CSC 2515: Introduction to Machine Learning

Lecture 7: Probabilistic Models

Amir-massoud Farahmand[1]

University of Toronto and Vector Institute

[1] Credit for slides goes to many members of the ML Group at the U of T, and beyond, including (recent past): Roger Grosse, Murat Erdogdu, Richard Zemel, Juan Felipe Carrasquilla, Emad Andrews, and myself.

Table of Contents

Overview

- Goal: A more focused discussion on models that explicitly represent probabilities
- MLE revisit
- Discriminative vs. Generative models
- Generative models
  - Naïve Bayes
  - Gaussian Discriminant Analysis (and Linear Discriminant Analysis)
- Bayesian approach to estimation and inference
- Maximum A-Posteriori Estimation (MAP) of parameters

Revisiting Maximum Likelihood Estimator

Revisiting Maximum Likelihood Estimiator (MLE)

- We have seen before that some ML algorithms can be derived using the Maximum Likelihood Estimation (MLE) principle.
  - Example: Regression with squared loss could be obtained as the MLE with Gaussian noise model
- Let's try to understand MLE better by starting with a simple example: Estimating the parameter of a biased coin
  - You flip a coin $N = 100$ times. It lands heads $N_H = 55$ times and tails $N_T = 45$ times.
  - What is the probability that it will come up heads if we flip again?
- Model: flips are independent Bernoulli random variables with parameter θ.
  - Assume the observations are independent and identically distributed (i.i.d.).

Maximum Likelihood

- The likelihood function is the density of the observed data, as a function of parameters θ.
- In our case, it is the probability of a *particular* sequence of H/T's.
- Under the Bernoulli model with i.i.d. observations: Let x_i be the # Hs in i-th flip ($x \in \{0, 1\}$)

$$p(x_i = 1|\theta) = \theta \quad \text{and} \quad p(x_i = 0|\theta) = 1 - \theta,$$

which can be written more compactly as

$$p(x_i|\theta) = \theta^{x_i}(1 - \theta)^{1-x_i}.$$

Maximum Likelihood

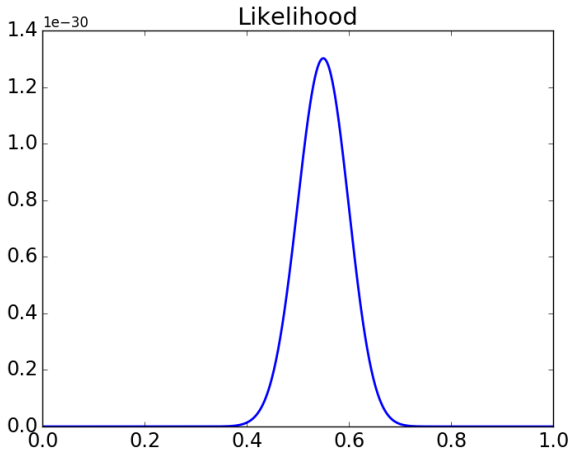
- Likelihood is

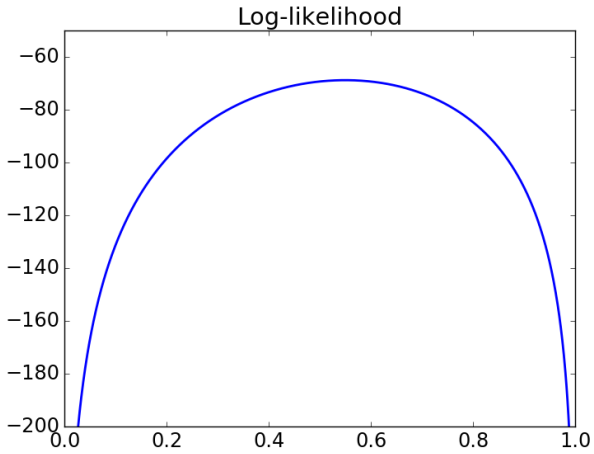
$$L(\theta) = p(x_1, ..., x_N|\theta) = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i} = \theta^{N_H}(1-\theta)^{N_T},$$

  where $N_H = \sum_i x_i$ and $N_T = N - \sum_i x_i$.
- We usually work with log-likelihoods:

$$\ell(\theta) = \log L(\theta) = N_H \log \theta + N_T \log(1-\theta).$$

- If $N_H = 55$, $N_T = 45$, the likelihood and log-likelihood as a function of θ are as follows:





Maximum Likelihood

- Good values of θ should assign high probability to the observed data. This motivates the maximum likelihood principle, i.e., choosing θ that maximizes the likelihood.
- We set the derivative of the likelihood function to zero finds its maximizer:

$$\begin{aligned}\frac{\mathrm{d}\ell}{\mathrm{d}\theta} &= \frac{\mathrm{d}}{\mathrm{d}\theta}\left(N_H \log\theta + N_T \log(1-\theta)\right) \\ &= \frac{N_H}{\theta} - \frac{N_T}{1-\theta}\end{aligned}$$

- Setting this to zero gives the maximum likelihood estimate:

$$\hat{\theta}_{\mathrm{ML}} = \frac{N_H}{N_H + N_T}.$$

- With this reminder, we are ready to talk about probabilistic models.

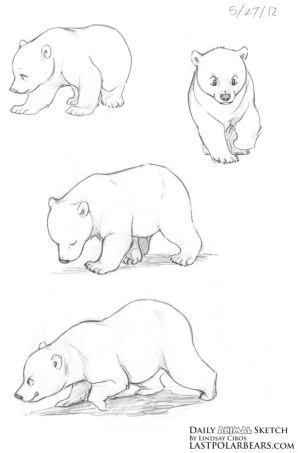
Generative vs. Discriminative Classifiers

Classification

- Given inputs $\mathbf{x}$ and classes t we can do classification in several ways. How?

Class 1

Class 2



$\mathbf{x}$

$t \in \{1, 2\}$

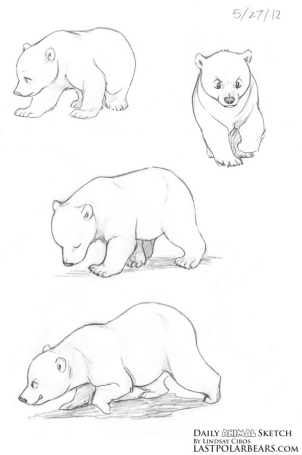
Features
(height, colour, etc.)

Discriminative Classifiers

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

Class 1

Class 2

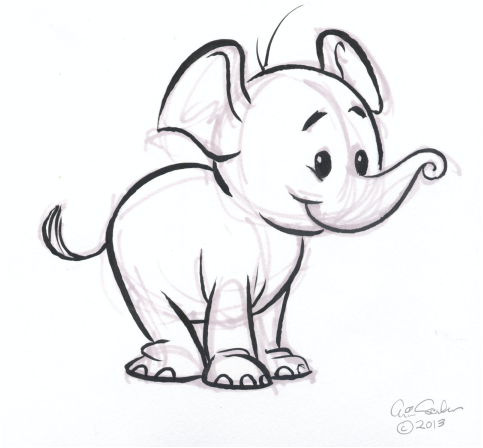


$$\mathbf{x} \longrightarrow t \in \{1, 2\}$$

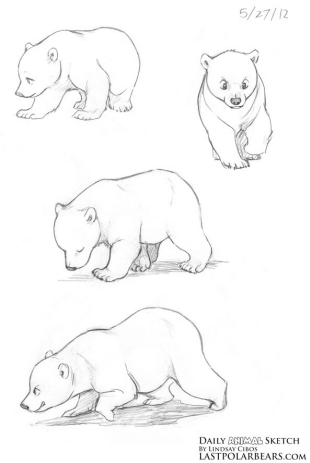
Discriminative Classifiers

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

Class 1



Class 2



$$\mathbf{x} \longrightarrow p(t|\mathbf{x})$$

Generative Classifiers

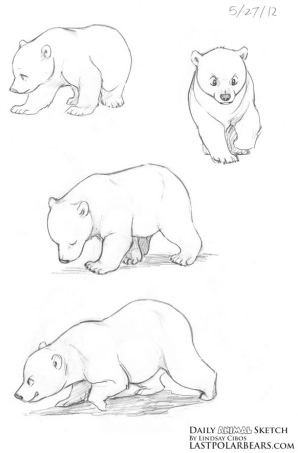
How about this approach: Build a model of "what data for a class looks like"

- Generative classifiers try to model $p(\mathbf{x}, t)$. If we know $p(t)$ we can easily compute $p(\mathbf{x}|t)$.

