Machine Learning

Notes taken by Runqiu Ye
Carnegie Mellon University

Spring 2025

Contents

1 Probability and statistical inference

1.1 Probability

Definition (Types of convergence)**.** Let $\{X_n\}_{n=1}^\infty$ be a sequence of random variables and X be another random variable. Let F_n be the CDF of X_n for each $n \in \mathbb{N}$ and F be the CDF of X.

1. X_n converges to X **in probability** and write $X_n \xrightarrow{\text{P}} X$ if for arbitrary $\varepsilon > 0$,
$$\mathbb{P}\left[|X_n - X| > \varepsilon\right] \to 0$$
as $n \to \infty$.
2. X_n converges to X **in distribution** and write $X_n \rightsquigarrow X$ if
$$\lim_{n\to\infty} F_n(t) = F(t)$$
for all t where F is continuous.
3. X_n converges to X in L^p if
$$\mathbb{E}\left[|X_n - X|^p\right] \to 0$$
as $n \to \infty$. In particular, say X_n converges to X in **quadratic mean** and write $X_n \xrightarrow{\text{qm}} X$ if X_n converges to X in L^2.
4. X_n converges to X **almost surely** and write $X_n \xrightarrow{\text{as}} X$ if
$$\mathbb{P}\left[\lim_{n\to\infty} X_n = X\right] = 1.$$

Theorem. The following implication holds:

1. If X_n converges to X almost surely, then X_n converges to X in probability.
2. If X_n converges to X in L^p, then X_n converges to X in probability.

Proof. 1. If X_n converges to X almost surely, the set of points $O = \{\omega : \lim_{n\to\infty} X_n(\omega) \neq X(\omega)\}$ has measure zero. Now fix $\varepsilon > 0$ and consider the sequence of sets
$$A_n = \bigcup_{m=n}^{\infty} \{|X_m - X| > \varepsilon\}.$$
Note that $A_n \supset A_{n+1}$ for each $n \in \mathbb{N}$ and let $A_\infty = \bigcap_{n=1}^\infty A_n$. Now show $\mathbb{P}[A_\infty] = 0$. If $\omega \notin O$, then $\lim_{n\to\infty} X_n(\omega) = X(\omega)$ and thus $|X_n(\omega) - X(\omega)| < \varepsilon$ for some $n \in \mathbb{N}$. Therefore, $\omega \notin A_\infty$. It follows that $A_\infty \subset O$ and $\mathbb{P}[A_\infty] = 0$.

By monotone continuity, we have $\lim_{n\to\infty} \mathbb{P}[A_n] = \mathbb{P}[A_\infty]$. It follows that
$$\mathbb{P}\left[|X_n - X| > \varepsilon\right] \leq \mathbb{P}\left[A_n\right] \to 0$$
as $n \to \infty$. This completes the proof.

2. From Chebyshev's inequality, we have
$$\mathbb{P}\left[|X - X_n| > \varepsilon\right] \leq \frac{1}{\varepsilon^p}\mathbb{E}[|X - X_n|^p].$$
The claim follows directly.

□

Theorem (Central Limit Theorem)**.** Let $X_1, \dots, X_n$ be i.i.d. with mean μ and variance σ^2. Let $S_n = \frac{1}{n}\sum_{i=1}^n X_i$. Then

$$Z_n = \frac{S_n - \mu}{\sqrt{\operatorname{Var} S_n}} = \frac{\sqrt{n}\,(S_n - \mu)}{\sigma} \rightsquigarrow Z,$$

where $Z \sim \mathcal{N}(0,1)$. In other words,

$$\lim_{n\to\infty} \mathbb{P}[Z_n < z] = \Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, dx.$$

Also write $Z_n \approx \mathcal{N}(0,1)$.

1.2 Statistical inference

Definition. Let $X_1, \dots, X_n$ be n i.i.d. data points observed from some distribution F with respect to parameter θ. A point estimator $\widehat{\theta}_n$ of the parameter θ is some function of $X_1, \dots, X_n$:

$$\widehat{\theta}_n = g(X_1, \dots, X_n).$$

The bias of an estimator is defined as

$$\operatorname{bias}(\widehat{\theta}_n) = \mathbb{E}_\theta[\widehat{\theta}_n] - \theta.$$

The mean squared error is defined as

$$\mathrm{MSE} = \mathbb{E}_\theta(\widehat{\theta}_n - \theta)^2.$$

Definition. A point estimator $\widehat{\theta}_n$ of a parameter θ is **consistent** if $\widehat{\theta}_n \xrightarrow{\text{P}} \theta$.

Next we an important relation between bias, variance, and MSE. This is a more rigorous way to express the **bias-variance tradeoff** of point estimators.

