DATA8015: Mathematical Foundation of Data Science **Fall 2025**

Part 2.2 Statistical Inference and Learning

Scriber: Yingyu Liang

1 Maximum Likelihood Estimation (MLE)

Statistical inference refers to the process of using sample data to draw conclusions about an entire distribution. There are four typical statistical inference paradigms: point estimation, hypothesis testing, interval estimation, and Bayesian inference. We have touched upon the basis of Bayesian inference (Bayes' rule). Here, we will review point estimation, which is arguably the most common paradigm used in data science applications.

First, recall the notion of estimation. Suppose we are interested in a quantity $\boldsymbol{\theta}$ of the distribution of some random variable/vector X. For example, $\boldsymbol{\theta}$ can be the parameters of the distribution, or the variance of X. Given samples $x_1, x_2, \cdots, x_n$ from the distribution, an estimator $\widehat{\boldsymbol{\theta}}(\cdot)$ is a mapping that takes samples as input and outputs a value to approximate $\boldsymbol{\theta}$. So the estimator $\widehat{\boldsymbol{\theta}}$ can be regarded as a computational method that estimates the value of $\boldsymbol{\theta}$ using samples. From here, we will assume the samples are independent and identically distributed (iid).

Maximum likelihood is by far the most popular technique for deriving estimators.

Definition 1 (Likelihood)**.** If $x_1, x_2, \cdots, x_n$ are iid samples from a distribution with probability density function (pdf) or probability mass function (pmf) $p(x|\boldsymbol{\theta})$, the likelihood function is defined by:

$$L(\boldsymbol{\theta}|x_1, \cdots, x_n) = p(x_1, x_2, \cdots, x_n|\boldsymbol{\theta}) := \prod_{i=1}^{n} p(x_i|\boldsymbol{\theta}).$$

Definition 2 (Maximum Likelihood)**.** For each set of $x_1, \cdots, x_n$, let $\widehat{\boldsymbol{\theta}}(x_1, \cdots, x_n)$ be a value at which the likelihood $L(\boldsymbol{\theta}|x_1, \cdots, x_n)$ attains its maximum as a function of $\boldsymbol{\theta}$ with $x_1, \cdots, x_n$ held fixed, i.e.,

$$\widehat{\boldsymbol{\theta}}(x_1, \cdots, x_n) \in \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ L(\boldsymbol{\theta}|x_1, \cdots, x_n).$$

The mapping $\widehat{\boldsymbol{\theta}}$ is called the maximum likelihood estimator (MLE) of $\boldsymbol{\theta}$.

Example: Normal MLE. Let $x_1, \cdots, x_n$ be iid samples from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, with both μ and σ^2 unknown. Then

$$L(\mu, \sigma^2|x_1, \cdots, x_n) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}}.$$

Note that maximizing likelihood is equivalent to maximizing the logarithm of the likelihood (log-likelihood). Usually, it is easier to handle the log-likelihood:

$$\log L(\mu, \sigma^2|x_1, \cdots, x_n) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2.$$

To get the maximum, set the partial derivatives to 0 to get the stationary points:

$$\frac{\partial}{\partial \mu} \log L(\mu, \sigma^2 | x_1, \cdots, x_n) = \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu),$$
$$\frac{\partial}{\partial \sigma^2} \log L(\mu, \sigma^2 | x_1, \cdots, x_n) = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2.$$

Solving these yields:

$$\widehat{\mu} = \bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i,$$
$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}).$$

One can verify that this is, in fact, the global optimum.

Example: Restricted range MLE. Let $x_1, \cdots, x_n$ be iid from a normal distribution $\mathcal{N}(\mu, 1)$, and it is known that μ must be nonnegative. With no restriction on μ, we saw the MLE of μ is $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$. However, if the average $\bar{x}$ is negative, it will be outside the range of the parameter.