Class 1

$$p(\mathbf{x}|t) \longleftarrow t \in \{1, 2\}$$

Class 2

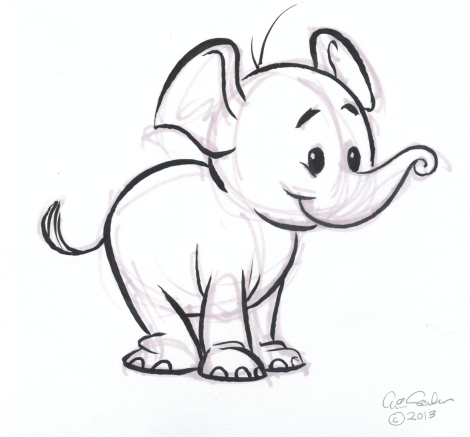


Generative Classifiers

How about this approach: Build a model of "what data for a class looks like"

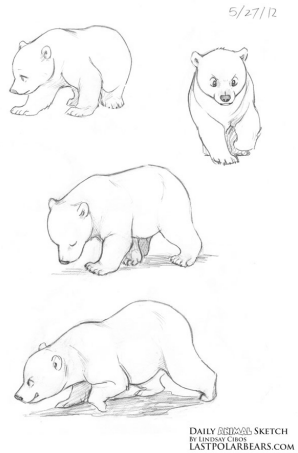
- Generative classifiers try to model $p(\mathbf{x}, t)$. If we know $p(t)$ we can easily compute $p(\mathbf{x}|t)$.
- Classification via Bayes rule (thus also called Bayes classifiers).

Class 1



$$p(\mathbf{x}|t) \longleftarrow t \in \{1, 2\}$$

Class 2



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

Generative vs. Discriminative

Two approaches to classification:

- Discriminative approach: estimate parameters of decision boundary/class separator directly from labeled examples.
  - Tries to solve: How do I separate the classes?
    - Sometimes this means modelling $p(t \mid \mathbf{x})$, e.g. logistic regression.
    - Sometimes this means learning a decision rule without a probabilistic interpretation, e.g., KNN or SVM.
- Generative approach: model the distribution of inputs generated from the class (Bayes classifier).
  - Tries to solve: What does each class "look" like?
  - Build a model of $p(\mathbf{x}|t)$ and $p(t)$.
  - Apply Bayes Rule to compute $p(t|\mathbf{x})$

A Generative Model: Bayes Classifier

- Aim to classify text into spam/not-spam (yes $t = 1$; no $t = 0$)
- Example: "You are one of the very few who have been selected as a winners for the free $1000 Gift Card."
- Use bag-of-words features, get binary vector $\mathbf{x}$ for each email
- Vocabulary:
  - "a": 1
  - ...
  - "car": 0
  - "card": 1
  - ...
  - "win": 0
  - "winner": 1
  - "winter": 0
  - ...
  - "you": 1

Bayes Classifier

- Given features $\mathbf{x} = [x_1, x_2, \cdots, x_D]^\top$ we want to compute class probabilities using Bayes Rule:

$$\underbrace{p(t|\mathbf{x})}_{\text{Pr. class given words}} = \frac{p(\mathbf{x}, t)}{p(\mathbf{x})} = \frac{\overbrace{p(\mathbf{x}|t)}^{\text{Pr. words given class}} \quad p(t)}{p(\mathbf{x})}$$

- Each of these terms have specific names:

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

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

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

- To compute $p(t|\mathbf{x})$ we need: $p(\mathbf{x}|t)$ and $p(t)$
- Q: How can we learn $p(\mathbf{x}|t)$ and $p(t)$?

Naïve Bayes Classifier

Naïve Bayes

- Assume that we have two classes: spam and non-spam. We have a dictionary of D words, and binary features $\mathbf{x} = [x_1, \ldots, x_D]$ saying whether each word appears in the e-mail.
- If we define a joint distribution $p(t, x_1, \ldots, x_D)$, this gives enough information to determine $p(t)$ and $p(\mathbf{x}|t)$.
- Q: How many parameters do you need to specify such a joint distribution?

Naïve Bayes

- Problem: specifying a joint distribution $p(t, x_1, \ldots, x_D)$ over $D+1$ binary variables requires $2^{D+1} - 1$ entries. This is computationally prohibitive and would require an absurd amount of data to fit.
- We'd like to impose structure on the distribution such that:
  - it can be compactly represented
  - learning and inference are both tractable

Naïve Bayes

- Naïve assumption: Naïve Bayes assumes that the word features x_i are conditionally independent given the class t.
  - This means that x_i and x_j are independent conditioned on the class label t, i.e.,
    $$p(x_i, x_j|t) = p(x_i|t)p(x_j|t),$$
    for $i \neq j$.
  - This doesn't mean they are independent.
    - Q: Why?
  - Therefore, we have
    $$p(t, x_1, \ldots, x_D) = p(t)p(x_1|t)\cdots p(x_D|t).$$

Naïve Bayes

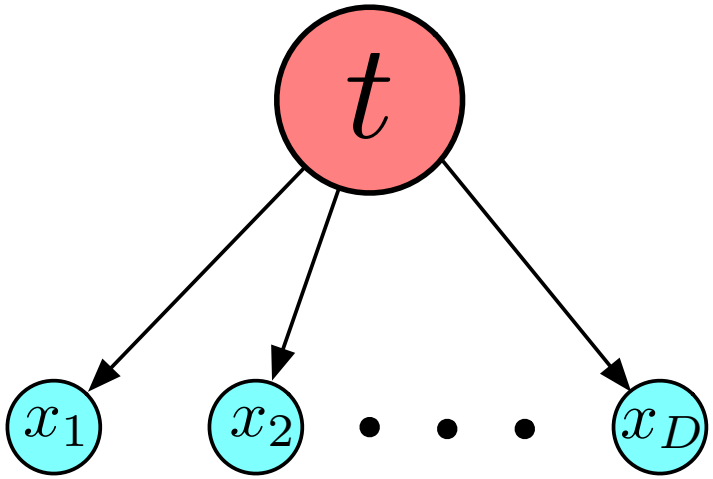
- The joint distribution is represented as

$$p(t, x_1, \ldots, x_D) = p(t)p(x_1|t) \cdots p(x_D|t).$$

- Compact representation of the joint distribution
  - Prior probability of class: $p(t = 1) = \pi$ (e.g., spam email), and hence $p(t = 0) = 1 - \pi$.
  - Conditional probability of word feature given class: $p(x_j = 1|t) = \theta_{jt}$ (e.g., word "price" appearing in spam), and hence $p(x_j = 0|t) = 1 - \theta_{jt}$.
  - $2D + 1$ parameters total (before $2^{D+1} - 1$)

Bayes Nets

- We can represent this model using a directed graphical model, or a Bayesian network:



- This graph structure means that the joint distribution factorizes as a product of conditional distributions for each variable given its parent(s).
- Intuitively, one can think of the edges as reflecting a causal structure. But mathematically, this doesn't hold without additional assumptions.
- This is a very simple graphical model. There are more complex structures too.

