기계학습 (2022년도 2학기)

Probabilistic Models II

전북대학교 컴퓨터공학부

Overview

- Bayesian parameter estimation
- MAP estimation
- Gaussian discriminant analysis

Data Sparsity

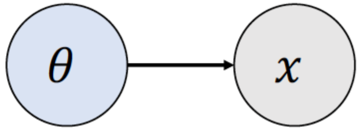
- Maximum likelihood has a pitfall: if you have too little data, it can overfit.
- E.g., what if you flip the coin twice and get H both times?

$$\theta_{\text{ML}} = \frac{N_H}{N_H + N_T} = \frac{2}{2+0} = 1$$

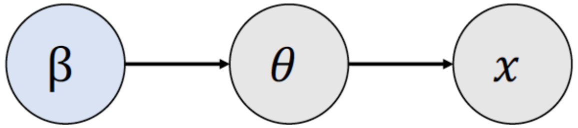
- Because it never observed T, it assigns this outcome probability 0. This problem is known as data sparsity.
- If you observe a single T in the test set, the log-likelihood is $-\infty$.

Bayesian Parameter Estimation

- In maximum likelihood, the observations are treated as random variables, but the parameters are not.



- The Bayesian approach treats the parameters as random variables as well. β is the set of parameters in the prior distribution of θ.



- To define a Bayesian model, we need to specify two distributions:
  - The prior distribution $\mathrm{p}(\theta)$, which encodes our beliefs about the parameters before we observe the data
  - The likelihood $\mathrm{p}\left(\mathcal{D} \,\middle|\, \theta\right)$, same as in maximum likelihood

Bayesian Parameter Estimation

- When we update our beliefs based on the observations, we compute the posterior distribution using Bayes' Rule:

$$p(\theta \mid \mathcal{D}) = \frac{p(\theta)p(\mathcal{D} \mid \theta)}{\int p(\theta')p(\mathcal{D} \mid \theta')\, \mathrm{d}\theta'}$$

- We rarely ever compute the denominator explicitly.

Bayesian Parameter Estimation

- Let's revisit the coin example. We already know the likelihood:

$$L(\theta) = p(\mathcal{D}) = \theta^{N_H}(1-\theta)^{N_T}$$

- It remains to specify the prior $\mathrm{p}(\theta)$.
  - We can choose an uninformative prior, which assumes as little as possible. A reasonable choice is the uniform prior.
  - But our experience tells us 0.5 is more likely than 0.99. One particularly useful prior that lets us specify this is the beta distribution:

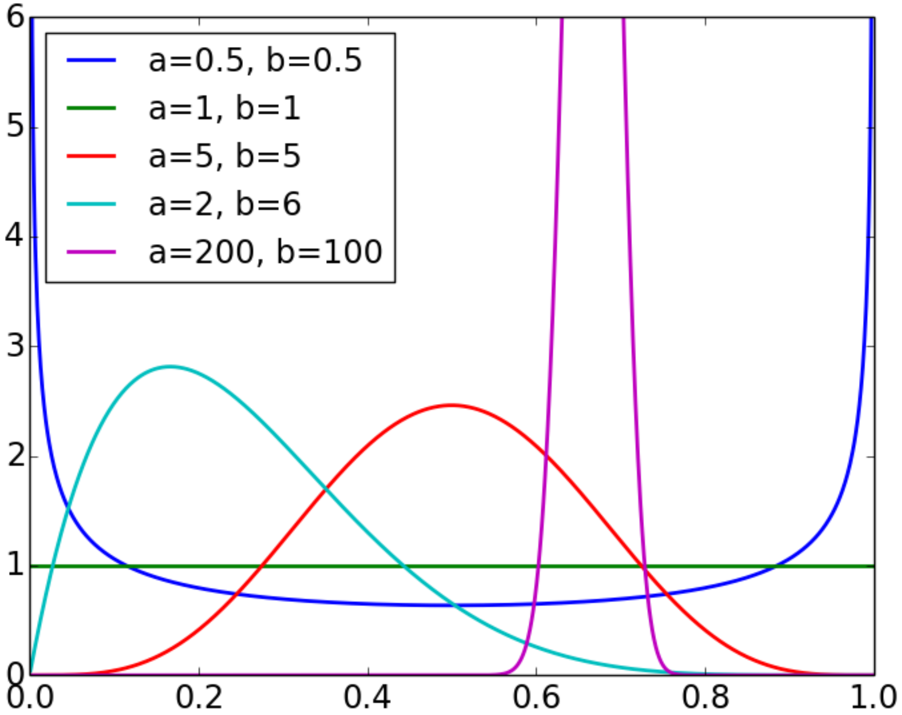
$$p(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1}(1-\theta)^{b-1}$$

- This notation for proportionality lets us ignore the normalization constant:

$$p(\theta; a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

Bayesian Parameter Estimation

- Beta distribution for various values of a, b:



- Some observations:
  - The expectation $\mathbb{E}[\theta] = a/(a + b)$.
  - The distribution gets more peaked when a and b are large.
  - The uniform distribution is the special case where $a = b = 1$.
- The main thing the beta distribution is used for is as a prior for the Bernoulli distribution.

Bayesian Parameter Estimation

- Computing the posterior distribution:

$$\begin{aligned} p(\theta \mid \mathcal{D}) &\propto p(\theta) p(\mathcal{D} \mid \theta) \\ &\propto \left[\theta^{a-1}(1-\theta)^{b-1}\right] \left[\theta^{N_H}(1-\theta)^{N_T}\right] \\ &= \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}. \end{aligned}$$

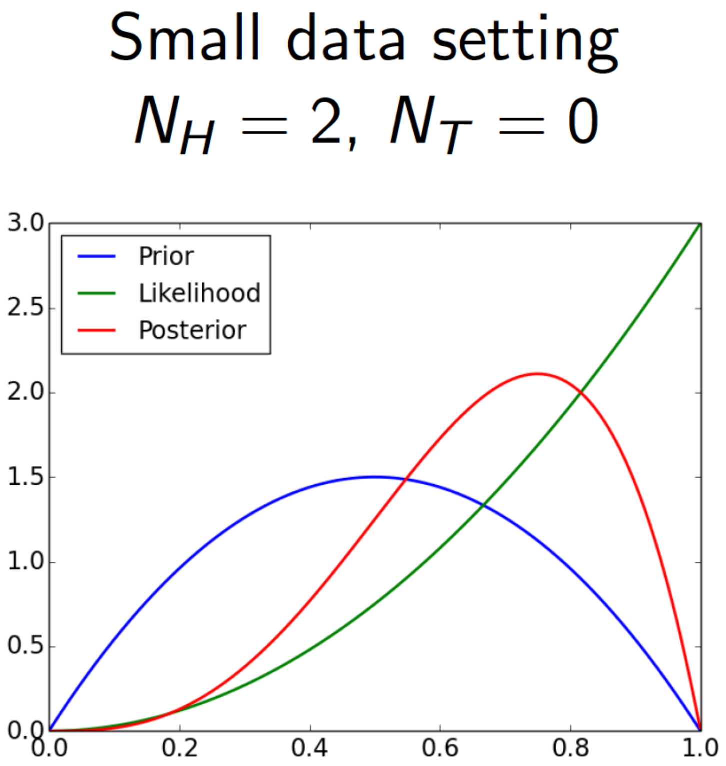
- This is just a beta distribution with parameters $N_H + a$ and $N_T + b$.
- The posterior expectation of θ is:

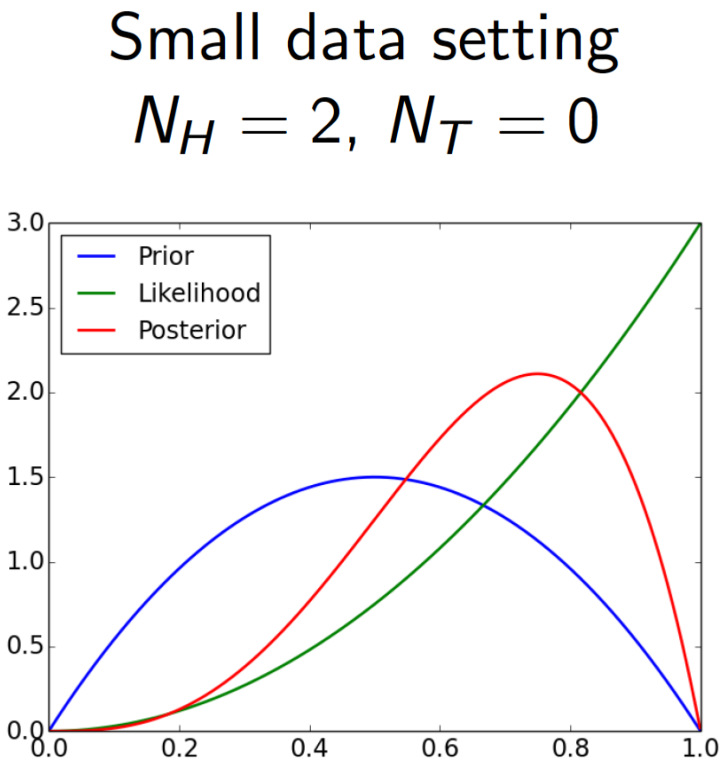
$$\mathbb{E}[\theta \mid \mathcal{D}] = \frac{N_H + a}{N_H + N_T + a + b}$$

- The parameters a and b of the prior can be thought of as pseudo-counts.
  - The reason this works is that the prior and likelihood have the same functional form. This phenomenon is known as conjugacy, and it's very useful.