If $\bar{x}$ is negative, the likelihood is decreasing in μ for $\mu \geq 0$ and thus is maximized at $\widehat{\mu} = 0$. Hence, in this case, the MLE of μ is

$$\widehat{\mu} = \begin{cases} \bar{x} & \text{if } \bar{x} \geq 0, \\ 0 & \text{if } \bar{x} < 0. \end{cases}$$

Example: Bernoulli MLE. Let $x_1, \cdots, x_n$ be iid from Bernoulli(p). Then

$$L(p | x_1, \cdots, x_n) = \prod_{i=1}^{n} p^{x_i} (1-p)^{1-x_i} = p^y (1-p)^{n-y},$$
$$\log L(p | x_1, \cdots, x_n) = y \log p + (n-y) \log(1-p).$$

where $y = \sum_i x_i$.

If $0 < y < n$, differentiating $\log \log L(p | x_1, \cdots, x_n)$ and setting the result to 0 gives the solution $\widehat{p} = y/n$, and it is easy to verify that y/n is the global maximum.

If $y = 0$ or $y = n$, then

$$\log L(p | x_1, \cdots, x_n) = \begin{cases} n \log(1-p) & \text{if } y = 0, \\ n \log p & \text{if } y = n. \end{cases}$$

In either case, $\log L(p | x_1, \cdots, x_n)$ is a monotone function of p, and it is again easy to verify that $\widehat{p} = y/n$. Thus, the MLE is $\widehat{p} = \sum_i x_i / n$.

2 Maximum A Posteriori Estimation (MAP)

Suppose we have some prior knowledge about $\boldsymbol{\theta}$, represented by a prior distribution $p(\boldsymbol{\theta})$. Then the posterior of $\boldsymbol{\theta}$ after seeing samples $x_1, \cdots, x_n$, by Bayes' rule, is

$$p(\boldsymbol{\theta}|x_1, \cdots, x_n) = \frac{p(x_1, \cdots, x_n|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(x_1, \cdots, x_n)}.$$

To get a point estimation, take the maximizer of the posterior:

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}(x_1, \cdots, x_n) &\in \operatorname*{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|x_1, \cdots, x_n) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} p(x_1, \cdots, x_n|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|x_1, \cdots, x_n)p(\boldsymbol{\theta}).\end{aligned}$$

Definition 3 (Maximum A Posteriori)**.** Given a prior $p(\boldsymbol{\theta})$, let

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}(x_1, \cdots, x_n) \in \operatorname*{argmax}_{\boldsymbol{\theta}} \; L(\boldsymbol{\theta}|x_1, \cdots, x_n)p(\boldsymbol{\theta}).$$

This mapping $\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}$ is called the maximum a posteriori estimator (MAP) of $\boldsymbol{\theta}$ (with respect to the prior $p(\boldsymbol{\theta})$).

Example: Normal MAP. Let $x_1, \cdots, x_n$ be iid samples from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, with only μ unknown. Let the prior distribution of μ be $\mathcal{N}(\mu_0, \sigma_0^2)$. Then

$$p(\mu)L(\mu, \sigma^2|x_1, \cdots, x_n) = \frac{1}{(2\pi\sigma_0^2)^{1/2}} e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}} \cdot \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (x_i-\mu)^2}{2\sigma^2}}.$$

Maximizing this is equivalent to maximizing the following function of μ:

$$\sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma}\right)^2 + \left(\frac{\mu - \mu_0}{\sigma_0}\right)^2.$$

So the MAP estimator of μ is:

$$\widehat{\mu}_{\mathrm{MAP}} = \frac{\sigma_0^2 n}{\sigma_0^2 n + \sigma^2}\left(\frac{1}{n}\sum_{i=1}^n x_i\right) + \frac{\sigma^2}{\sigma_0^2 n + \sigma^2}\mu_0$$

which turns out to be a linear interpolation between the prior mean and the sample mean.

When $\sigma_0 \to \infty$, we get a so-called non-informative prior which is an improper probability distribution. In this case, the MAP estimator $\widehat{\mu}_{\mathrm{MAP}}$ converges to the MLE estimator $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^n x_i$.