Naïve Bayes: Learning

- The parameters can be learned efficiently because the log-likelihood decomposes into independent terms for each feature.

$$
\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(t^{(i)}, \mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \left\{ p(\mathbf{x}^{(i)} | t^{(i)}) p(t^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \log \left\{ p(t^{(i)}) \prod_{j=1}^{D} p(x_j^{(i)} \mid t^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \left[\log p(t^{(i)}) + \sum_{j=1}^{D} \log p(x_j^{(i)} \mid t^{(i)}) \right] \\
&= \underbrace{\sum_{i=1}^{N} \log p(t^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{of labels}}} + \sum_{j=1}^{D} \underbrace{\sum_{i=1}^{N} \log p(x_j^{(i)} \mid t^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{for feature } x_j}}
\end{aligned}
$$

- Each of these log-likelihood terms depends on different sets of parameters, so they can be optimized independently.

Naïve Bayes: Learning

- We can handle these terms separately. For the prior we maximize:

$$\sum_{i=1}^{N} \log p(t^{(i)})$$

- This is a minor variant of our coin flip example:
- Let $p(t^{(i)} = 1) = \pi$.
- Note: $p(t^{(i)}) = \pi^{t^{(i)}}(1-\pi)^{1-t^{(i)}}$. [Q: Why?]
- Log-likelihood:

$$\sum_{i=1}^{N} \log p(t^{(i)}) = \sum_{i=1}^{N} t^{(i)} \log \pi + \sum_{i=1}^{N} (1 - t^{(i)}) \log(1-\pi)$$

- Obtain MLEs by setting derivatives to zero (Verify it!):

$$\hat{\pi} = \frac{\sum_i \mathbb{I}\{t^{(i)} = 1\}}{N} = \frac{\#\text{ spams in dataset}}{\text{total }\#\text{ samples}}$$

Naïve Bayes: Learning

- Each θ_{jt}'s can be treated separately:

$$\max_{\theta_{jt}} \sum_{i=1}^{N} \log p(x_j^{(i)} \,|\, t^{(i)}).$$

- This is (again) a minor variant of our coin flip example.
- Recall that $\theta_{jt} = p(x_j^{(i)} = 1 \,|\, t)$.
- Note that

$$p(x_j^{(i)} \,|\, t) = p(x_j^{(i)} = 1 \,|\, t)^{x_j^{(i)}} p(x_j^{(i)} = 0 \,|\, t)^{1-x_j^{(i)}} = \theta_{jt}^{x_j^{(i)}} (1-\theta_{jt})^{1-x_j^{(i)}}.$$

- Log-likelihood:

$$\begin{aligned}\sum_{i=1}^{N} \log p(x_j^{(i)} \,|\, t^{(i)}) = &\sum_{i=1}^{N} t^{(i)} \left\{ x_j^{(i)} \log \theta_{j1} + (1 - x_j^{(i)}) \log(1-\theta_{j1}) \right\} \\ &+ \sum_{i=1}^{N} (1 - t^{(i)}) \left\{ x_j^{(i)} \log \theta_{j0} + (1 - x_j^{(i)}) \log(1-\theta_{j0}) \right\}\end{aligned}$$

Naïve Bayes: Learning

- Log-likelihood:

$$\sum_{i=1}^{N} \log p(x_j^{(i)} \mid t^{(i)}) = \sum_{i=1}^{N} t^{(i)} \left\{ x_j^{(i)} \log \theta_{j1} + (1 - x_j^{(i)}) \log(1 - \theta_{j1}) \right\} + \sum_{i=1}^{N} (1 - t^{(i)}) \left\{ x_j^{(i)} \log \theta_{j0} + (1 - x_j^{(i)}) \log(1 - \theta_{j0}) \right\}$$

- Obtain MLEs by setting derivatives to zero:

$$\hat{\theta}_{jt} = \frac{\sum_i \mathbb{I}\{x_j^{(i)} = 1 \;\&\; t^{(i)} = t\}}{\sum_i \mathbb{I}\{t^{(i)} = t\}} \overset{\text{for } t=1}{=} \frac{\#\text{word } j \text{ appears in spams}}{\#\text{ spams in dataset}}$$

Naïve Bayes: Inference

- We predict the category of an input $\mathbf{x}$ by performing inference in the model.
- Apply Bayes' Rule:

$$p(t \,|\, \mathbf{x}) = \frac{p(t)p(\mathbf{x} \,|\, t)}{\sum_{t'} p(t')p(\mathbf{x} \,|\, t')} = \frac{p(t) \prod_{j=1}^{D} p(x_j \,|\, t)}{\sum_{t'} p(t') \prod_{j=1}^{D} p(x_j \,|\, t')}$$

- We need not compute the denominator if we merely want to determine the most likely t (why?).
- Shorthand notation:

$$p(t \,|\, \mathbf{x}) \propto p(t) \prod_{j=1}^{D} p(x_j \,|\, t)$$

- For input $\mathbf{x}$, predict by computing the values of $p(t) \prod_{j=1}^{D} p(x_j \,|\, t)$ for different t and choose the largest.

Naïve Bayes

- Naïve Bayes is an amazingly cheap learning algorithm!
- Training time: estimate parameters using maximum likelihood
  - Compute co-occurrence counts of each feature with the labels.
  - Requires only one pass through the data.
- Test time: apply Bayes' Rule
  - Cheap because of the model structure. (For more general models, Bayesian inference can be very expensive and/or complicated.)
- We covered the Bernoulli case for simplicity. But our analysis easily extends to other probability distributions.
- Unfortunately, it is usually less accurate in practice compared to discriminative models due to its "naïve" independence assumption.

Bayesian Parameter Estimation

MLE Issue: Data Sparsity

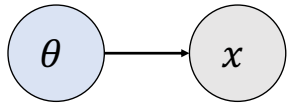
- Maximum likelihood has a pitfall: if you have too little data, it can overfit.
- Example: What if you flip the coin twice and get H both times?

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

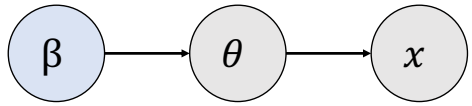
- Because it never observed T, it would assign the probability of 0 to the future appearance of T. This is not an intuitively reasonable answer. It was just unlucky that we did not observe any T in two flips, but it does not mean that the coin would never be a T. This is an example of overfitting. And this problem is sometimes known as data sparsity.
- We can mitigate this issue by using a Bayesian approach to estimation and inference.

Bayesian Parameter Estimation and Inference

- 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. The parameter θ has a prior probability, specified by another parameter β.



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

Bayesian Parameter Estimation and Inference

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

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

- We rarely ever compute the denominator explicitly. In general, it is computationally intractable.

Remark

There is a subtle difference between the interpretation of probability according to a Bayesian and a frequentist (who recommends MLE or its regularized variants). For the former, probability is a degree of belief about the truth of a statement; for the latter, a probability is the number of times a statement is true when we observe a lot of samples.

