DEPARTMENT OF ENGINEERING SCIENCE

READING NOTE

October 22, 2017

Information Theory, Inference, and Learning Algorithm

Author:

Shuqi ZHANG

Supervisors:

Prof. Michael OSBORNE

This pdf is a personal reading note for the book "Information Theory, Inference, and Learning Algorithm". The words in black are mainly from the book itself, while **the words in red are personal understanding on the contents and futher explaination in my own words.**

Contents

1 Probability, Entropy, and Inference

Probability To describe *degress of belief* in propositions that do not involve random variables.

You cannot do inference without making asumptions.

Bayesian method Bayesians also use probabilities to describe inferences.

non-Bayesian method probabilities are allowed to describe only random variables.

Probability calculations often fall into one of two categories **forward probability** and **inverse probability**.

forward probability *generative model* that describes a process that is assumed to give rise to some data, and the data allow people to compute the probability distribution or expectation of some quantity.

inverse probability *generative model* that describes a process that is assumed to give rise to some data, however, we compute the conditional probability of one or more of the unobserved variables in the process, given the observed variables. **Bayesian theorem is required.**

1.1 Terminologt of inverse probability

Inverse probability is the one applied **Bayesian theorem**.

$$P(u|n_B, N) = \frac{P(u)P(n_B|u, N)}{P(n_B|N)} \tag{1.1}$$

prior the marginal probability $P(u)$ is the *prior* probability of u.

likelihood $P(n_B|u, N)$ is called the *likelihood* of u. For fixed u, $P(n_B|u, N)$ defines a probability over n_B. For fixed n_B, $P(n_B|u, N)$ defines the likelihood of u. In predition, n_B is like the data set which is fixed, and u is the model, therefore, $P(n_B|u, N)$ is the likelihood and it is the outcomes of the believed model. *Always say "the likelihood of the parameters". The likelihood function is not a probability distribution.*

posterior $P(u|n_B, N)$ is called the *posterior* probability of u given n_B.

evidence the normalizing constant $P(n_B|N)$

$$posterior = \frac{likelihood \times prior}{evidence} \tag{1.2}$$

2 An Example Inference Task: Clustering

Regularity is to put a set of objects into groups that are similar to each other. The operation of grouping things togther is called *clustering*. If the *cluster* is further sub-divided into *sub-clusters*, it is called *hierarchical clustring*.

The motivations for clustering:

Reason 1 A good clustering has predictive power, helping people to better allocate their resources. The underlying cluster labels are meaningful, will lead to a more efficient description of our data, and will help people choose better actions. The information content of the data, $log\frac{1}{P(\mathrm{x})}$.

Reason 2 Clusters can be a useful aid to communication because they allow lossy compression. Clusters can help people identify an object as they contain sufficient information.

Vector quantizer in terms of an *assignment rule* $x \to k(x)$ for assigning datapoints x to one of K codenames, and a *reconstruction rule* $k \to m^{(k)}$. The aim is to choose the function $k(x)$ and $m^(k)$ so as to minimize the *expected distortion* for the reconstruction, which might be defined to be

$$D = \sum_x P(\mathrm{x}) \frac{1}{2} [\mathrm{m^{kx}} - \mathrm{x}]^2 \tag{2.1}$$

So the goal is to minimize D.

Reason 3 Failures of the cluster model may highlight interesting objects that deserve special attention. If something goes wrong in a vector quantizer, it is easier for people to figure out the reasons behind due to clusters.

Reason 4 Clustering algorithms may serve as models of learning processes in neural systems. The K-means algorithm is an example of a *competitive learning* algorithm. The algorithm works by having the K clusters compete with each other for the right to own the data points.

2.1 K-means clustering

An algorithm for putting *N* data points in an *I*-dimensional space into *K* clusters. Each cluster is parameterized by a vector m^k which is its mean. The data points are denoted by x^n and n means N data

points. Each vector x has I components x_i, as it is I-dimension.

The K-means clustering algorithm:

Initialization. Set K means m^k to random values.