Bayesian Parameter Estimation

- Bayesian inference for the coin flip example: $(a = b = 2)$





- When you have enough observations, the data overwhelm the prior.

Bayesian Parameter Estimation

- What do we actually do with the posterior?
- The posterior predictive distribution is the distribution over future observables given the past observations. We compute this by marginalizing out the parameter(s):

$$p(\mathcal{D}' \mid \mathcal{D}) = \int p(\theta \mid \mathcal{D}) p(\mathcal{D}' \mid \theta) \, \mathrm{d}\theta$$

- For the coin flip example:

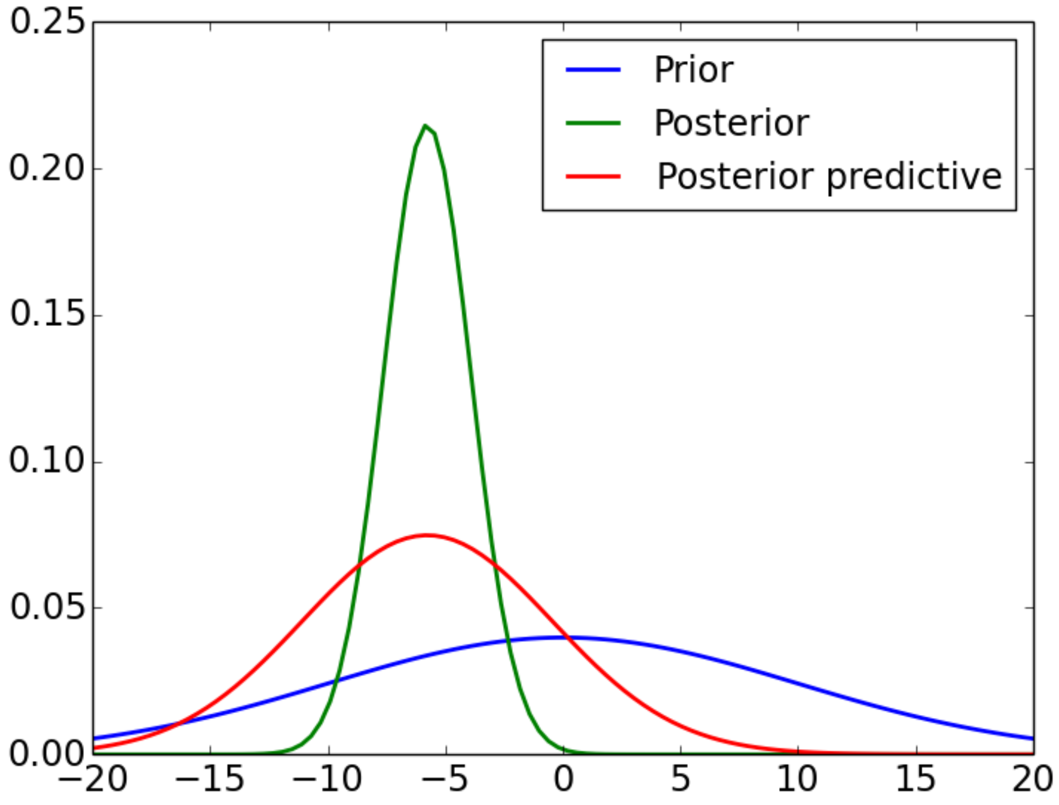
$$\begin{aligned}
\theta_{\text{pred}} &= Pr(x' = H \mid \mathcal{D}) \\
&= \int p(\theta \mid \mathcal{D}) Pr(x' = H \mid \theta) \, \mathrm{d}\theta \\
&= \int Beta(\theta; N_H + a, N_T + b) \cdot \theta \, \mathrm{d}\theta \\
&= \mathbb{E}_{Beta(\theta; N_H + a, N_T + b)}[\theta] \\
&= \frac{N_H + a}{N_H + N_T + a + b}.
\end{aligned}$$

$$\begin{aligned}
p(\theta \mid \mathcal{D}) &\propto p(\theta) p(\mathcal{D} \mid \theta) \\
&\propto \left[\theta^{a-1}(1-\theta)^{b-1}\right] \left[\theta^{N_H}(1-\theta)^{N_T}\right] \\
&= \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}.
\end{aligned}$$

동전 앞면이 나올 확률

Bayesian Parameter Estimation

- Bayesian estimation of the mean temperature in Toronto
  - Assume observations are i.i.d. Gaussian with known standard deviation σ and unknown mean μ
- Broad Gaussian prior over μ, centered at 0
- We can compute the posterior and posterior predictive distributions analytically (full derivation in notes)
  - 직접 해볼 것! (여기 10페이지 참고)
- Why is the posterior predictive distribution more spread out than the posterior distribution?



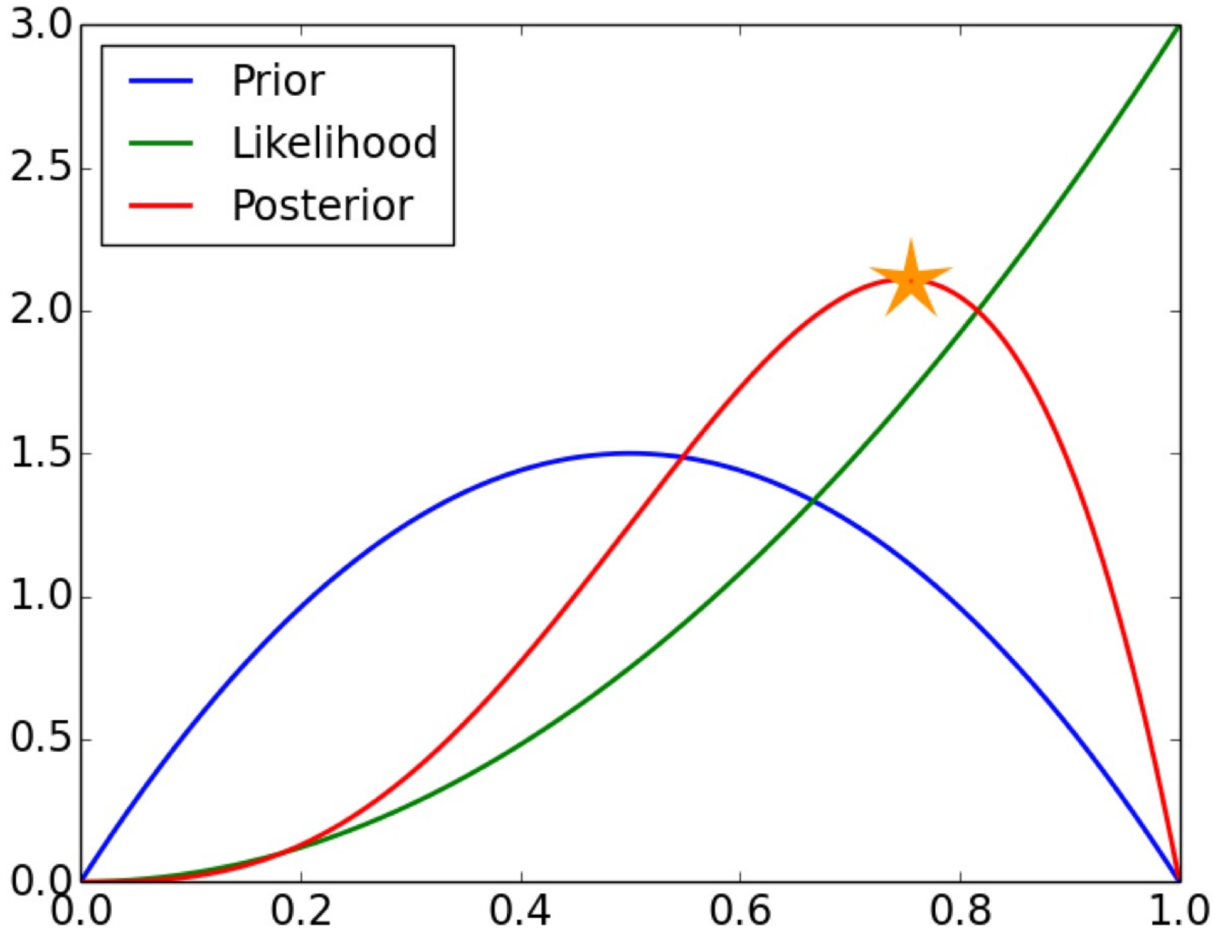
Bayesian Parameter Estimation

Comparison of maximum likelihood and Bayesian parameter estimation

- The Bayesian approach deals better with data sparsity
- Maximum likelihood is an optimization problem, while Bayesian parameter estimation is an integration problem (taking expectation).
  - This means maximum likelihood is much easier in practice, since we can just do gradient descent.
  - Automatic differentiation packages make it really easy to compute gradients.
  - There aren't any comparable black-box tools for Bayesian parameter estimation.

