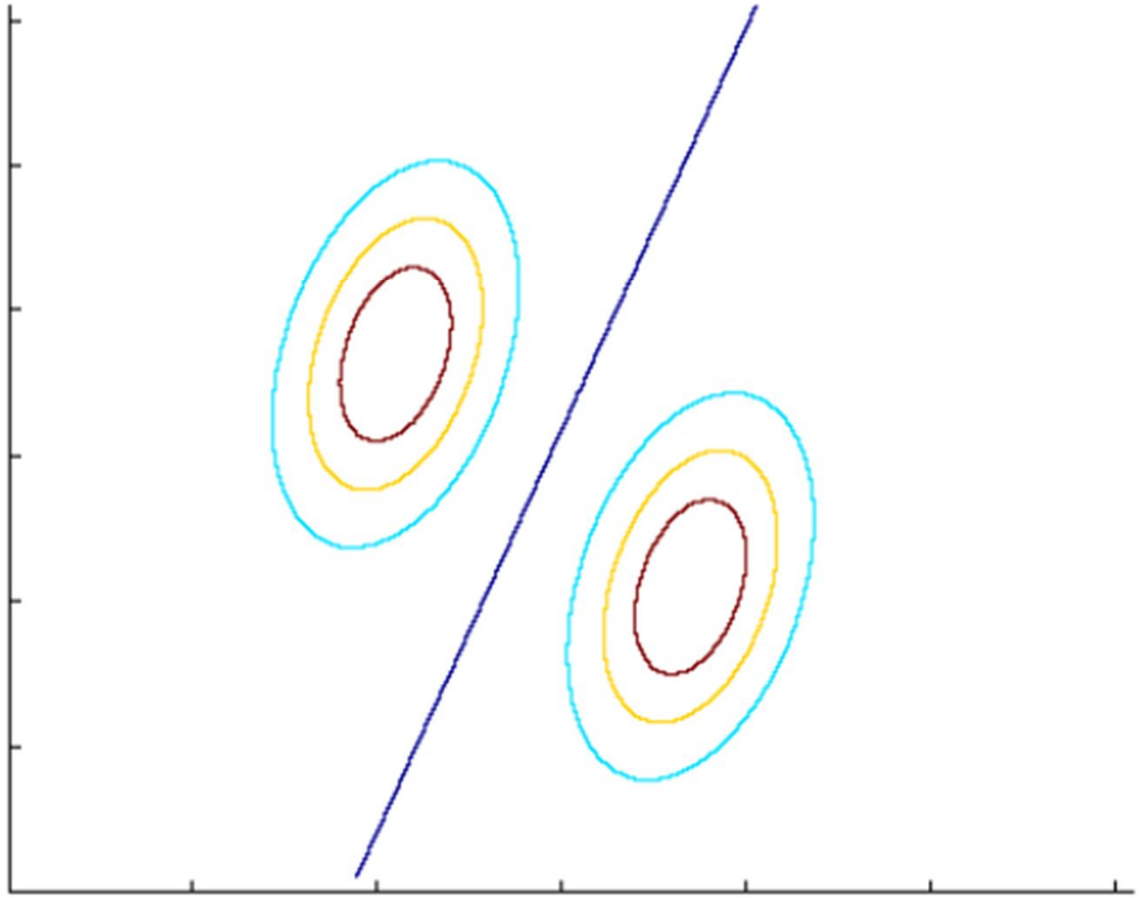
Figure: Contour plot of the pdf

✓400

مرز تصمیم


likelihoods
discriminant:
$P(t_1|\boldsymbol{x}) = 0.5$
posterior for t_1

مرز تصمیم برای ماتریس کوواریانس اشتراکی



دسته بند بیز گاسی – یادگیری

- Learn the parameters using maximum likelihood

$$\begin{aligned} \ell(\phi, \mu_0, \mu_1, \Sigma) &= -\log \prod_{n=1}^{N} p(\mathbf{x}^{(n)}, t^{(n)} | \phi, \mu_0, \mu_1, \Sigma) \\ &= -\log \prod_{n=1}^{N} p(\mathbf{x}^{(n)} | t^{(n)}, \mu_0, \mu_1, \Sigma) p(t^{(n)} | \phi) \end{aligned}$$

- What have we assumed?