3 Statistical Learning

3.1 Elements of Statistical Learning

Let's review some basics of machine learning. Let $\mathcal{X}$ denote the input space, and $\mathcal{Y}$ the label space. Typically, $\mathcal{X} \subseteq \mathbb{R}^d$, and $\mathcal{Y} = \{-1, +1\}$ for classification and $\mathcal{Y} = \mathbb{R}$ for regression. The learning algorithm is given a training data set $S = \{z_i = (x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, and a hypothesis/model class $\mathcal{H}$ with functions $h : \mathcal{X} \mapsto \mathcal{Y}$. Its goal is to find a function $h \in \mathcal{H}$ using the training data such that h can have good prediction performance on future test data.

Some connection between the training and test data is thus needed; the typical assumption is that they are all i.i.d. samples from some unknown ground-truth data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. This puts the learning problem as a statistical inference problem: given samples $S = \{z_i\}_{i=1}^n$ from the distribution $\mathcal{D}$, one would like to infer the parameter $h \in \mathcal{H}$.

Framing unsupervised learning as statistical inference is also straightforward. Many unsupervised learning problems (e.g., generative modeling) can indeed be regarded as given samples x_i's from a distribution $\mathcal{D}$, one would like to learn some parameter $\boldsymbol{\theta}$ (e.g., the parameter of the generative model) of the distribution.

3.2 Example: Linear Regression via MLE

In the regression task, we are given training data points $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^p$, and would like to learn a model $h(\boldsymbol{x})$ for predicting the label y on new test points $\boldsymbol{x}$. (Usually we aim to predict one value $\mathbb{E}(y|x)$ instead of outputting the whole distribution $p(y|\boldsymbol{x})$.)

As mentioned above, for the goal of prediction, we will assume the training/test data are from some unknown distribution $p(\boldsymbol{x}, y) = p(\boldsymbol{x})p(y|\boldsymbol{x})$. Here assumes the following distribution[1] for $p(y|\boldsymbol{x})$:

$$y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

which is equivalent to that $y|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}^\top \boldsymbol{\theta}, \sigma^2)$. Note that we do not need assumptions on the input distribution $p(\boldsymbol{x})$.

Learning via MLE. Now we use MLE for learning $\boldsymbol{\theta}$. Recall that $\epsilon \sim \mathcal{N}(0, \sigma^2)$ has density $\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$. The likelihood is then:

$$\begin{aligned} L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) = p\left(\{\boldsymbol{x}_i, y_i\}_{i=1}^n \mid \boldsymbol{\theta}\right) &= \prod_{i=1}^n p\left(\boldsymbol{x}_i, y_i \mid \boldsymbol{\theta}\right) \\ &= \prod_{i=1}^n p(\boldsymbol{x}_i) p\left(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}\right) \\ &= \prod_{i=1}^n p(\boldsymbol{x}_i) \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2}{2\sigma^2}\right). \end{aligned}$$

[1]More precisely, we assume $y = \boldsymbol{x}^\top \boldsymbol{\theta} + \theta_0 + \epsilon$. But by letting $\boldsymbol{x}' \leftarrow [\boldsymbol{x}; 1]$ and $\boldsymbol{\theta}' \leftarrow [\boldsymbol{\theta}; \theta_0]$, the assumption is equivalent to $y = (\boldsymbol{x}')^\top \boldsymbol{\theta}' + \epsilon$. We thus simply use the latter for convenience.

The MLE is:

$$\begin{aligned}\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \log L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ -\sum_{i=1}^n (\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2.\end{aligned}$$

Therefore, MLE leads to the following least squares estimate:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \sum_{i=1}^n (\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2.$$

It will be convenient to arrange the input in a matrix $\boldsymbol{X}_{n\times p}$ where each row is a feature vector, and a label vector $\boldsymbol{y}_{n\times 1}$. Then the least squares estimate can be written as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2.$$