Bayesian Parameter Estimation and Inference

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

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

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

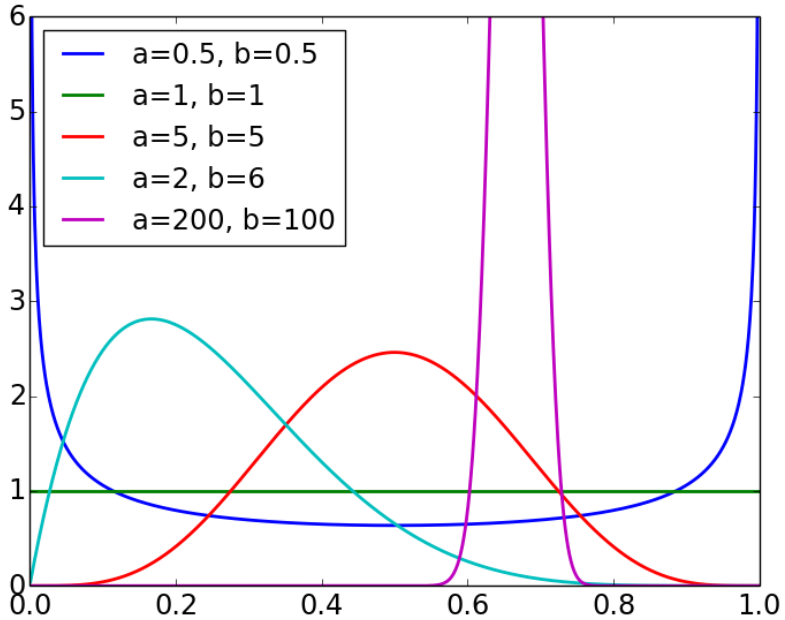
$$p(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1}.$$

  - Γ is the gamma function and has the property of $\Gamma(n) = (n-1)!$ for positive integer n.
  - 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 and Inference

- Beta distribution for various values of a, b:



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

Bayesian Parameter Estimation and Inference

- Computing the posterior distribution:

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

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

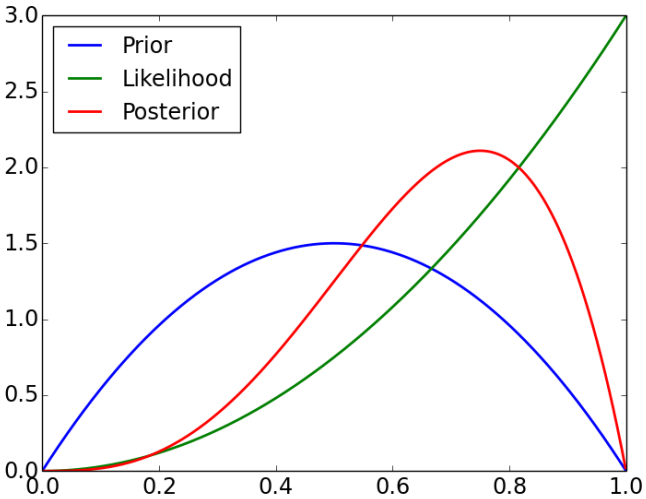
$$\mathbb{E}[\theta \mid \mathcal{D}] = \frac{N_H + a}{N_H + N_T + a + b}$$

- 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 (conjugate priors), and it is very useful in computation of posteriors.

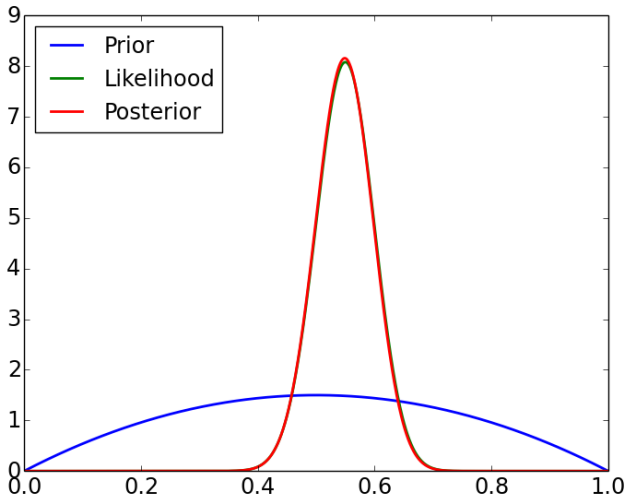
Bayesian Parameter Estimation and Inference

Bayesian inference for the coin flip example:

Small data setting
$N_H = 2,\ N_T = 0$



Large data setting
$N_H = 55,\ N_T = 45$



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

Bayesian Parameter Estimation and Inference

- 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):

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

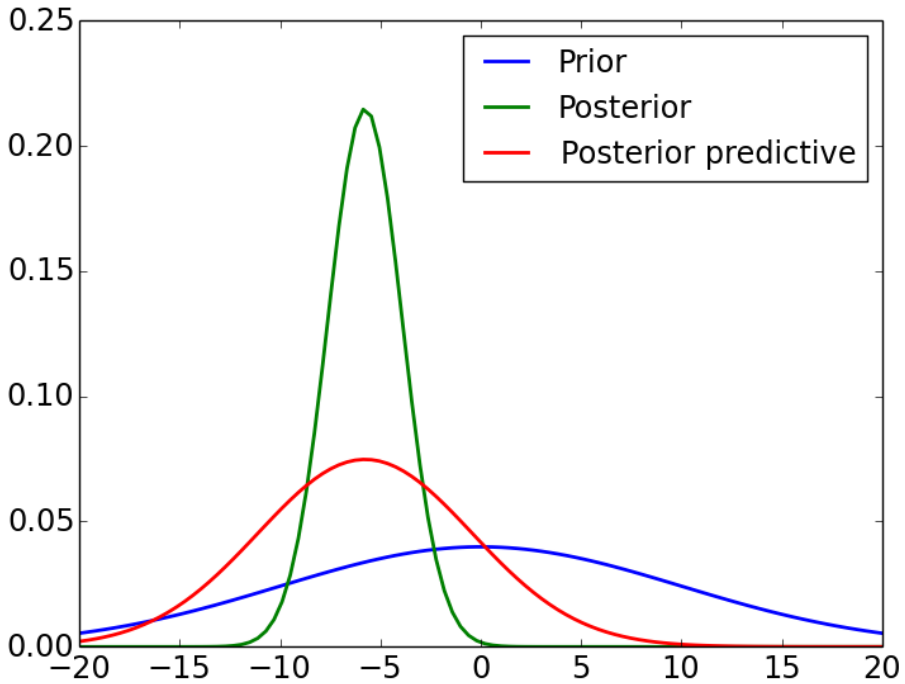
- For the coin flip example:

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

Bayesian Parameter Estimation and Inference

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 (derivation omitted)
- Why is the posterior predictive distribution more spread out than the posterior distribution?