دسته بند بیز گاسی – تخمین بیشترین شباهت

- Assume the prior is Bernoulli (we have two classes)

$$p(t|\phi) = \phi^t(1-\phi)^{1-t}$$

- You can compute the ML estimate in closed form

$$\begin{aligned}
\phi &= \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}[t^{(n)}=1] \\
\mu_0 &= \frac{\sum_{n=1}^{N}\mathbb{1}[t^{(n)}=0]\cdot \mathbf{x}^{(n)}}{\sum_{n=1}^{N}\mathbb{1}[t^{(n)}=0]} \\
\mu_1 &= \frac{\sum_{n=1}^{N}\mathbb{1}[t^{(n)}=1]\cdot \mathbf{x}^{(n)}}{\sum_{n=1}^{N}\mathbb{1}[t^{(n)}=1]} \\
\Sigma &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}^{(n)}-\mu_{t^{(n)}})(\mathbf{x}^{(n)}-\mu_{t^{(n)}})^T
\end{aligned}$$

دسته بند بیز گاسی – تصمیم گیری

- GDA (GBC) decision boundary is based on class posterior:

$$\begin{aligned}\log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k^{-1}| - \frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) + \\ &\quad + \log p(t_k) - \log p(\mathbf{x})\end{aligned}$$

- Decision: take the class with the highest posterior probability

دسته بند بیز گاسی در برابر رگرسیون منطقی

- If you examine $p(t = 1|\mathbf{x})$ under GDA, you will find that it looks like this:

$$p(t|\mathbf{x}, \phi, \mu_0, \mu_1, \Sigma) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

  where $\mathbf{w}$ is an appropriate function of $(\phi, \mu_0, \mu_1, \Sigma)$

- So the decision boundary has the same form as logistic regression!
- When should we prefer GDA to LR, and vice versa?

دسته بند بیز گاسی در برابر رگرسیون منطقی

- GDA makes stronger modeling assumption: assumes class-conditional data is multivariate Gaussian
- If this is true, GDA is asymptotically efficient (best model in limit of large N)
- But LR is more robust, less sensitive to incorrect modeling assumptions
- Many class-conditional distributions lead to logistic classifier
- When these distributions are non-Gaussian, in limit of large N, LR beats GDA

✓456

ساده سازی مدل

What if $\mathbf{x}$ is high-dimensional?

- For Gaussian Bayes Classifier, if input $\mathbf{x}$ is high-dimensional, then covariance matrix has many parameters
- Save some parameters by using a shared covariance for the classes
- Any other idea you can think of?

مدل بیز ساده

- **Naive Bayes** is an alternative generative model: Assumes features independent given the class

$$p(\mathbf{x}|t=k) = \prod_{i=1}^{d} p(x_i|t=k)$$

- Assuming likelihoods are Gaussian, how many parameters required for Naive Bayes classifier?
- Important note: Naive Bayes does not assume a particular distribution

دسته بند بیز ساده

Given

- prior $p(t = k)$
- assuming features are conditionally independent given the class
- likelihood $p(x_i|t = k)$ for each x_i

The decision rule

$$y = arg \max_k p(t = k) \prod_{i=1}^{d} p(x_i|t = k)$$

- If the assumption of conditional independence holds, NB is the optimal classifier
- If not, a heavily regularized version of generative classifier
- What's the regularization?
- Note: NB's assumptions (cond. independence) typically do not hold in practice. However, the resulting algorithm still works well on many problems, and it typically serves as a decent baseline for more sophisticated models

دسته بند بیز ساده گاسی

- **Gaussian Naive Bayes** classifier assumes that the likelihoods are Gaussian:

$$p(x_i|t=k) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \exp\left[\frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}\right]$$

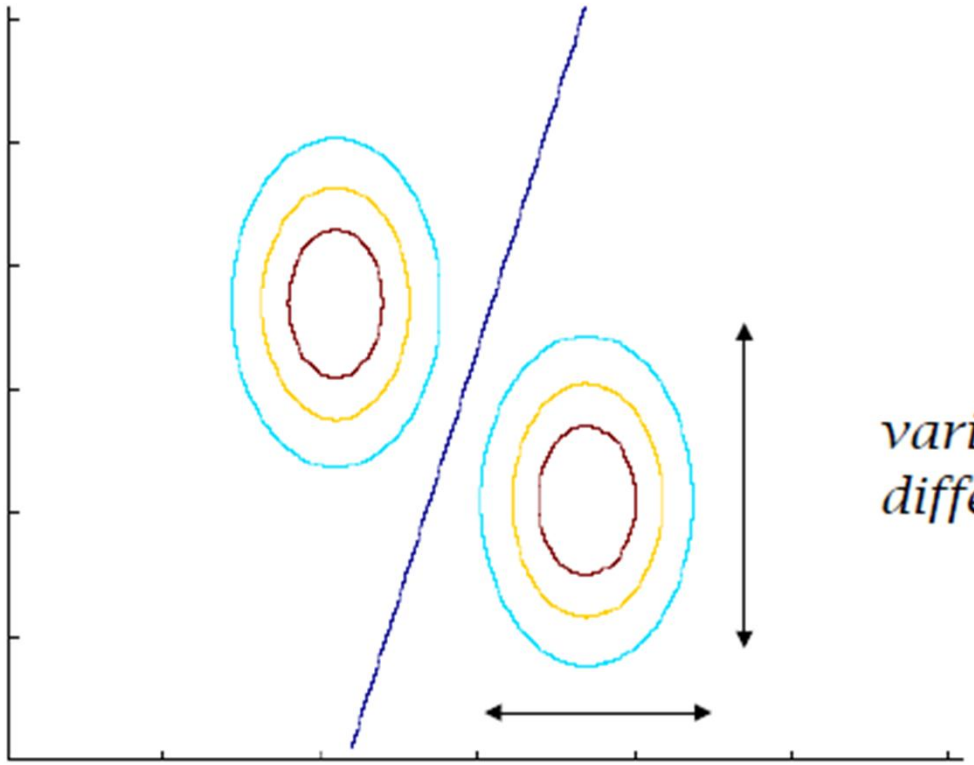
(this is just a 1-dim Gaussian, one for each input dimension)