Following the same process, one can show that when assuming $y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) + \epsilon$ for some function $h_{\boldsymbol{\theta}} \in \mathcal{H}$ with parameter $\boldsymbol{\theta}$, MLE leads to the least squares estimate:

$$\widehat{\boldsymbol{\theta}} = \underset{h_{\boldsymbol{\theta}}\in\mathcal{H}}{\operatorname{argmin}}\ \sum_{i=1}^n (h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2.$$

3.3 Example: Regularized Regression via MAP

Consider the above setting with the assumption $y = \boldsymbol{\theta}^\top \boldsymbol{x} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$. Additionally, we assume a prior on $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} \sim \mathcal{N}(0, \sigma_{\boldsymbol{\theta}}^2 I).$$

Learning via MAP. Recall that the likelihood is:

$$L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) = \prod_{i=1}^n p(\boldsymbol{x}_i) \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2}{2\sigma^2}\right).$$

The MAP is then:

$$\begin{aligned}\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) \cdot p(\boldsymbol{\theta}) &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \log L(\boldsymbol{\theta}|\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n) + \log p(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ -\frac{1}{2\sigma^2}\sum_{i=1}^n (\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i)^2 - \frac{1}{2\sigma_{\boldsymbol{\theta}}^2}\|\boldsymbol{\theta}\|^2 \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \frac{\sigma^2}{\sigma_{\boldsymbol{\theta}}^2}\|\boldsymbol{\theta}\|^2.\end{aligned}$$

This is the ridge regression objective with $\lambda = \frac{\sigma^2}{\sigma_{\boldsymbol{\theta}}^2}$.

3.4 Example: Naïve Bayes for Multinominal Data via MLE

Recall that Naïve Bayes, we first frame the prediction problem into a problem of conditional probability $p(y|\boldsymbol{x})$, then apply Bayes' rule and the conditional independence assumption:

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})} = \frac{p(y)\prod_{j=1}^{p} p(\boldsymbol{x}_j|y)}{p(\boldsymbol{x})}$$

The learning then reduces to estimating each of the distributions $p(y)$ and $p(\boldsymbol{x}_j|y)$, which depends on their statistical models. Here we consider one term $p(\boldsymbol{x}_j|y=1)$. For notation convenience, let $x := \boldsymbol{x}_j$ and $q(x) := p(\boldsymbol{x}_j|y=1)$. Assume it is a multinomial distribution, that is, $x \in \{1, 2, \cdots, k\}$ and

$$x \sim \text{Multi}(\boldsymbol{q})$$

where $\boldsymbol{q} \in \mathbb{R}^k$ with $\boldsymbol{q}_i > 0$ and $\sum_{i=1}^{k} \boldsymbol{q}_i = 1$. Our goal is then to estimate $\boldsymbol{q}$.

Learning via MLE. Suppose among the training data S, there are n_{total} points with $y = 1$. Among these data points with $y = 1$, $\boldsymbol{x}_j = i$ for n_i times. Note that $\sum_{i=1}^{k} n_i = n_{total}$. For simplicity, assume $n_i > 0$ for all i. The likelihood is then:

$$L(\boldsymbol{q}|n_1, \cdots, n_k) = n! \prod_{i=1}^{k} \frac{\boldsymbol{q}_i^{n_i}}{n_i!}. \tag{1}$$

MLE leads to:

$$\underset{\boldsymbol{q}}{\text{argmax}}\ L(\boldsymbol{q}|n_1, \cdots, n_k) = \prod_{i=1}^{k} \frac{\boldsymbol{q}_i^{n_i}}{n_i!} \tag{2}$$

$$\text{s.t.} \sum_{i=1}^{k} \boldsymbol{q}_i = 1. \tag{3}$$

The Lagrangian Multiplier is $L(\boldsymbol{q}|n_1, \cdots, n_k) - \lambda(\sum_{i=1}^{k} \boldsymbol{q}_i - 1)$. It implies that the optimal should satisfy:

$$\frac{\partial}{\partial \boldsymbol{q}_i} L(\boldsymbol{q}|n_1, \cdots, n_k) = \lambda \frac{\partial}{\partial \boldsymbol{q}_i} \sum_{i=1}^{k} \boldsymbol{q}_i, \tag{4}$$

$$\frac{n_i}{\boldsymbol{q}_i} L(\boldsymbol{q}|n_1, \cdots, n_k) = \lambda, \tag{5}$$

which leads to

$$\boldsymbol{q}_i = \frac{n_i}{\lambda} L(\boldsymbol{q}|n_1, \cdots, n_k). \tag{6}$$