Assignment step. Each data points n is assigned to the nearest mean. We denote our guess for the cluster k^n that the point x^n belongs to by $\hat{k}^n$.

$$\hat{k}^n = \underset{x}{\operatorname{argmin}}\{d(m^k, x^n)\} \tag{2.2}$$

The equation 2.2 about is trying to minimize the distance between the assumed means and the actual data points.

$$responsibilities,\ r_k^n = \begin{cases} 1, & \text{if } \hat{k}^n = k. \\ 0, & \text{if } \hat{k}^n \neq k. \end{cases} \tag{2.3}$$

If mean k is the closest mean to datapoint, the responsibility is equal to 1, therefore $\hat{k}^n$ is equal to k. It means that the assumed cluster $\hat{k}^n$ is actually the correct cluster. However, if not, then the responsibility is zero. If r is 1, then the corresponding $\hat{k}^n$ is the winning k.

Update step. m^k is updated to match the means of the data points.

$$m^k = \frac{\sum_n r_k^n x^n}{R^k} \tag{2.4}$$

where R^k is the total responsibility of mean k, and the sum is for each data point 's responsibility and the data value for to corresponding cluster.

$$R^k = \sum_n r_k^n \tag{2.5}$$

Repeat the assignment step and update step. Keep repeating the loop untill the assginment step gives the mean is the cloest one.

Mean m has been assigned with a random value, and the minimum $\hat{k}$ has been found for each k k^n *means that the* k *which is the cluster has the same number as the data points* n. As the original m is a random value, it is updated by the equation 2.7.

Personal understanding of K-means algorithm There are two inputs in the K-means algorithm: $K \rightarrow$ number of cluster; $x_i \rightarrow$ represents the set of points with i ranges from 1 to n. At the beginning of the process, centroids C_j, where i is equal to the number of clusters, are placed at random locations. For each data points x_i, $argmin$ has been used to compare the distance between each data point and each centroid, generating the nearest centroid C_j. The data point x_i is assgined to the cluster C_j individually. For example, if there are two centroids, all the data points have been assigned to two clusters based on their distances. After assignment, the position or value of clusters will be recomputed according to the positions or values of the data points belong to them. Equation 2.7 shows that the update only related to the points belong to the cluster as responsibility, r, needs to be 1 to be meaningful in the calculation. By taking the average of the positions of cluster 's data points, the new cluster 's position is computed. The loop of assigment of data points to clusters and update of clusters keeps running untill the assigment does not change.

1. Data points have been assigned to clusters.
2. Clusters have been updated based on their data points.
3. Data points are assigned again based on the updated clusters.
4. Clusters are updated as the data points 'assigments have been changed.
5. Keep assigning and updating untill assigment does not change.
6. The number of iterations is related to the clusters 'initial locations.

The algorithm is based on distance between clustes and data points (distance between the means and the data points), so the algorithm may igore the general patterns, ending up with some points may be incorrectly assigned to the wrong clusters. *(refer to the book Figure 20.5 and Figure 20.6)* The K-means algorithm has no representation of the weight or breadth of each cluster. Consequently, data points that actually belong to the broad cluster are incorrectly assigned to the narrow cluster. **The K-means algorithm has no way of representing the size or shape of a cluster.** This algoritm is rather too hard as all points assigned to a cluster are equals in that cluster. However, some points are closer to the centroid while some are placed near the border between two or more clusters should play a partial role in determining the locations of all the clusters that they could plausibly be assigned to *(further determination befer jumping into conclusion recklessly)*.

2.2 Soft K-means clustering

β is inputted for people to term the *stiffness*. The stiffness β is an inverse-length-squared, so it can be associated a lengthscale, $\theta \equiv 1/\sqrt{\beta}$.

Assignment step. Each data point x^n is given a soft 'degree of assignment' to each of the means *(each of the centroids)*. We call the degree to which x^n is assigned to cluster k the responsibility r_k^n. The β is considered into the calculation of responsibility, r.

$$r_k^n = \frac{exp(-\beta d(m^k, x^n))}{\sum_{k'} exp(-\beta d(m^{k'}, x^n))} \tag{2.6}$$

The sum of the K responsibilities for the nth point is 1.