Comparison of MLE and Bayesian Estimation

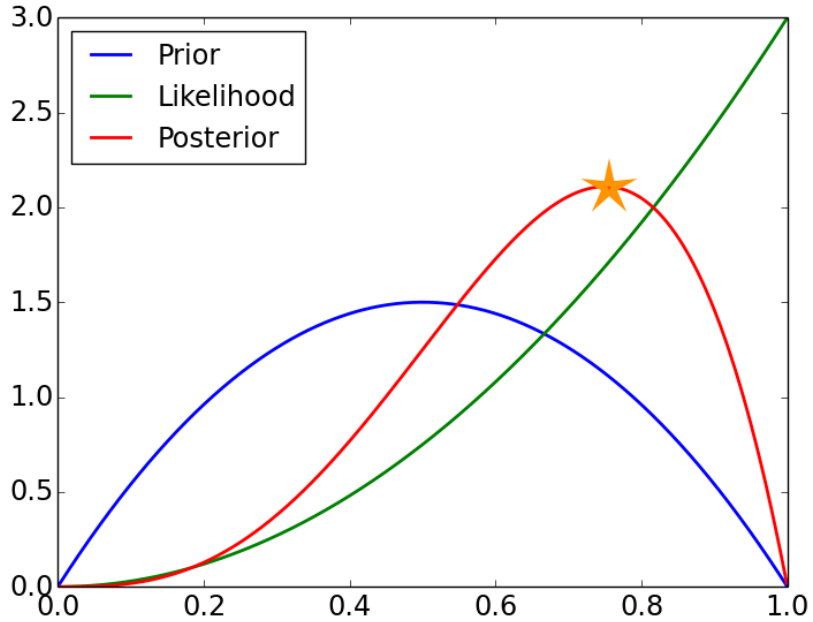
- The Bayesian approach deals better with data sparsity
- The Bayesian approach allows us to incorporate prior knowledge in some cases.
  - But choosing prior is not always easy.
  - Q: How do we incorporate our prior knowledge in MLE?

Comparison of MLE and Bayesian Estimation

- 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.

Maximum A-Posteriori Estimation

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



- This is an approximation of the full Bayesian estimation and inference, because it only finds one parameter instead of having a probability distribution over them.

Maximum A-Posteriori Estimation

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

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

Maximum A-Posteriori Estimation

- Joint probability in the coin flip example:
  Recall that the posterior is the Beta distribution:

$$p(\boldsymbol{\theta} \,|\, \mathcal{D}) \propto p(\boldsymbol{\theta}) p(\mathcal{D} \,|\, \boldsymbol{\theta}) \propto \left[\theta^{a-1}(1-\theta)^{b-1}\right] \left[\theta^{N_H}(1-\theta)^{N_T}\right],$$

  so we get that

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

- 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,

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

Comparison: MLE, MAP, and Bayesian

Comparison of estimates in the coin flip example:

	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	$\frac{55}{100} = 0.55$
$\mathbb{E}[\theta\|\mathcal{D}]$	$\frac{N_H+a}{N_H+N_T+a+b}$	$\frac{4}{6} \approx 0.67$	$\frac{57}{104} \approx 0.548$
$\hat{\theta}_{\text{MAP}}$	$\frac{N_H+a-1}{N_H+N_T+a+b-2}$	$\frac{3}{4} = 0.75$	$\frac{56}{102} \approx 0.549$

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

Gaussian Discriminant Analysis

Gaussian Discriminant Analysis

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

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

- Naïve Bayes used a conditional independence assumption to get

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

  What else could we do?
  - Choose a simple distribution.
- Next, we will discuss fitting Gaussian distributions to our data.

Bayes Classifier

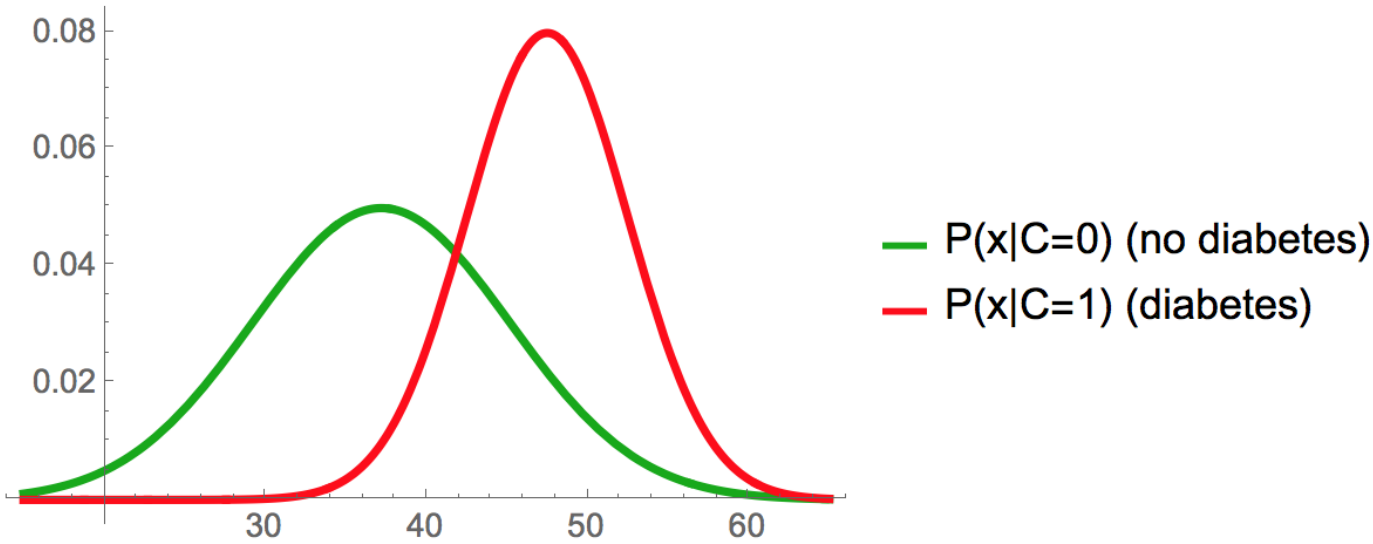
- Let's take a step back.
- Bayes Classifier

$$h(\mathbf{x}) = \operatorname*{argmax}_k p(t = k|\mathbf{x}) = \operatorname*{argmax}_k \frac{p(\mathbf{x}|t = k)p(t = k)}{p(\mathbf{x})}$$
$$= \operatorname*{argmax}_k p(\mathbf{x}|t = k)p(t = k)$$

- We previously talked about discrete $\mathbf{x}$. What if $\mathbf{x}$ is continuous?

Classification: Diabetes Example

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



- How can we model $p(x|t = k)$?
  - Multivariate Gaussian

Multivariate Data

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

$$\mathbf{X} = \begin{bmatrix} [\mathbf{x}^{(1)}]^\top \\ [\mathbf{x}^{(2)}]^\top \\ \vdots \\ [\mathbf{x}^{(N)}]^\top \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}$$

Multivariate Parameters

- Mean

$$\mathbb{E}[\mathbf{x}^{(i)}] = \boldsymbol{\mu} = [\mu_1, \cdots, \mu_d]^\top \in \mathbb{R}^D$$

- Covariance

$$\boldsymbol{\Sigma} = \mathbf{Cov}\left(\mathbf{x}^{(i)}\right) = \mathbb{E}[(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1D} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{D1} & \sigma_{D2} & \cdots & \sigma_D^2 \end{bmatrix}$$

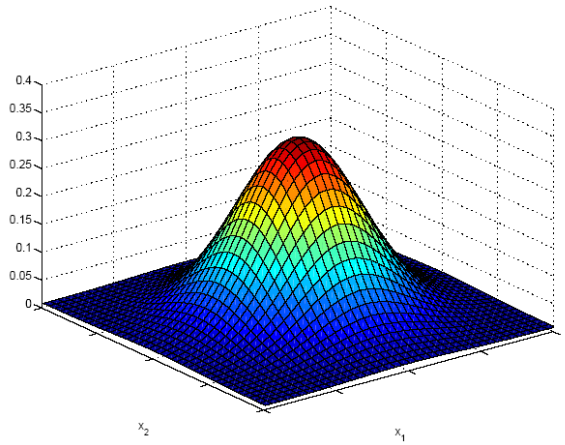
- The mean and covariance are enough to represent a Gaussian distribution. This is not true for all distributions.