Plugging into $\sum_{i=1}^{k} \boldsymbol{q}_i = 1$, we have

$$1 = \sum_{i=1}^{k} \frac{n_i}{\lambda} L(\boldsymbol{q}|n_1, \cdots, n_k) = \frac{n_{total}}{\lambda} L(\boldsymbol{q}|n_1, \cdots, n_k).$$

This means

$$\lambda = n_{total} L(\boldsymbol{q} | n_1, \cdots, n_k)$$

and thus

$$\widehat{\boldsymbol{q}}_i = \frac{n_i}{\lambda} L(\boldsymbol{q} | n_1, \cdots, n_k) = \frac{n_i}{n_{total}}.$$

3.5 Example: k-means via MLE on Mixtures of Gaussians

Clustering is a typical unsupervised learning task, and k-means is probably the most popular clustering method in practice. Here we derive k-means via MLE.

Given data points $\boldsymbol{x}_i \in \mathbb{R}^d$, the goal of clustering is to find k centers $c_y \in\in \mathbb{R}^d (y = 1, \cdots, k)$ and assign each point $\boldsymbol{x}_i$ to the $y(\boldsymbol{x}_i)$-th center $\boldsymbol{c}_{y(\boldsymbol{x}_i)}$. Let $y_i = y(\boldsymbol{x}_i)$. This will give k clusters $C_y = \{\boldsymbol{x}_i : y(\boldsymbol{x}_i) = y\} (y = 1, \cdots, k)$.

We assume the data are from a distribution of Mixtures of Gaussians, i.e., an equal mixture of k Gaussians centered at c_y's: $\frac{1}{k} \sum_{y=1}^{k} \mathcal{N}(\boldsymbol{c}_y, \sigma^2)$. This is equivalent to:

$$\begin{aligned} y_i &\sim \text{Uniform}(1, 2, \cdots, k), \\ \boldsymbol{x}_i | y_i &\sim \mathcal{N}(\boldsymbol{c}_{y_i}, \sigma^2). \end{aligned}$$

Given the data $\boldsymbol{x}_i$'s, our goal is to infer $\boldsymbol{c}_y$'s and y_i's.

Learning via MLE. The likelihood is:

$$\begin{aligned} L(\{y_i\}, \{\boldsymbol{c}_y\} | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) &= p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n | \{y_i\}, \{\boldsymbol{c}_y\}) \\ &= \prod_{i=1}^{n} p(\boldsymbol{x}_i | \{y_i\}, \{\boldsymbol{c}_y\}) \\ &= \prod_{i=1}^{n} \frac{1}{(2\pi\sigma^2)^{k/2}} \exp\left(- \frac{\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|^2}{2\sigma^2} \right) \end{aligned}$$

Then MLE gives:

$$\begin{aligned} \underset{\{y_i\}, \{\boldsymbol{c}_y\}}{\text{argmax}} \; L(\{y_i\}, \{\boldsymbol{c}_y\} | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) &= \underset{\{y_i\}, \{\boldsymbol{c}_y\}}{\text{argmax}} \; \log L(\{y_i\}, \{\boldsymbol{c}_y\} | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \\ &= \underset{\{y_i\}, \{\boldsymbol{c}_y\}}{\text{argmin}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|^2. \end{aligned}$$

This is the k-means optimization objective:

$$\min_{\{y_i\}, \{\boldsymbol{c}_y\}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|^2.$$