Updated step. The clusters 'positions are updated by the same method as the normal K-mean algorithm.

$$m^k = \frac{\sum_n r_k^n x^n}{R^k} \tag{2.7}$$

where R^k is the total responsibility of mean k, and the sum is for each data point 's responsibility and the data value for to corresponding cluster.

$$R^k = \sum_n r_k^n \tag{2.8}$$

The responsibility, r_k^n, can take values between 0 and 1, therefore, the points assigned to a specific cluster are not equal. **Therefore, weights have added to the data points assigned to a cluster. The points closer to the centroid are 1 and the points futher or in between are from 1 to 0, but still assigned to the same cluster. When the centroid is updated, different points have different weights in the new postion calculation, so the new cluster can be placed more cenral to the correct points.**

3 Exact Inference by Complete Enumeration

The phenomenon, that one of the possible causes ($b=1$) of some data (another action, a) becomes *less* probable when another of the causes ($c=1$) becomes *more* probable, even though those two causes are independent varibales a *prior*, is known as *explaining away*.

This kind of problems is solved byenumerating all hypotheses about the variables (b,c), finding their posterior probabilities, and marginalizing to obtain the required inferences about one, $(borc)$ over the other, $(corb)$.

3.1 Exact inference for continuous hypothesis spaces

Many of the hypothesis psaces we will consider are naturally thought of as contiuous. For example, standard deviation of a Gaussian μ,δ live in a continuous two-dimensional space. In computer implementation, such continuous spaces will necessaryily be discretized or be enumerated at a grid of parameter values.

A two-parameter model The one-dimensional Gaussian distribution is parameterized by a mean μ and a standard deviation δ:

$$P(x|\mu,\delta) \equiv Normal(x;\mu,\delta^2) \tag{3.1}$$

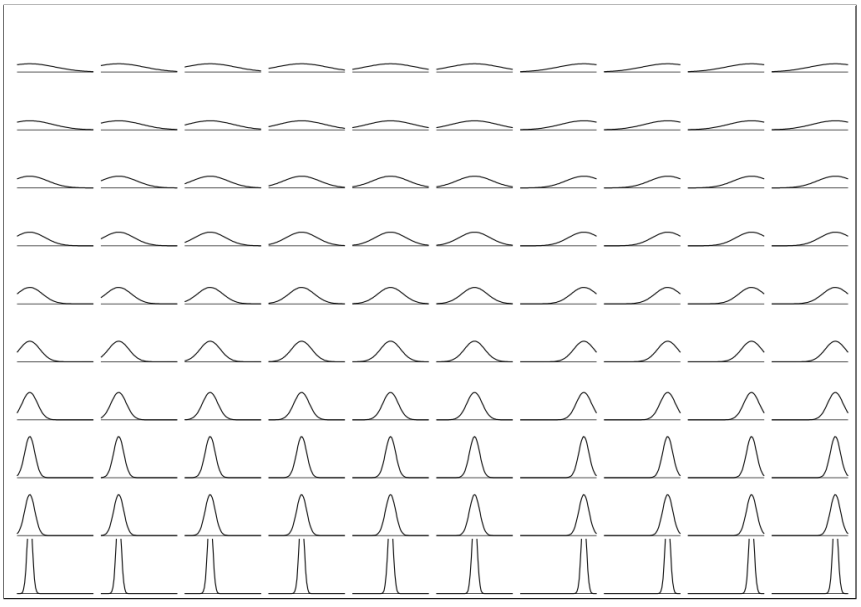

Figure 3.1: Enumeration of an entire (discretized) hypothesis space for one Gaussian with parameters μ (horizontal axis) and δ (vertical).

Figure 3.1 shows that hypotheses are evenly spaced in a ten by ten square grid covering ten values of μ nd ten values of δ. The inference of μ and δ are examined given data points x_n for n ranging for 1 to N.The model is discretized in two-dimension and the inference is examnied by the data points. The data points can be placed in the inference to find the maximum likelihood. The line thikcness, figure 3.3, can be used to encode the value of the likelihood and the sub-hypotheses with likelihood smaller than e^{-8} times the maximum likelihood are deleted.

Figure 3.2: Five datapoints.

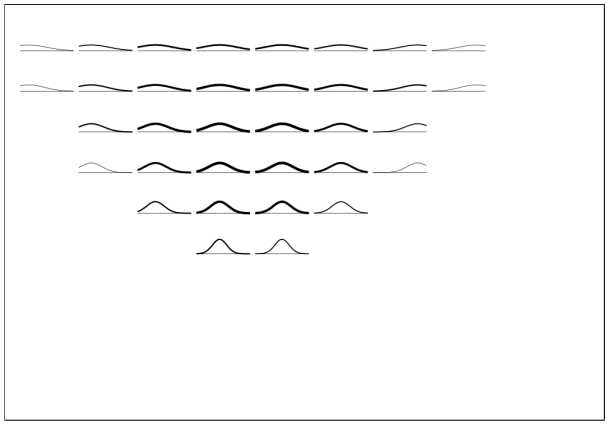

Figure 3.3: Likelihood function from figure 3.1, given the data of figure 3.2, represented by line thickness.