Multivariate Gaussian Distribution

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

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

  where $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix $\boldsymbol{\Sigma}$.



- The Central Limit Theorem says that sums of independent random variables are approximately Gaussian.
  - The r.v. do not need to be Gaussians themselves.
- We often use Gaussian distributions because they make the calculations easy.

Bivariate Normal

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

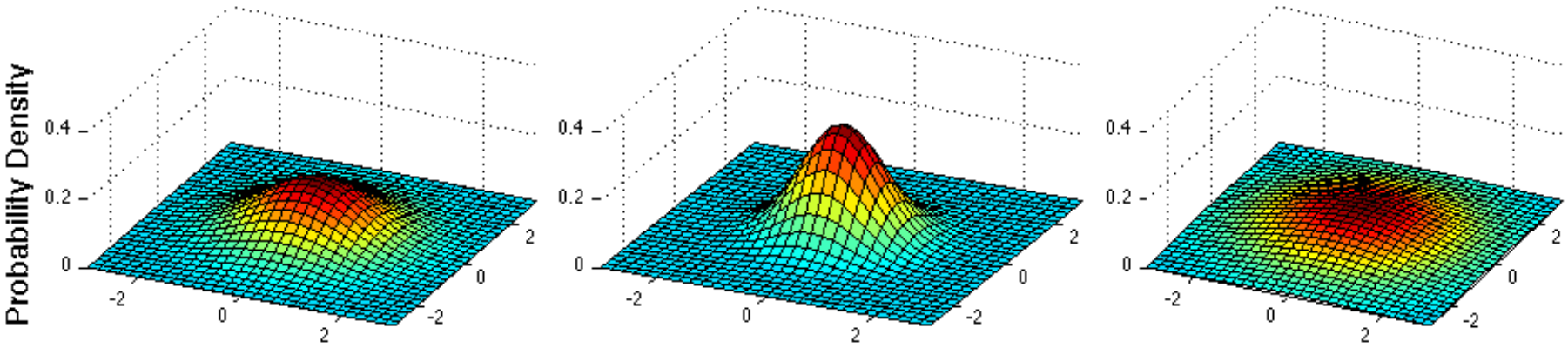


Figure: Probability density function

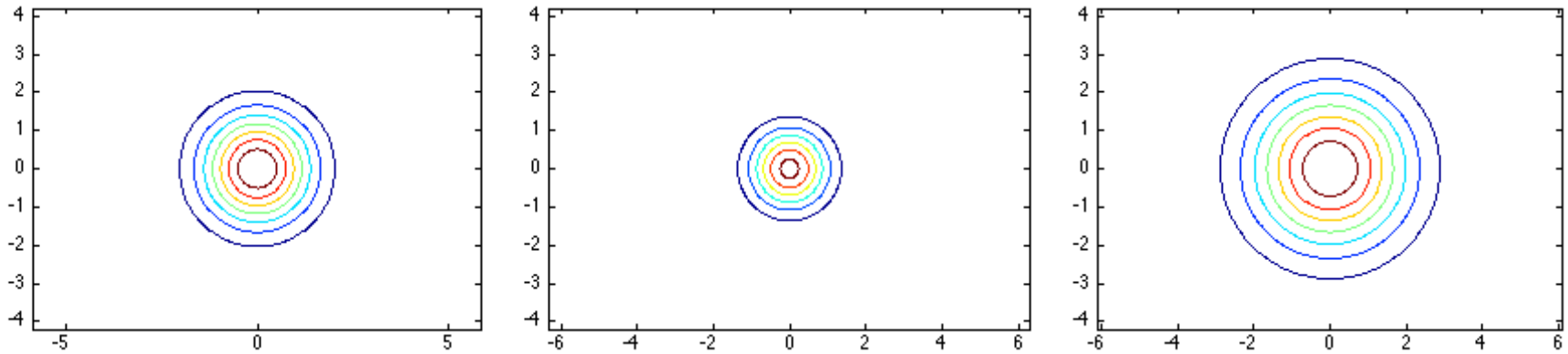


Figure: Contour plot of the pdf

Bivariate Normal

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

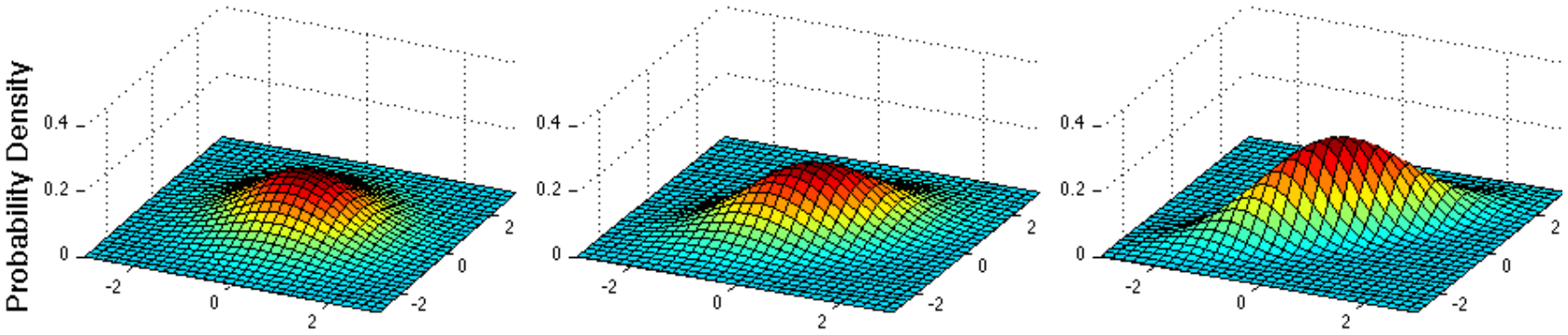


Figure: Probability density function

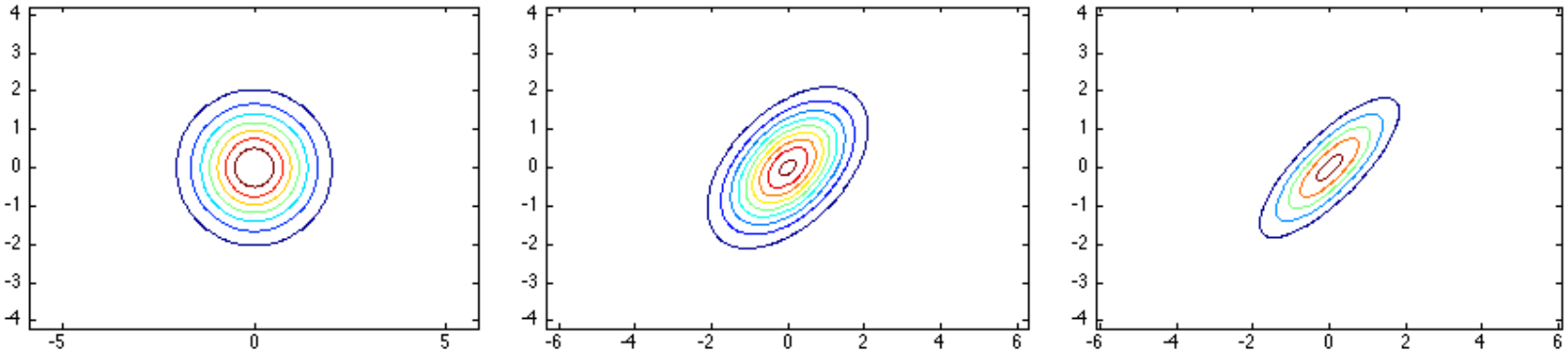


Figure: Contour plot of the pdf

Maximum Likelihood

- Suppose we want to model the distribution of highest and lowest temperatures in Toronto in March, and we've recorded the following observations :(