Maximum A-Posteriori Estimation

- Maximum a-posteriori (MAP) estimation: find the most likely parameter settings under the posterior



Maximum A-Posteriori Estimation

- This converts the Bayesian parameter estimation problem into a maximization problem

$$
\begin{aligned}
\hat{\theta}_{\mathrm{MAP}} &= \arg\max_{\theta} \; p(\theta \,|\, \mathcal{D}) \\
&= \arg\max_{\theta} \; p(\theta, \mathcal{D}) \\
&= \arg\max_{\theta} \; p(\theta)\, p(\mathcal{D} \,|\, \theta) \\
&= \arg\max_{\theta} \; \log p(\theta) + \log p(\mathcal{D} \,|\, \theta)
\end{aligned}
$$

Maximum A-Posteriori Estimation

- Joint probability in the coin flip example:

$$\begin{aligned} p(\theta \mid \mathcal{D}) &\propto p(\theta)p(\mathcal{D} \mid \theta) \\ &\propto \left[\theta^{a-1}(1-\theta)^{b-1}\right]\left[\theta^{N_H}(1-\theta)^{N_T}\right] \\ &= \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}. \end{aligned}$$

$$\begin{aligned} \log p(\theta, \mathcal{D}) &= \log p(\theta) + \log p(\mathcal{D} \mid \theta) \\ &= \text{Const} + (a-1)\log\theta + (b-1)\log(1-\theta) + N_H \log\theta + N_T \log(1-\theta) \\ &= \text{Const} + (N_H + a - 1)\log\theta + (N_T + b - 1)\log(1-\theta) \end{aligned}$$

- Maximize by finding a critical point

$$0 = \frac{\mathrm{d}}{\mathrm{d}\theta}\log p(\theta, \mathcal{D}) = \frac{N_H + a - 1}{\theta} - \frac{N_T + b - 1}{1-\theta}$$

- Solving for θ,

$$\hat{\theta}_{\text{MAP}} = \frac{N_H + a - 1}{N_H + N_T + a + b - 2}$$

Maximum A-Posteriori Estimation

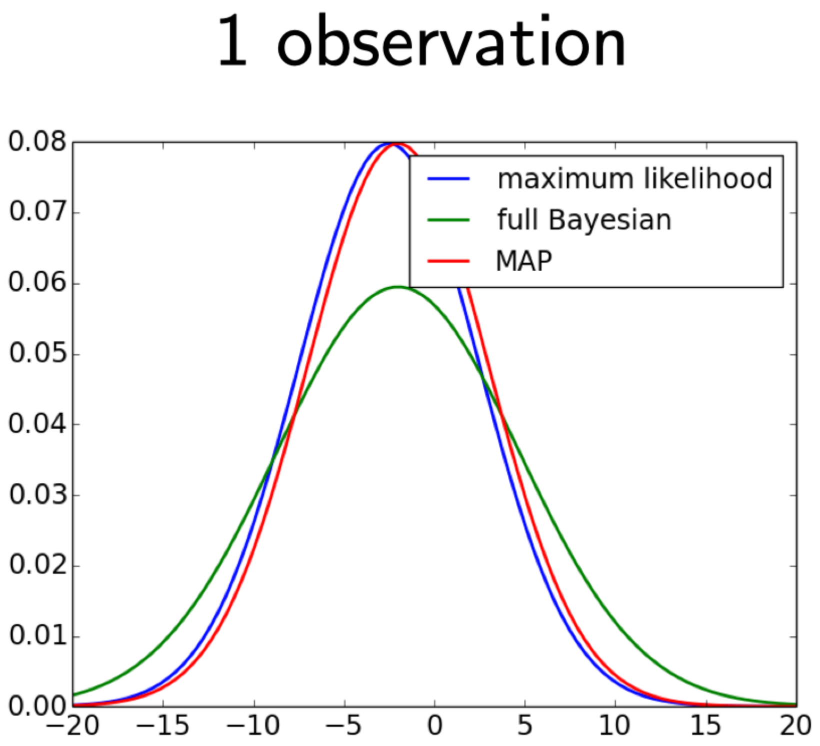
- Comparison of estimates in the coin flip example: $(a = b = 2)$

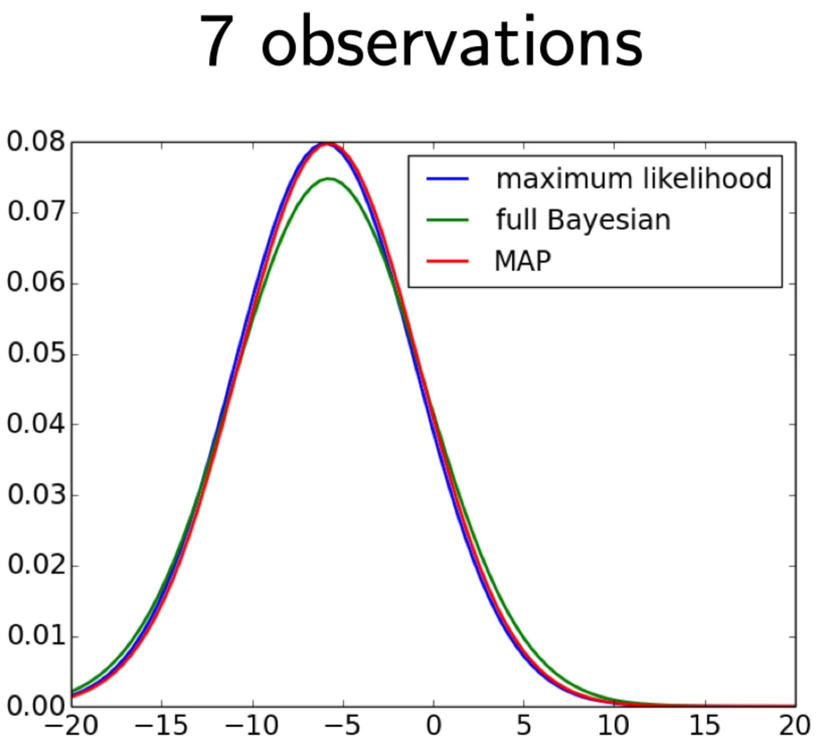
	Formula	$N_H = 2, N_T = 0$	$N_H = 55, N_T = 45$
$\hat{\theta}_{\text{ML}}$	$\frac{N_H}{N_H+N_T}$	1	$\frac{55}{100} = 0.55$
θ_{pred}	$\frac{N_H+a}{N_H+N_T+a+b}$	$\frac{4}{6} \approx 0.67$	$\frac{57}{104} \approx 0.548$
$\hat{\theta}_{\text{MAP}}$	$\frac{N_H+a-1}{N_H+N_T+a+b-2}$	$\frac{3}{4} = 0.75$	$\frac{56}{102} \approx 0.549$

- $\hat{\theta}_{MAP}$ assigns nonzero probabilities as long as a, b > 1.

Maximum A-Posteriori Estimation

- Comparison of predictions in the Toronto temperatures example





Gaussian Discriminant Analysis

- Generative models - model $\mathrm{p}(\mathbf{x}|t = \mathrm{k})$
- Instead of trying to separate classes, try to model what each class "looks like".
- Recall that $\mathrm{p}(\mathbf{x}|t = \mathrm{k})$ may be very complex

$$p(x_1, \cdots, x_d, y) = p(x_1|x_2, \cdots, x_d, y) \cdots p(x_{d-1}|x_d, y)p(x_d, y)$$

- Naive bayes used a conditional independence assumption. What else could we do? Choose a simple distribution.
- Today we will discuss fitting Gaussian distributions to our data.