A five-parameter mixture model A mixture of two Gaussian to represent the points in figure 3.2

$$P(x|\mu_1,\delta_1,\pi_1,\mu_2,\delta_2,\pi_2) \equiv Normal(x;\mu_1,\delta_1^2,\pi_1,\mu_2,\delta_2^2,\pi_2) \tag{3.2}$$

Where π is weighting between two Gaussian distributions. Because $\pi_1 + \pi_2 = 1$, it only represents one dimension. There are five dimensions in total, $\mu_1,\delta_1,\mu_2,\delta_2,\pi$.

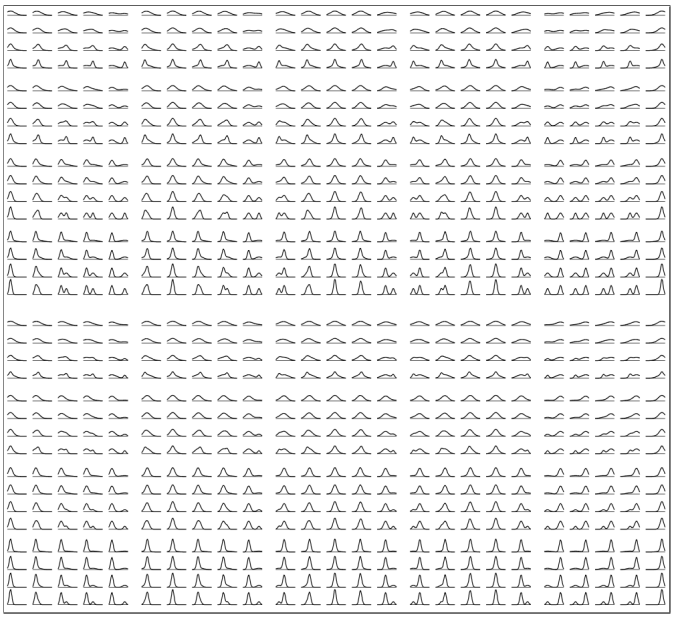

Figure 3.4: Enumeration of the entire (discretized) hypothesis space for a mixture of two Gaussians. Weight of the mixture components is $\pi_1, \pi_2 = 0.6, 0.4$ in the top half and 0.8, 0.2 in the bottom half. Means $\mu_1 and \mu_2$ vary horizontally, and standard deviations $\delta_1 and \delta_2$ vary vertically.

For the first row, there are five blocks. μ_2 changes evenly in horizontal direction, while $\delta_2 and \pi$ are fixed. Within each block, μ_1 changes evenly in horizontal direction and δ_1 changes evenly in vertical direction. Same rule applies to whole figure 3.4.

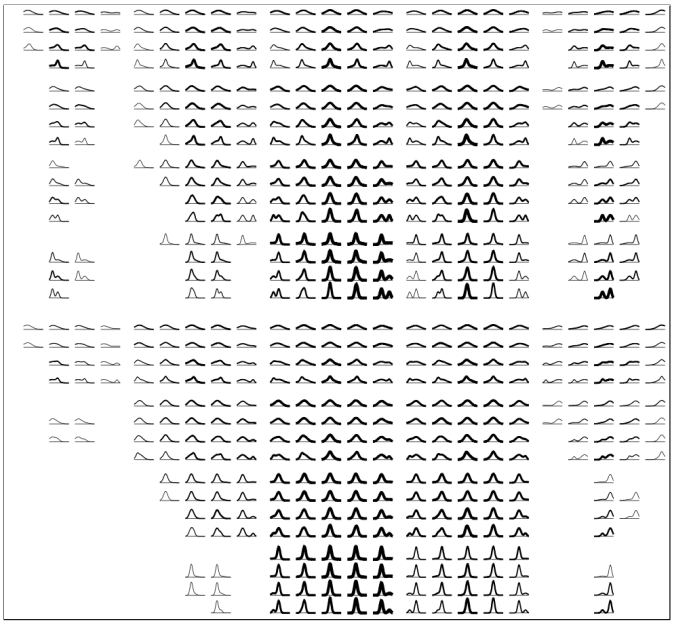

Figure 3.5: Inferring a mixture of two Gaussians. Likelihood function, given the data points, represented by line thikcness.