  (-2.5,-7.5) (-9.9,-14.9) (-12.1,-17.5) (-8.9,-13.9) (-6.0,-11.1)

- Assume they're drawn from a Gaussian distribution with mean $\boldsymbol{\mu}$, and covariance $\boldsymbol{\Sigma}$. We want to estimate these using data.
- Log-likelihood function:

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \prod_{i=1}^{N} \left[\frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \right\} \right]$$

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

$$= \sum_{i=1}^{N} \underbrace{-\log(2\pi)^{d/2}}_{\text{constant}} - \log |\boldsymbol{\Sigma}|^{1/2} - \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})$$

Maximum Likelihood

- Maximize the log-likelihood by setting the derivative to zero:

$$0 = \frac{\mathrm{d}\ell}{\mathrm{d}\boldsymbol{\mu}} = -\sum_{i=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\mu}} \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})$$
$$= -\sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0$$

- Solving we get $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$.
- This is just the sample mean of the observed values, or the empirical mean.

Maximum Likelihood

- Similar calculation for the covariance matrix Σ yields:
- Set the *partial* derivatives to zero, just like before

$$0 = \frac{\partial \ell}{\partial \boldsymbol{\Sigma}} \implies \hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top$$

- This is called the empirical covariance and comes up quite often, e.g., PCA in the next lecture.
- Derivation in multivariate case is tedious. No need to worry about it. But it is good practice to derive this in one dimension. See appendix.

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

- 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 conditioned on class $t = k$:

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

  where $|\boldsymbol{\Sigma}_k|$ denotes the determinant of the covariance matrix $\boldsymbol{\Sigma}_k$ for class k, and D is dimension of $\mathbf{x}$
- Each class k has a mean vector $\boldsymbol{\mu}_k$ and a covariance matrix $\boldsymbol{\Sigma}_k$
- $\boldsymbol{\Sigma}_k$ has $\mathcal{O}(D^2)$ parameters - could be hard to estimate

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

- GDA (GBC) decision boundary is based on class posterior $p(t_k|\mathbf{x})$. We choose a class with the highest posterior probability, i.e., $\text{argmax}_k\, p(t_k|\mathbf{x})$.
- This is equivalent to choosing $\text{argmax}_k \log p(t_k|\mathbf{x})$.
- Let us take a closer look at $\log p(t_k|\mathbf{x})$.

$$\begin{aligned}\log p(t_k|\mathbf{x}) &= \log \frac{p(\mathbf{x}|t_k)p(t_k)}{p(x)} = \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k^{-1}| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \\ &\quad + \log p(t_k) - \log p(\mathbf{x})\end{aligned}$$

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

$$\log p(t_k|\mathbf{x}) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k^{-1}| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \log p(t_k) - \log p(\mathbf{x})$$

- Where is the decision boundary between class k and $l \neq k$?
- It is where

$$\log p(t_k|\mathbf{x}) = \log p(t_l|\mathbf{x}).$$

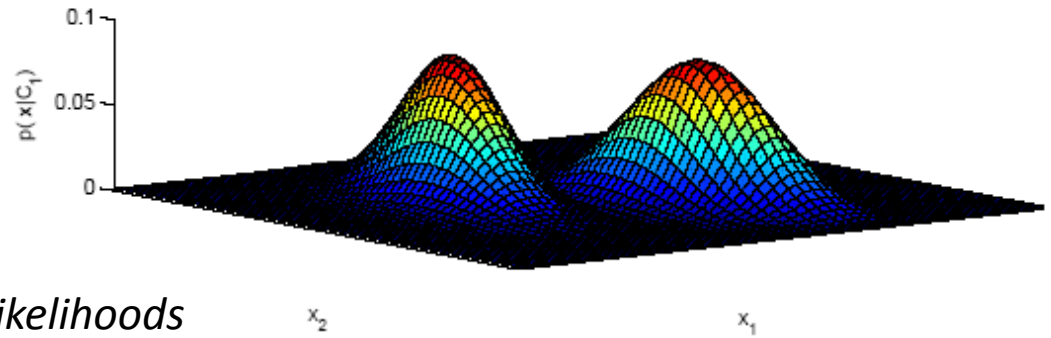
- Let us write it down

$$(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) = (\mathbf{x}-\boldsymbol{\mu}_\ell)^T\boldsymbol{\Sigma}_\ell^{-1}(\mathbf{x}-\boldsymbol{\mu}_\ell) + C_{k,l}$$

$$\mathbf{x}^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x} - 2\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x} = \mathbf{x}^T\boldsymbol{\Sigma}_\ell^{-1}\mathbf{x} - 2\boldsymbol{\mu}_\ell^T\boldsymbol{\Sigma}_\ell^{-1}\mathbf{x} + C_{k,l}$$

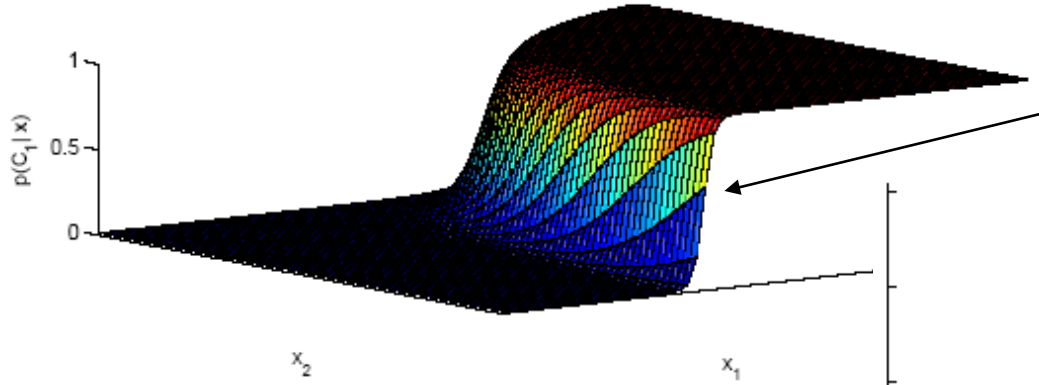
- Quadratic function in $\mathbf{x} \implies$ quadratic decision boundary
- What is $C_{k,l}$? What if $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_\ell$?