Bayes Classifier

- Let's take a step back...
- Bayes Classifier

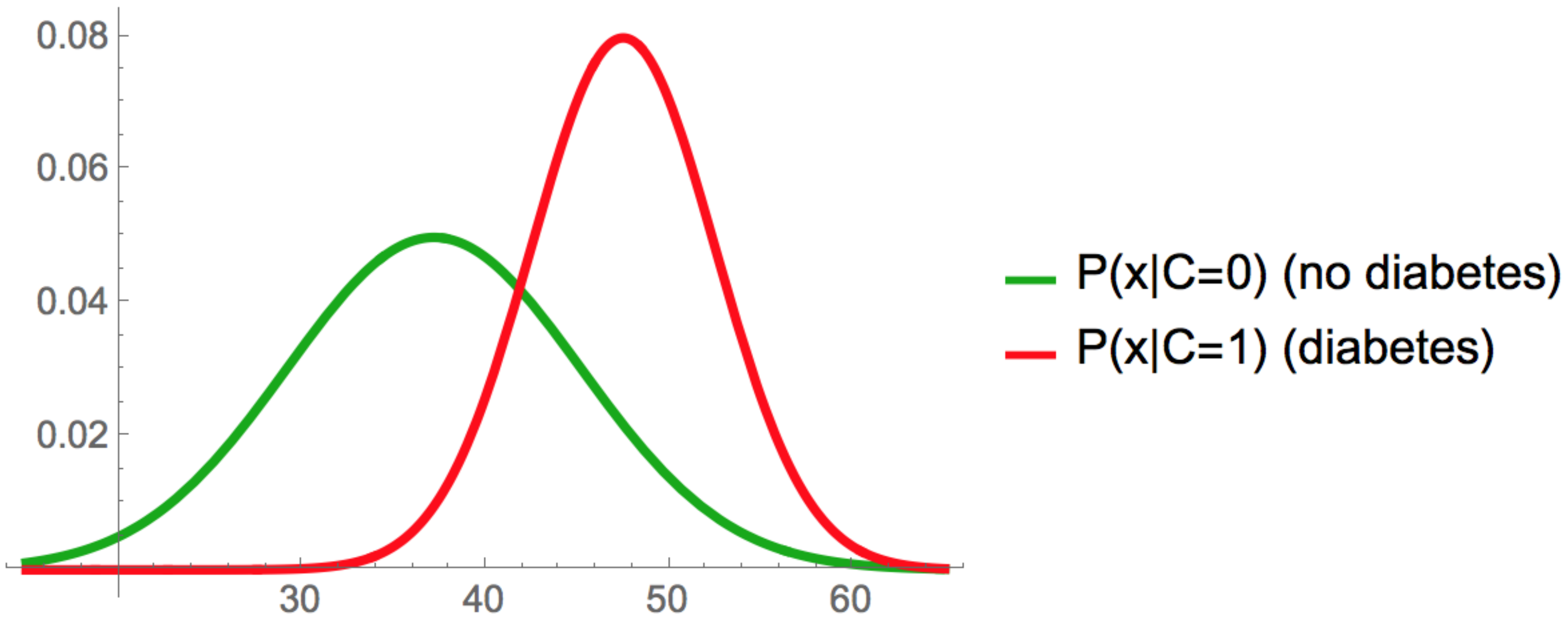
$$h(\mathbf{x}) = \arg\max_k p(t = k|\mathbf{x}) = \arg\max \frac{p(\mathbf{x}|t = k)p(t = k)}{p(\mathbf{x})}$$

$$= \arg\max_k p(\mathbf{x}|t = k)p(t = k)$$

- Talked about Discrete **x**, what if **x** is continuous?

Classification: Diabetes Example

- Observation per patient: White blood cell count & glucose value.



- How can we model $\mathrm{p}(\mathbf{x}|t = \mathrm{k})$? Multivariate Gaussian

Multivariate Data

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

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

attribute (또는 feature)

data sample

Multivariate Parameters

- Mean

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

- Covariance

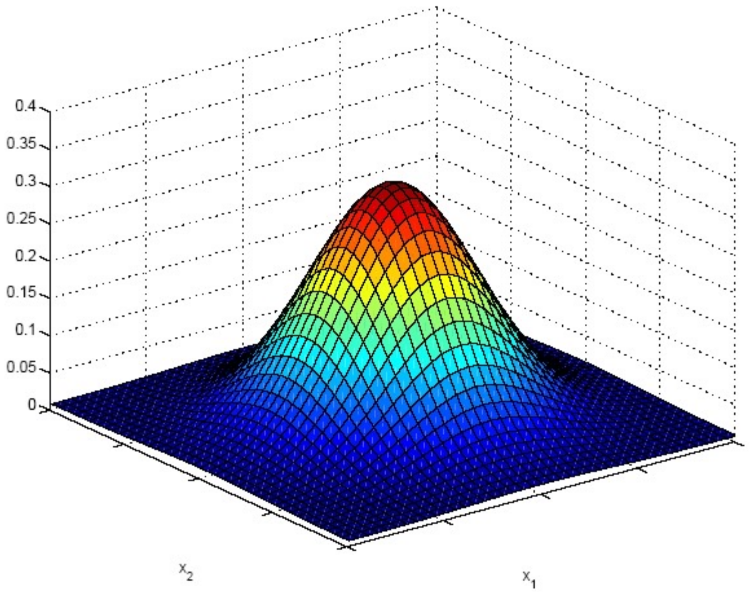
$$\Sigma = Cov(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mu)^T(\mathbf{x} - \mu)] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

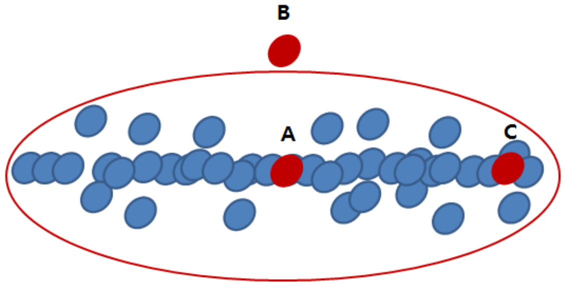
- For Gaussians - all you need to know to represent (not true in general)

Multivariate Gaussian Distribution

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

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



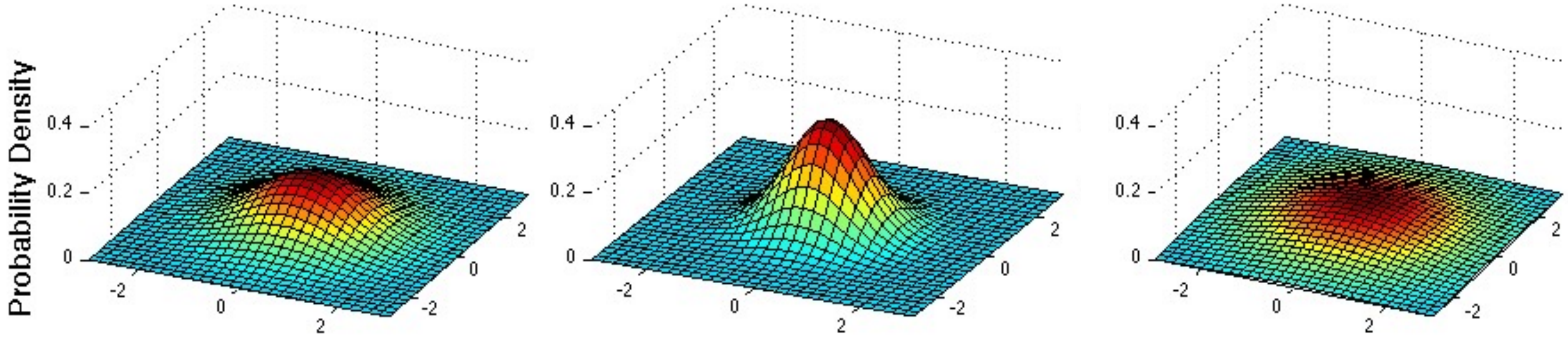


Euclidean distance: $d_E(A,B) < d_E(A,C)$
Mahalanobis distance: $d_E(A,B) > d_E(A,C)$