Models can be compared by the models 'posterior probabilities *(found by evaluating the marginal*

likelihood or evidence for each model $H, P(x|H)$*).* It is same as finding marginal over one parameter and compare the posterior of the model over that parameter. Moreover, normally, a grid of at least 10^k is required for the enumeration way to solve K parameters model. Completing enumeration is rarely a feasible computational strategy due to the exponential growth of computation.

4 Maximum Likelihood and Clustering

Comparing to enumerate all hypotheses, homing in on one good hypothesis that fits the data well saves a lot of time. It is supported by the maximum likelihood method. It is easier to solve with the *logarithm* of the likelihood as likelihoods multiple; log likelihoods add.

4.1 Maximum likelihood for one Gaussian

For data $x_n{}_{n=1}^{N}$, the log likelihood is:

$$\ln[P((x_n)_{n=1}^{N}|\mu,\delta)] = -N\ln(\sqrt{2\pi}) - \sum_n \frac{(x_n-\mu)^2}{2\mu^2} \tag{4.1}$$

Where μ is the maximum likelihood meand of the Gaussian. The log can be xpressed in terms of two functions of the data, the sample mean, $\bar{x}$,

$$\bar{x} \equiv \sum_{n=1}^{N} \frac{x_n}{N}, \tag{4.2}$$

and the sum of deviations

$$S \equiv \sum_n (x_n - \bar{x})^2 : \tag{4.3}$$

$$\ln[P((x_n)_{n=1}^{N}|\mu,\delta)] = -N\ln(\sqrt{2\pi}) - \frac{N(\mu-\bar{x})^2 + S}{2\mu^2} \tag{4.4}$$

Do not quite understand how to derive the equation 4.4 from equation 4.2, equation 4.3 and the original form, equation 4.1.

The likelihood, equation 4.4, only depends on $\bar{x}$ and S, so these two quantities are known as *sufficient statistics*.

$$\frac{\partial}{\partial\mu}\ln P = -\frac{N(\mu-\bar{x})}{\delta^2} \tag{4.5}$$

$$= 0 \qquad \text{when } \mu = \bar{x} \tag{4.6}$$

So $\ln P$ reaches maximum when the condition displayed above satisfied.

Quadratic approximation is used to estimate how far from the maximum-likelihood parameter before the likelihood falls by, $e^{1/2} or e^{4/2}$.

The error bars on μ is found by calcualting the second derivative of $\ln P$ over μ and square root the product of $-1 \times theresult$. The error bars on δ are similar. *The size of error bars is the distance to the average value.* From eqation 4.6, we know that for fixed δ, we can maximize the likelihood by equalling $\mu = \bar{x}$. By differentiate $\ln P$ over $\ln\delta$, and at the time $\mu = \bar{x}$ for the maximum likelihood, we have $\delta = \sqrt{S/N}$. Therefore, $\mu, \delta_M L = \bar{x}, \delta_N = \sqrt{S/N}$ jointly maximize the likelihood.

4.2 Maximum likelihood for a mixture of Gaussians

A random variable x is assumed to have a probability distribution that is a $mixture of two Gaussians$,

$$P(x|\mu_1, \mu_2, \delta) = [\sum_{k=1}^{2} p_k \frac{1}{\sqrt{2\pi\delta^2}} \exp(-\frac{(x-\mu_k)^2}{2\delta^2})] \tag{4.7}$$

$k = 1, k = 2$ represent there are two Gaussians and the prior of the class label k is $p_1 = p_2 = \frac{1}{2}$; μ_k are the means of the two Gaussians and both have standard deviation δ.