Decision Boundary

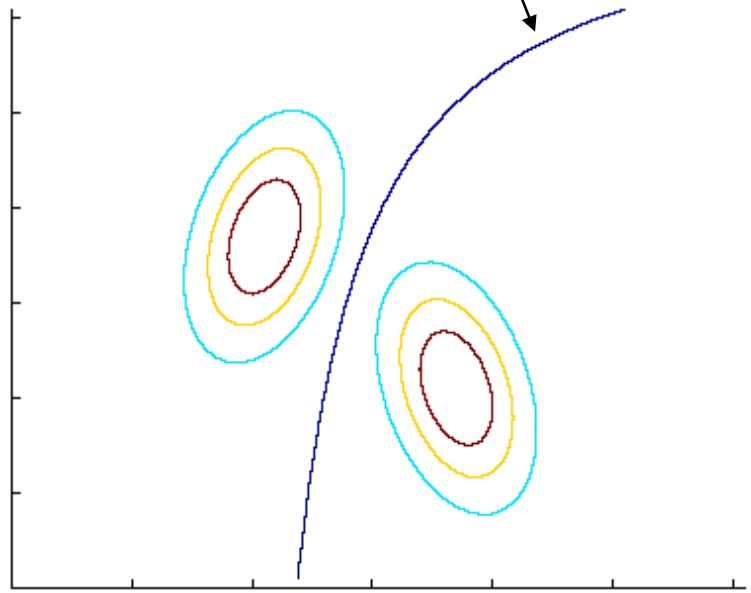


likelihoods

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



posterior for t_1



Learning

- Learn the parameters for each class using maximum likelihood
- Let us assume that we have two classes $t = \{0, 1\}$, and the prior over them is specified by a Bernoulli distribution

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

- We can compute the MLE in closed form (good exercise):

$$\hat{\phi} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}\{t^{(n)} = 1\}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{n=1}^{N} \mathbb{I}\{t^{(n)} = k\} \mathbf{x}^{(n)}}{\sum_{n=1}^{N} \mathbb{I}\{t^{(n)} = k\}}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{\sum_{n=1}^{N} \mathbb{I}\{t^{(n)} = k\}} \sum_{n=1}^{N} \mathbb{I}\{t^{(n)} = k\} (\mathbf{x}^{(n)} - \hat{\mu}_{t^{(n)}})(\mathbf{x}^{(n)} - \hat{\mu}_{t^{(n)}})^T$$

Simplifying the Model

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

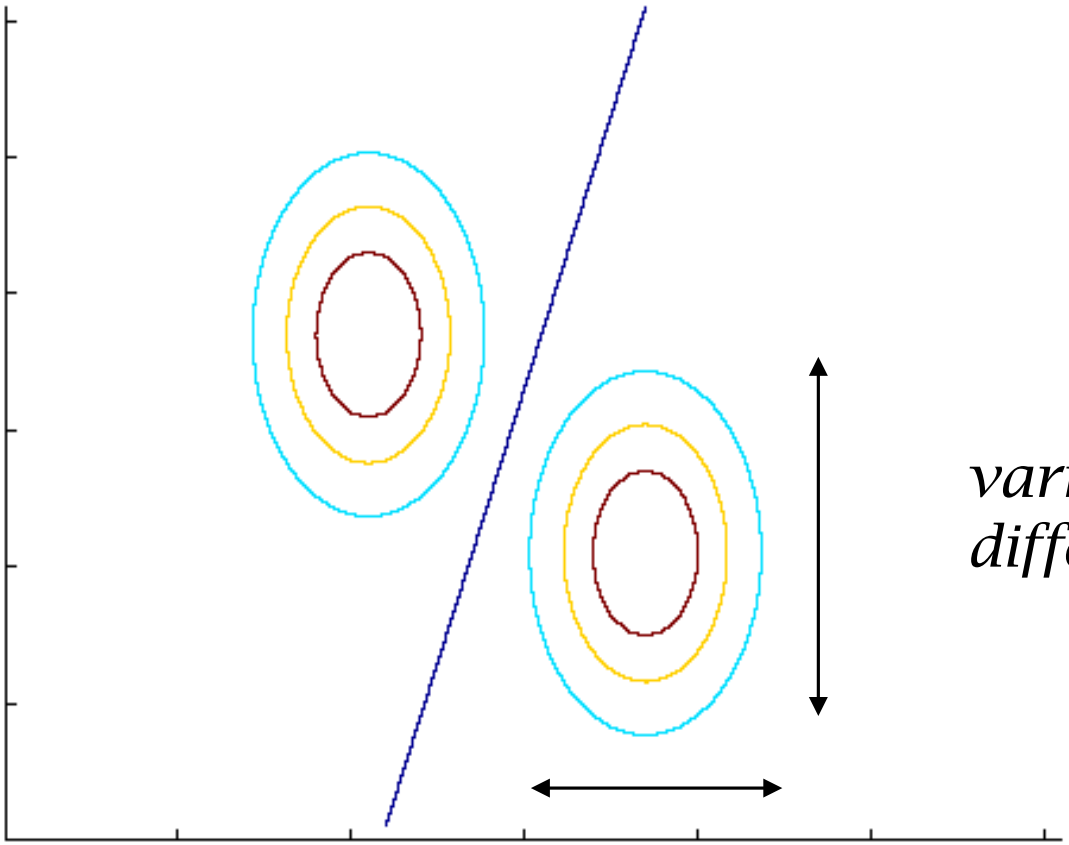
- For Gaussian Bayes Classifier, if input $\mathbf{x}$ is high-dimensional, then covariance matrix has many parameters $O(D^2)$.
- Save some parameters by using a shared covariance for the classes, i.e., $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_l$.
- Any other idea you can think of? (next lecture)
- MLE in this case:

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

- Linear decision boundary (verify this mathematically!).
- This is often called Linear Discriminant Analysis (LDA).
- Naïve Bayes for Gaussian: Assumes that the features x_i and x_j $(i \neq j)$ are independent given the class t:

$$p(\mathbf{x}|t = k) = \prod_{j=1}^{D} p(x_j|t = k)$$

Decision Boundary: Shared Variances (between Classes)



variances may be different

Gaussian Discriminative Analysis vs. Logistic Regression

- Binary classification: If you examine $p(t = 1|\mathbf{x})$ 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 $\mathbf{w}$ is an appropriate function of $(\phi, \mu_0, \mu_1, \Sigma)$, $\phi = p(t = 1)$.
- GDA is similar to logistic regression (LR), but parameter are estimated differently.
- When should we prefer GDA to LR, and vice versa?

Gaussian Discriminative Analysis vs. Logistic Regression

- GDA is a generative model, LR is a discriminative model.
- GDA makes stronger modelling assumption that the class-conditional data is a multivariate Gaussian.
- If this is true, GDA is asymptotically efficient.
- But LR is more robust, less sensitive to incorrect modelling assumptions (what loss is it optimizing?)
- When these distributions are non-Gaussian (true almost always), LR usually beats GDA
- GDA can easily handle missing features

Generative Models – Recap

- GDA has a quadratic decision boundary; LR has a linear decision boundary.
- With shared covariance, GDA leads to a model similar to logistic regression (but with different estimation procedure).
- Generative models:
  - Flexible models, easy to add/remove class.
  - Handle missing data naturally
  - More "natural" way to think about how data is generated.
- Tries to solve a hard problem in order to solve a easy problem.

Appendix: MLE for univariate Gaussian

$$0 = \frac{\partial \ell}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{i=1}^{N} \mathbf{x}^{(i)} - \mu$$

$$\begin{aligned} 0 = \frac{\partial \ell}{\partial \sigma} &= \frac{\partial}{\partial \sigma}\left[\sum_{i=1}^{N} -\frac{1}{2}\log 2\pi - \log\sigma - \frac{1}{2\sigma^2}(\mathbf{x}^{(i)} - \mu)^2\right] \\ &= \sum_{i=1}^{N} -\frac{1}{2}\frac{\partial}{\partial \sigma}\log 2\pi - \frac{\partial}{\partial \sigma}\log\sigma - \frac{\partial}{\partial \sigma}\frac{1}{2\sigma}(\mathbf{x}^{(i)} - \mu)^2 \\ &= \sum_{i=1}^{N} 0 - \frac{1}{\sigma} + \frac{1}{\sigma^3}(\mathbf{x}^{(i)} - \mu)^2 \\ &= -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \mu)^2 \end{aligned}$$

$$\hat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)}$$

$$\hat{\sigma}_{\text{ML}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \mu)^2}$$