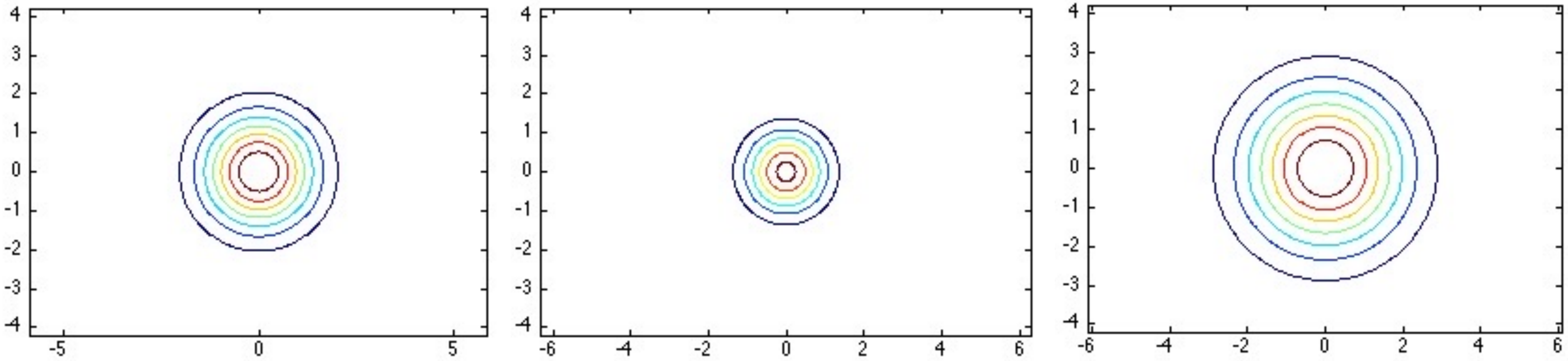
- Mahalanobis distance $(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k)$ measures the distance from $\mathbf{x}$ to μ_k in terms of Σ
- It normalizes for difference in variances and correlations