Assuming $\theta \equiv \mu_k, \delta$ are know, the posterior probability of the class lable k_n of the nth point can be written as

$$\begin{aligned} P(k_n = 1|x_n, \theta) &= \frac{P(x_n|k_n = 1, \theta) P(k_n = 1|\theta)}{P(x_n|\theta)} \\ &= \frac{p_1 \frac{1}{\sqrt{2\pi\delta^2}} \exp(-\frac{(x-\mu_1)^2}{2\delta^2} p_1)}{P(x|\mu_1, \mu_2, \delta)} \\ &= \frac{\frac{1}{2}\exp(-\frac{(x-\mu_1)^2}{2\delta^2})\frac{1}{2}}{\frac{1}{2}\exp(-\frac{(x-\mu_1)^2}{2\delta^2}) + \frac{1}{2}\exp(-\frac{(x-\mu_2)^2}{2\delta^2})} \\ &= \frac{\frac{1}{2}}{1 + \exp(-\frac{(x-\mu_1)^2}{2\delta^2} + \frac{(x-\mu_2)^2}{2\delta^2})} \\ &= \frac{\frac{1}{2}}{\exp(-(2(\mu_1 - \mu_2)x + (\mu_2^2 - \mu_1^2)))} \\ &= \frac{1}{1 + \exp(-(w_1 x_n + w_0))} \end{aligned} \tag{4.8}$$

similarly,

$$P(k_n = 2|x_n, \theta) = \frac{1}{1 + \exp((w_1 x_n + w_0))} \tag{4.9}$$

In order to find the μ_k that maximize the likelihhod,

$$P(x_n{}_n^N = 1|\mu_k, \delta) = \prod_n P(x_n|\mu_k, \delta) \tag{4.10}$$

Therefore, the $\ln$ of the likelihood can turn the product into sum and the derivative of the log likelihood with respect to μ_k is give by

$$\frac{\partial}{\partial \mu_k} L = \sum_n p_{k|n} \frac{x_n - \mu_k}{\delta^2} \text{ ,and } p_{k|n} \equiv P(k_n = k|x_n, \theta) \tag{4.11}$$

Neglecting μ_k in $p_{k|n}$,

$$\frac{\partial^2}{\partial^2 \mu_k} L = -\sum_n p_{k|n} \frac{1}{\delta^2} \tag{4.12}$$

Finding zeros for the first derivative and the second derivative gives the values for μ_k that maximize the likelihood.

Don 's understand the contour plot of the likelihood functions for 32 points on page 303 and the solution on page 310.

4.3 Enhancements to soft K-means

In previous chapters, we assumed that for the solt K-means, there are sizes assigned to clusters, allowing accurate modelling of data from clusters of unequal weights during the position updates of centroids. In this case here, we can change the size of the cluster during each update.

Spherical Gaussians can be fitted to data points and the variance of the Gaussian is the same in all directions.

Assignment step. The responsibilities are

$$r_k^{(n)} = \frac{\pi_k \frac{1}{(\sqrt{2\pi}\delta_k)^I} \exp(-\frac{1}{\delta_k^2} d(\mathrm{m}^{(k)}, \mathrm{x}^{(n)})}{\sum_{k'} \pi_k \frac{1}{(\sqrt{2\pi}\delta_{k'})^I} \exp(-\frac{1}{\delta_{k'}^2 d(\mathrm{m}^{(k')}, \mathrm{x}^{(n)})})} \tag{4.13}$$

where I is the dimensionality of x.

Update step.

$$\mathrm{m}^{(k)} = \frac{\sum_n r_k^{(n)} \mathrm{x}^{(n)}}{R^{(k)}} \tag{4.14}$$

$$\delta_k^2 = \frac{\sum_n r_k^{(n)} (\mathrm{x}^{(n)} - \mathrm{m}^{(k)})^2}{IR^{(k)}} \tag{4.15}$$

$$\pi_k = \frac{R^{(k)}}{\sum_k R^{(k)}} \tag{4.16}$$

Where $R^{(k)}$ is the total responsibility of the mean k.

However, the shape of the clusters is fixed as the variance is the same. Therefore, for better fit the data points, the clusters are modelled by axis-aligned Gaussians with possibly-unequal variances. Both the size and the shape of the cluster change during update every iterations. *For better examples, pls check the book page 304 - 305.*

4.4 A fatal flaw of maximum likelihood

Overfitting Put one cluster exactly on one data point and let it 's variance go to zero (the size of the clutser is infinite in this case and it 's inverse of the variance), you can obtain an arbitrarily large likelihood, containing all the data points. However, it is not very likely to happen.

Other than overfitting, the maximum of the likelihood is often unrepresentative in high-dimensional problems as the number of likelihood is infinitly large in high dimensions.

- 1112