To solve this, we use the alternative minimization method: alternate between (1) updating the assignments y_i's while fixing the centers $\boldsymbol{c}_y$'s and (2) updating the centers $\boldsymbol{c}_y$'s while fixing the assignments y_i's. For (1), it is easy to see that the optimal assignments for fixed centers are just assigning each point individually to its nearest center, i.e., $y_i = \operatorname{argmin}_y \|\boldsymbol{x}_i - \boldsymbol{c}_y\|^2, \forall i = 1, \cdots, n$. For (2), it is also easy to see that the optimal centers are just the average of the points in each cluster, i.e., $\boldsymbol{c}_y = \text{average}\{\boldsymbol{x}_i : y_i = y\}, \forall y = 1, \cdots, k$. This leads to the k-means algorithm:

1. Initialize the centers $\boldsymbol{c}_y$. E.g., by randomly picking data points as the centers.
2. Assign each point to its nearest center:
$$y_i = \operatorname*{argmin}_y \|\boldsymbol{x}_i - \boldsymbol{c}_y\|^2, \ \forall i = 1, \cdots, n.$$
3. Update the centers to the average of the points in their clusters:
$$\boldsymbol{c}_y = \text{average}\{\boldsymbol{x}_i : y_i = y\}, \ \forall y = 1, \cdots, k.$$
4. If not converged, goto Step 2.

3.6 Statistical Learning Framework

We can summarize the typical approach for statistical learning as follows:

1. Formalize the application problem as a probabilistic problem. For example, prediction problems are formalized as a conditional probability problem $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$, while generative models are formalized as a distribution estimation problem $p_{\boldsymbol{\theta}}(\boldsymbol{x})$.
2. Potentially manipulate probabilistic problem, and determine the statistical inference problem. This can use prior knowledge/assumptions and tools from probability theory. For example, Naïve Bayes uses Bayes' rule and the conditional independence assumption, and reduces the estimation of $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ to estimations of individual ones $p(y)$ and $p(\boldsymbol{x}_j|y)$'s.
3. Choose proper statistical model class for the inference problem. For example, the linear signal + Gaussian noise models for regression, or the multinomial distribution for the terms in Naïve Bayes.
4. Use statistical principles like MLE/MAP to derive the estimator. This usually leads to some optimization objective for computing the parameters. For example, for prediction problems, MLE leads to the negative log-likelihood loss function $\ell(\boldsymbol{x}_i, y_i, \boldsymbol{\theta}) := -\log p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i)$, and the objective is $\min_{\boldsymbol{\theta}} \sum_i \ell(\boldsymbol{x}_i, y_i, \boldsymbol{\theta})$.
5. Train the model by optimization on the objective.
6. Test the obtained model on future test data.

Note 1. The training can involve convex or non-convex optimization. Sometimes we can get closed-form solution, e.g., for linear regression, but this is rare. We typically will need to perform some iterative optimization algorithm. The standard optimization algorithm for modern machine learning is stochastic gradient descent, or its variants. We will learn more about the optimization theory and methods in later lectures.

Note 2. The learning typically obtains a model by optimizing the training objective, which is the average over the data points $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of some loss function ℓ derived via MLE/MAP or other means:

$$\widehat{R}(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{x}_i, y_i, \boldsymbol{\theta}).$$

This is called the empirical risk (or empirical loss, training loss, etc).

However, our goal is good performance on the future test data, which is typically measured by the expected loss over the underlying distribution:

$$R(\boldsymbol{\theta}) := \mathbb{E}\ \ell(\boldsymbol{x}, y, \boldsymbol{\theta}).$$

This is called the population risk, or simply the risk (or population loss, test loss, generalization loss, etc).

We would like the learned model $\widehat{\boldsymbol{\theta}}$ to have a small risk, i.e., it generalizes to the whole distribution. The risk decomposition

$$R(\widehat{\boldsymbol{\theta}}) = \widehat{R}(\widehat{\boldsymbol{\theta}}) + [R(\widehat{\boldsymbol{\theta}}) - \widehat{R}(\widehat{\boldsymbol{\theta}})]$$

means we should get a small empirical loss, and also make sure that the population risk is close to the empirical loss. The former is handled by optimization. The latter is related to concentration of the sample mean to the expectation, which will be reviewed in later lectures.