- Model the same as Gaussian Discriminative Analysis with diagonal covariance matrix
- Maximum likelihood estimate of parameters

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k] \cdot x_i^{(n)}}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k]}$$

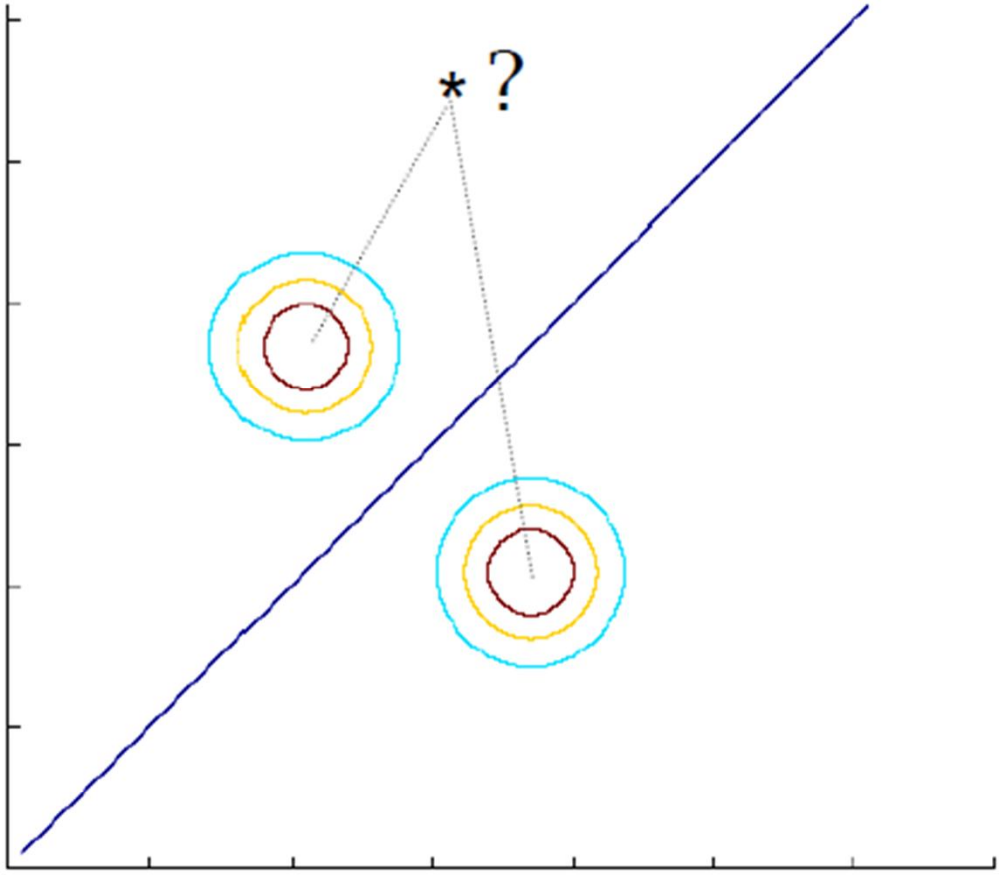
$$\sigma_{ik}^2 = \frac{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k] \cdot (x_i^{(n)}-\mu_{ik})^2}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k]}$$

مرز تصمیم - واریانس های مشترک بین دسته ها



variances may be different

مرز تصمیم - متقارن شعاعی



- Same variance across all classes and input dimensions, all class priors equal
- Classification only depends on distance to the mean. Why?

مرز تصمیم - متقارن شعاعی

- In this case: $\sigma_{i,k} = \sigma$ (just one parameter), class priors equal (e.g., $p(t_k) = 0.5$ for 2-class case)
- Going back to class posterior for GDA:

$$\begin{aligned}
\log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\
&= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k^{-1}| - \frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) + \\
&\quad + \log p(t_k) - \log p(\mathbf{x})
\end{aligned}$$

where we take $\Sigma_k = \sigma^2 I$ and ignore terms that don't depend on k (don't matter when we take max over classes):

$$\log p(t_k|\mathbf{x}) = -\frac{1}{2\sigma^2}(\mathbf{x}-\mu_k)^T(\mathbf{x}-\mu_k)$$