Bivariate Normal

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



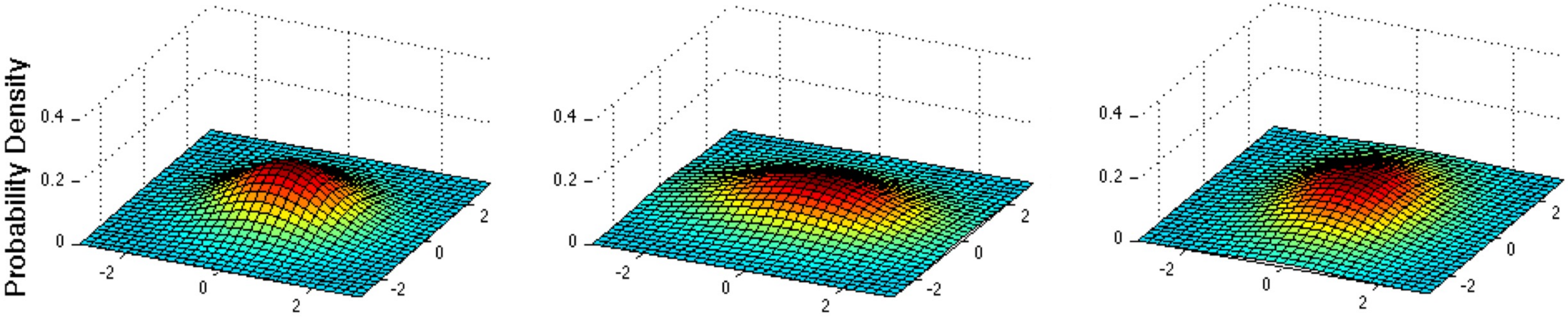
Probability density function



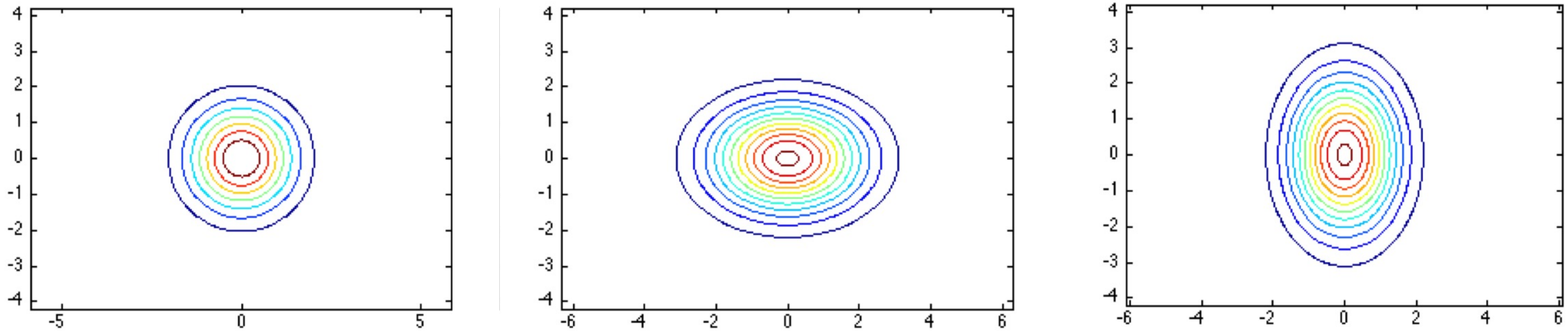
Contour plot of the pdf

Bivariate Normal

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



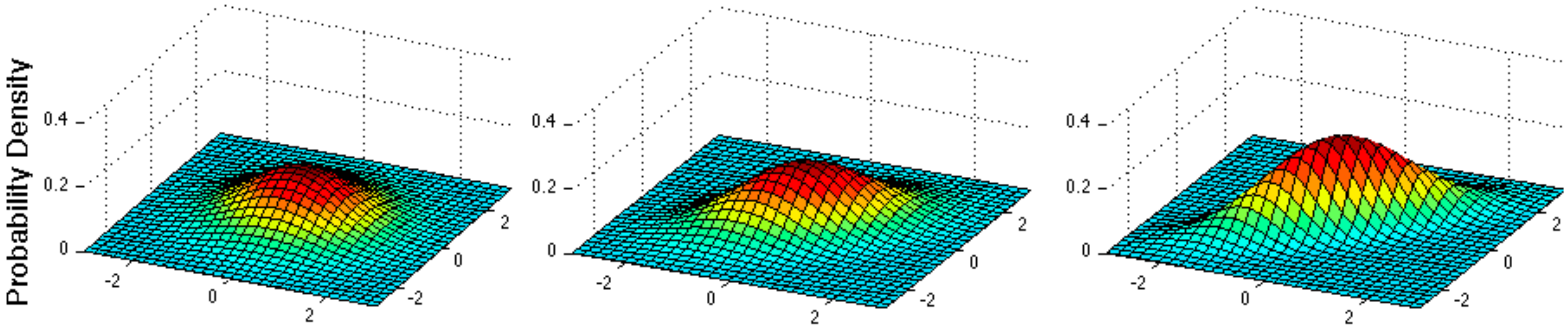
Probability density function



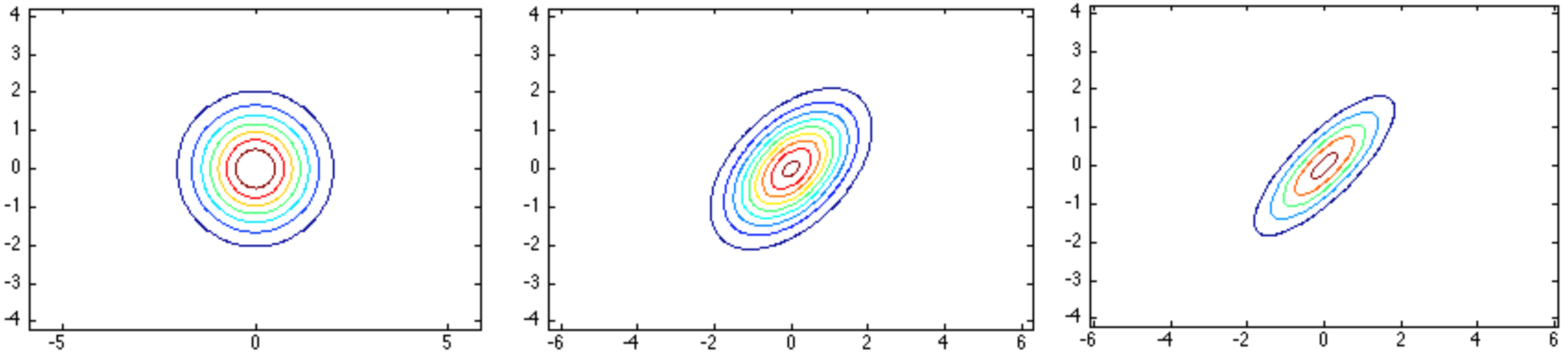
Contour plot of the pdf

Bivariate Normal

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



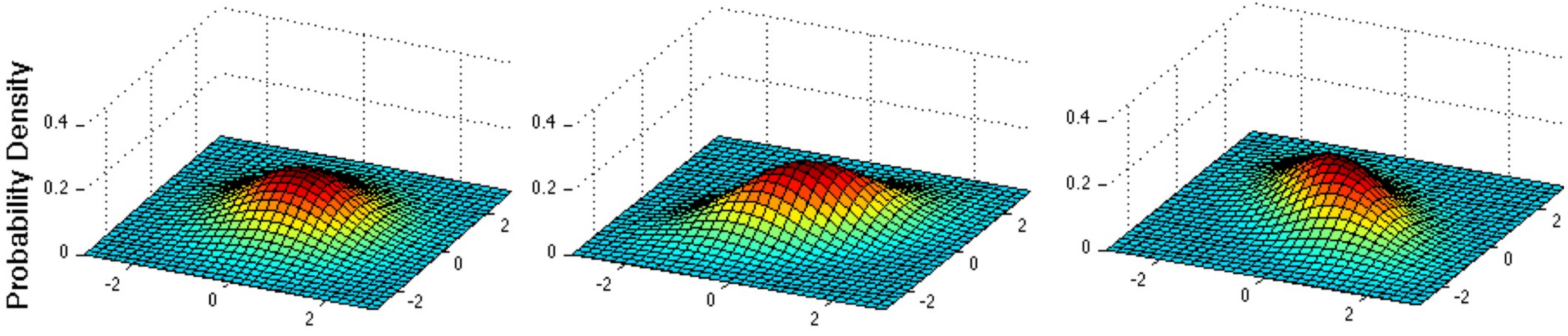
Probability density function



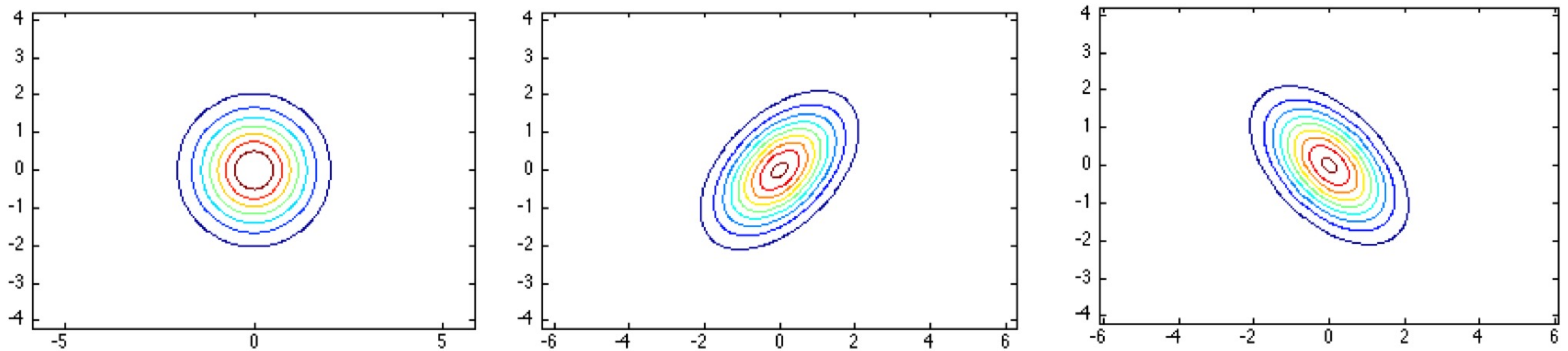
Contour plot of the pdf

Bivariate Normal

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



Probability density function



Contour plot of the pdf

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

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

- Multivariate Gaussian distribution:

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

  where $|\Sigma_k|$ denotes the determinant of the matrix, and *d* is dimension of **x**

- Each class *k* has associated mean vector μ_k and covariance matrix Σ_k
- Σ_k has $O(d^2)$ parameters - could be hard to estimate

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

- 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 boundary:
  (**x**의 class *k*와 class *l*에서의 score가 같음)

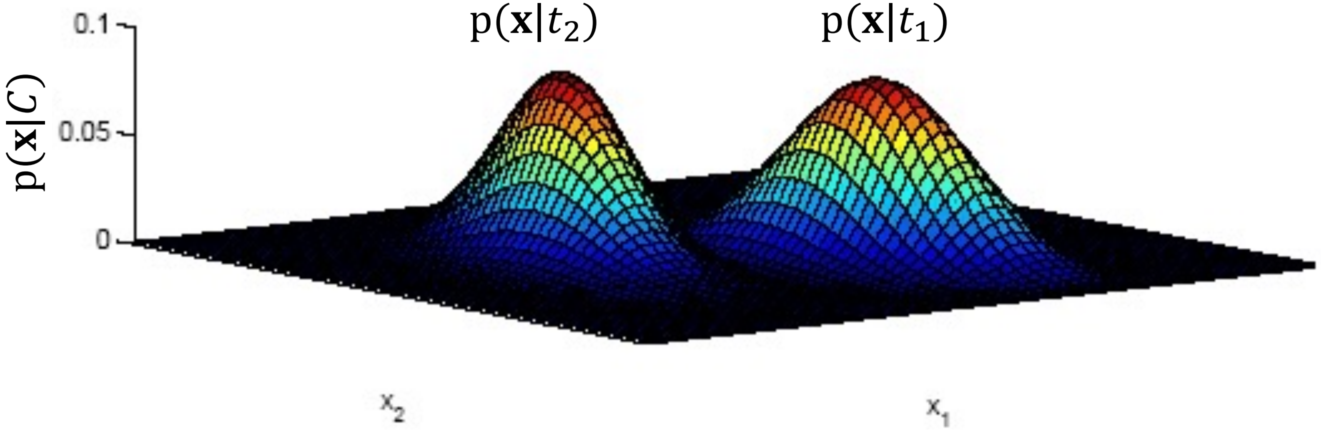
$$(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) = (\mathbf{x}-\mu_\ell)^T\Sigma_\ell^{-1}(\mathbf{x}-\mu_\ell) + \text{Const}$$

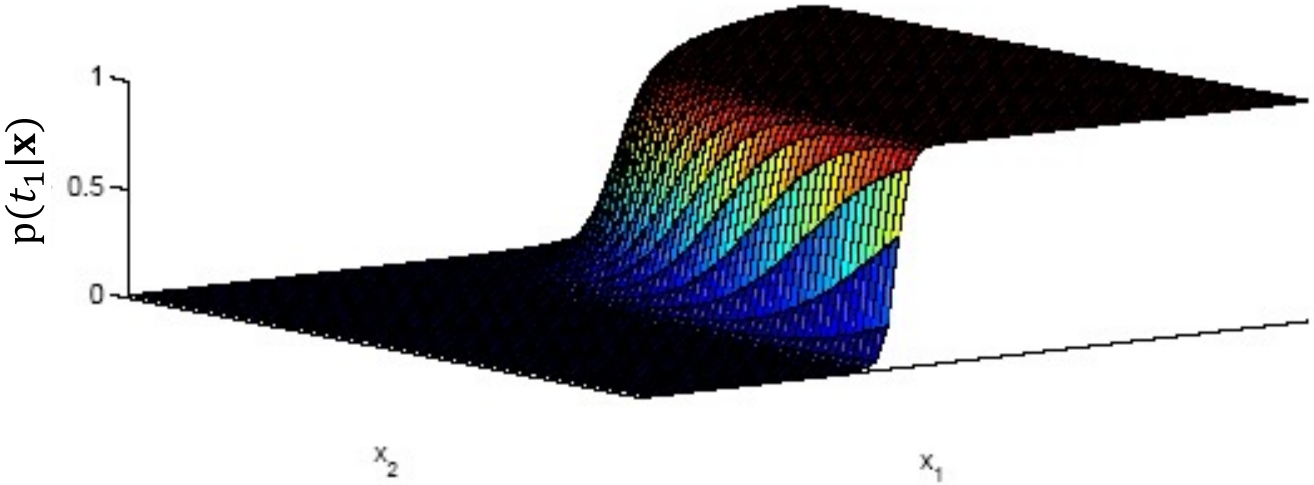
$$\mathbf{x}^T\Sigma_k^{-1}\mathbf{x} - 2\mu_k^T\Sigma_k^{-1}\mathbf{x} = \mathbf{x}^T\Sigma_\ell^{-1}\mathbf{x} - 2\mu_\ell^T\Sigma_\ell^{-1}\mathbf{x} + \text{Const}$$

- Quadratic function in **x**

Decision Boundary

likelihood for t_1 and t_2

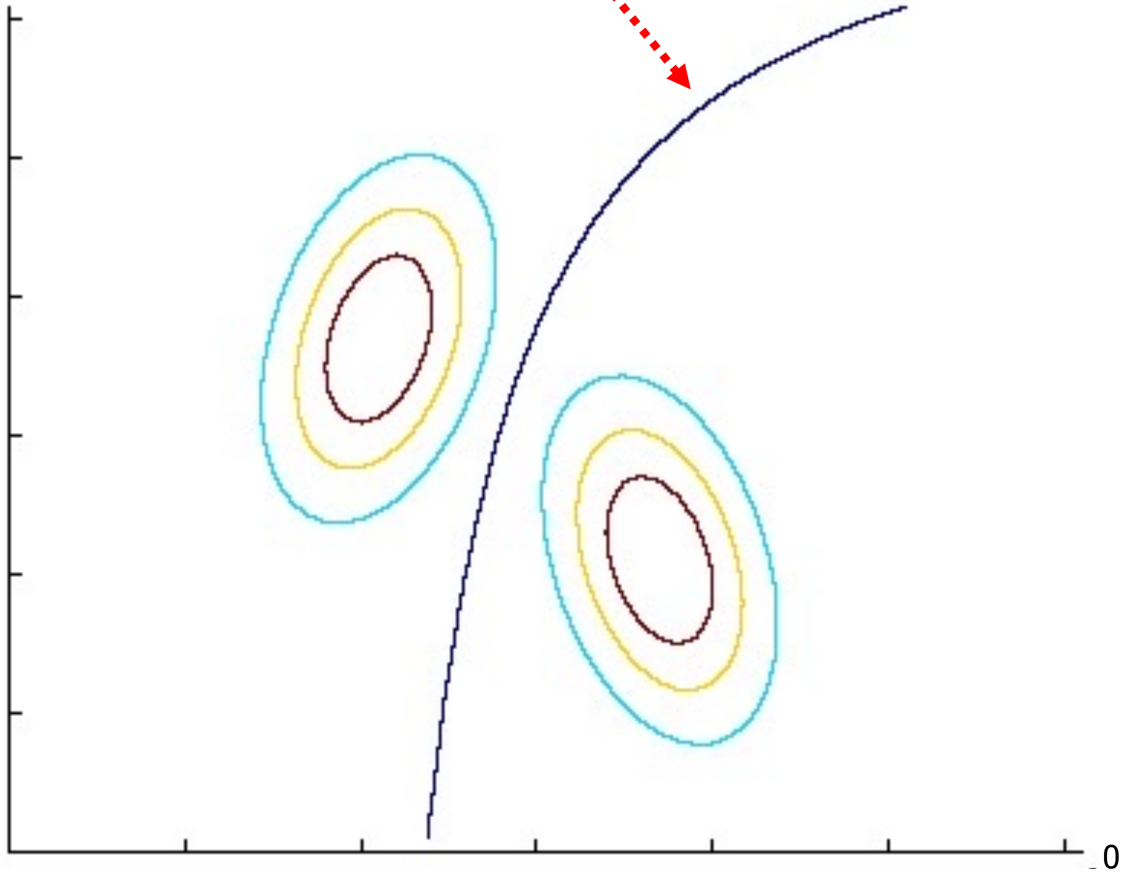




posterior for t_1

discriminant: $P(t_1|\mathbf{x}) = 0.5$

→ $\mathbf{x}$에 대한 2차 함수



Learning

- Learn the parameters for each class using maximum likelihood
- 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

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

Simplifying the Model

What if **x** is high-dimensional?

- For Gaussian Bayes Classifier, if input **x** is high-dimensional, then covariance matrix has many parameters
- Save some parameters by using a shared covariance for the classes

➔ $\Sigma_k = \Sigma_l$

- MLE in this case:

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}^{(n)} - \mu_{t^{(n)}})(\mathbf{x}^{(n)} - \mu_{t^{(n)}})^T$$

➔ Linear decision boundary!

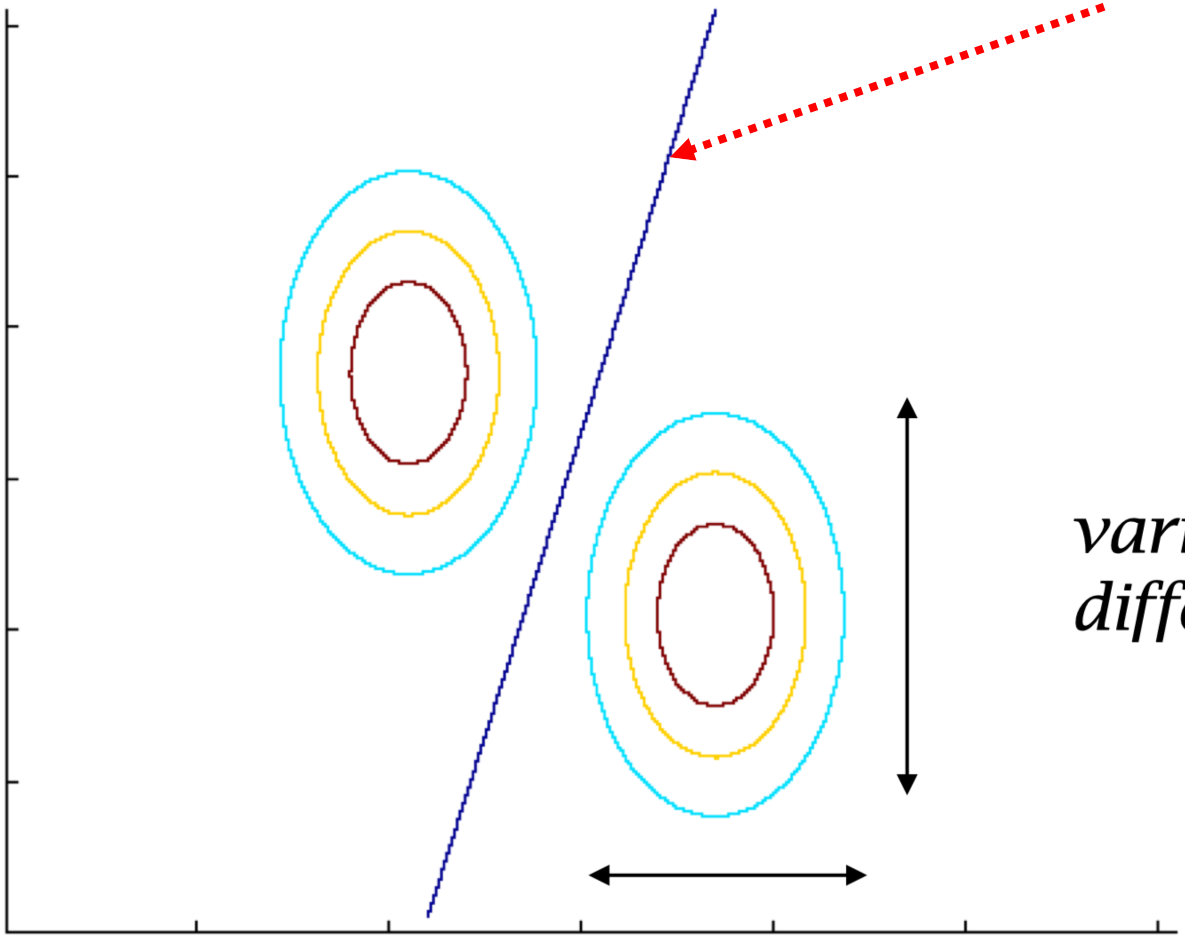
$$\cancel{\mathbf{x}^T\Sigma_k^{-1}}\mathbf{x} - 2\mu_k^T\Sigma_k^{-1}\mathbf{x} = \cancel{\mathbf{x}^T\Sigma_\ell^{-1}}\mathbf{x} - 2\mu_\ell^T\Sigma_\ell^{-1}\mathbf{x} + \text{Const}$$

- This is often called Linear Discriminant Analysis (LDA).

Decision Boundary: Shared Variances (between Classes)

$$\cancel{\mathbf{x}^T\Sigma_k^{-1}\mathbf{x}} - 2\mu_k^T\Sigma_k^{-1}\mathbf{x} = \cancel{\mathbf{x}^T\Sigma_\ell^{-1}\mathbf{x}} - 2\mu_\ell^T\Sigma_\ell^{-1}\mathbf{x} + \mathrm{Const}$$

X에 대한 1차 함수



variances may be different

Gaussian Discriminative Analysis vs Logistic Regression

- Binary classification: If you examine $\mathrm{p}(t = 1|\mathbf{x})$ under GDA and assume $\Sigma_0 = \Sigma_1 = \Sigma$, you will find that it looks like this:

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

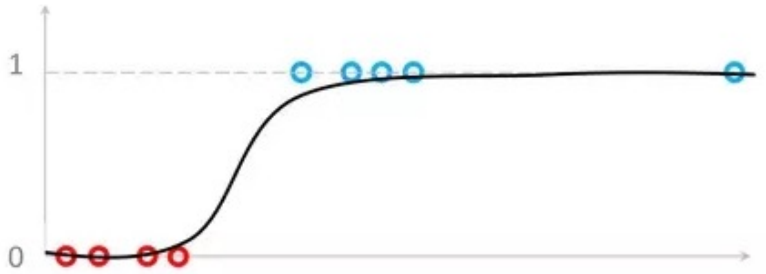
  where **w** is an appropriate function of $(\phi, \mu_0, \mu_1, \Sigma), \phi = \mathrm{p}(t = 1)$
  (직접 해볼 것!)

- Same model as logistic regression.

- When should we prefer GDA to LR, and vice versa?

Gaussian Discriminative Analysis vs Logistic Regression

- 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 (what loss is it optimizing?)



- Many class-conditional distributions lead to logistic classifier
- When these distributions are non-Gaussian (a.k.a almost always), LR usually beats GDA
- GDA can handle easily missing features

Naive Bayes

- Naive Bayes: 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?
- Equivalent to assuming Σ_k is diagonal.

Gaussian Naive Bayes

- 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

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

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

- What decision boundaries do we get?

Decision Boundary: Isotropic

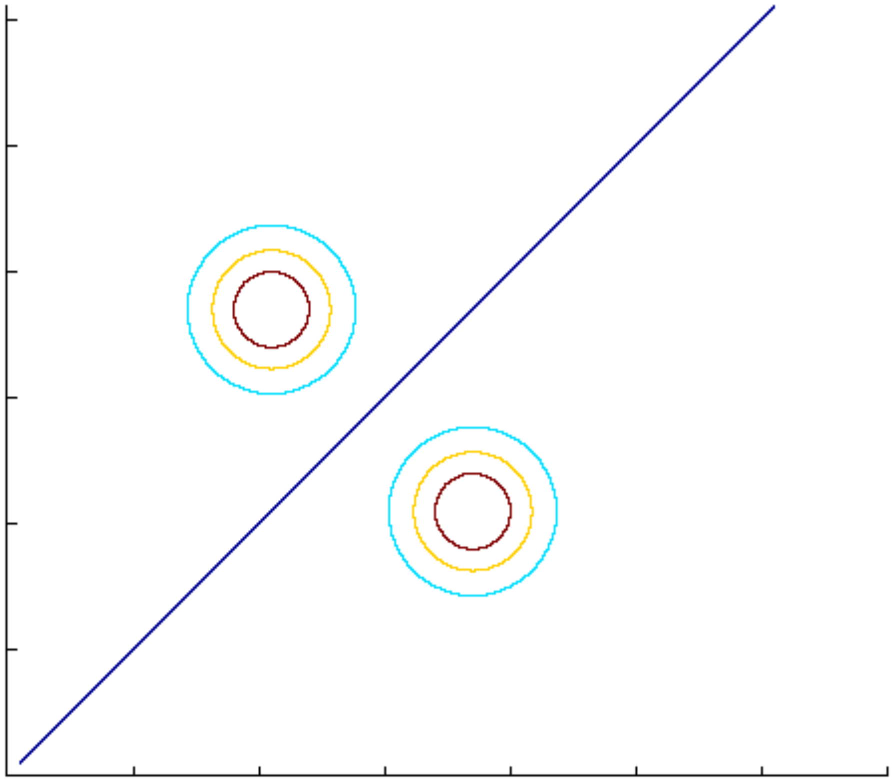
- In this case: $\sigma_{i,k} = \sigma$ (just one parameter), class priors equal (e.g., $\mathrm{p}(t_k)$ = 0.5 for 2-class case)
- Going back to class posterior for GDA:

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

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):

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

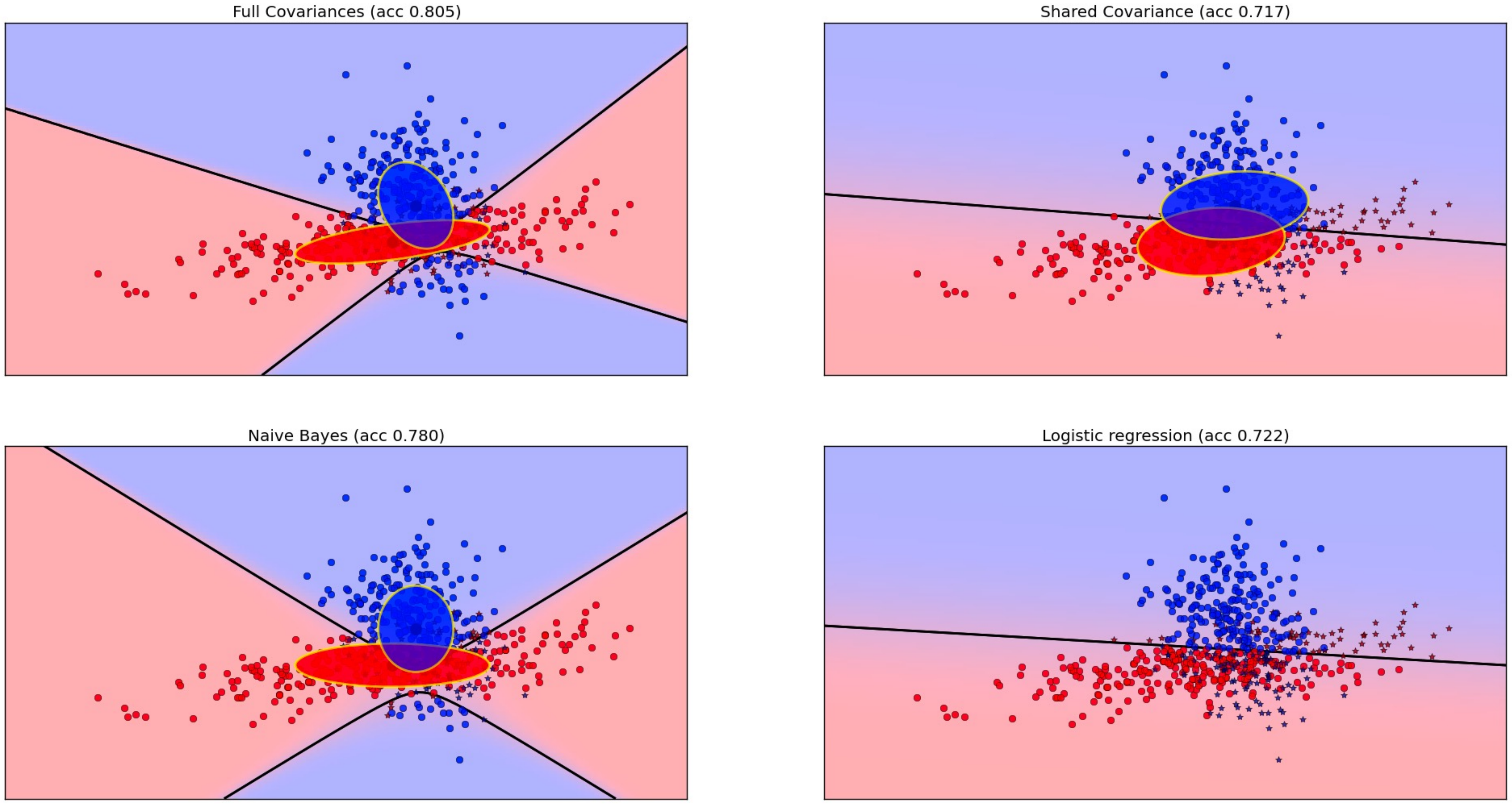
Decision Boundary: isotropic



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

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

Example



Full Covariances (acc 0.805)
Shared Covariance (acc 0.717)
Naive Bayes (acc 0.780)
Logistic regression (acc 0.722)

Generative models - Recap

- GDA - quadratic decision boundary.
- With shared covariance "collapses" to logistic regression.
- Generative models:
  - Flexible models, easy to add/remove class.
  - Handle missing data naturally
  - More "natural" way to think about things, but usually doesn't work as well.
- Tries to solve a hard problem in order to solve an easy problem.