Theorem. The MSE can be written as

$$\mathrm{MSE} = \operatorname{bias}^2(\widehat{\theta}_n) + \operatorname{Var}_\theta(\widehat{\theta}_n).$$

Proof. Let $\overline{\theta}_n = \mathbb{E}_\theta(\widehat{\theta}_n)$. Then we have

$$\begin{aligned}
\mathbb{E}_\theta(\theta - \widehat{\theta}_n)^2 &= \mathbb{E}_\theta(\theta - \overline{\theta}_n + \overline{\theta}_n - \widehat{\theta}_n)^2 \\
&= \mathbb{E}_\theta(\theta - \overline{\theta}_n)^2 - 2(\theta - \overline{\theta}_n)\mathbb{E}_\theta(\overline{\theta}_n - \widehat{\theta}_n) + \mathbb{E}_\theta(\overline{\theta}_n - \theta)^2 \\
&= (\theta - \overline{\theta}_n)^2 + \mathbb{E}_\theta(\overline{\theta}_n - \theta)^2 \\
&= \operatorname{bias}^2(\widehat{\theta}_n) + \operatorname{Var}_\theta(\widehat{\theta}_n),
\end{aligned}$$

where we have used the fact that $\mathbb{E}_\theta(\overline{\theta}_n - \widehat{\theta}_n) = \overline{\theta}_n - E_\theta(\widehat{\theta}_n) = \overline{\theta}_n - \overline{\theta}_n = 0$. □

Below is the definition of a confidence set/interval.

Definition. A $1-\alpha$ interval for a parameter θ is an interval $C_n = (a,b)$ where $a = a(X_1, \dots, X_n)$ and $b = b(X_1, \dots, X_n)$ are functions of data such that

$$\mathbb{P}_\theta[\theta \in C_n] \geq 1 - \alpha \text{ for all } \theta \in \Theta.$$

In other word, (a,b) traps θ with probablity $1-\alpha$.

Warning! In the above definition, C_n is random and θ is fixed.

1.3 PAC learning

PAC learning is short for Probably Approxinate Correct learning, and the setting of PAC learning is as follows:

– We have data $x \in \mathbb{R}^d$ and label $y \in \{-1, +1\}$.

– We collect features $x_1, \ldots, x_n$ iid from some distribution D. Note that we make no assumption on the distribution D of the features.

– We collect corresponding labels $y_1, \ldots, y_n$.

– Assume there exists some true classifier $h^\star$.

– Let $\mathscr{H}$ be the set of all hypotheses.

The goal of PAC learning is (ε, δ)-PAC, which is defined as follows.

Definition. An (ε, δ)-PAC learning algorithm refers to an algorithm that picks an hypothesis $h \in \mathscr{H}$ after observing training data $\{x_i, y_i\}_{i=1}^n$, where the hypothesis $h \in \mathscr{H}$ satisfies

$$\mathrm{err}_D(\widehat{h}) = \mathbb{P}_{x\sim D}\left[\widehat{h}(x) \neq h^\star(x)\right] < \varepsilon.$$

with probability at least $1-\delta$, where the probability is with respect to the randomness of the training set.

Putting it more concretely, suppose g is a point estimator (the algorithm), then we want

$$\mathbb{P}_{x_i\sim D}\left[\mathrm{err}_D(g(x_1, \ldots, x_n)) < \varepsilon\right] \geq 1-\delta.$$

We also say the algorithm is a PAC learner if it uses n samples and the running time is at most $\mathrm{poly}\left(d, \frac{1}{\varepsilon}, \log\frac{1}{\delta}, \mathrm{bits}(h^\star)\right)$, but in this section we do not focus on the runtime of the algorithm.

There are two types of PAC learning – realizable PAC learning, in which case $h^\star \in \mathscr{H}$, and agnostic PAC learning, in which case $h^\star \notin \mathscr{H}$. We first discuss realizable PAC learning, and we will find out agnostic PAC learning is a more general setting but than PAC learning but a natural extension.

Realizable PAC learning

For realizable PAC learning, we present a simple algorithm:

Algorithm (Consistent Learner)**.** Pick any $\widehat{h} \in \mathscr{H}$ such that $h(x_i) = y_i$ for all $1 \leq i \leq n$.

Now we analyze this algorithm in terms of the sample size needed to produce a desired hypothesis.

Theorem. Over the dataset of n iid samples, the consistent learning produces $\widehat{h}$ such that $\mathrm{err}_D(\widehat{h}) > \varepsilon$ with probability at most $|\mathscr{H}|\, e^{-n\varepsilon}$.

Proof. Suppose $h \in \mathscr{H}$ is such that $\mathrm{err}_D(h) = \mathbb{P}_{x\sim D}[h(x) \neq h^\star(x)] > \varepsilon$. For such an h and some data x_i, we have

$$\mathbb{P}_{x_i\sim D}[h(x_i) = y_i = h^\star(x_i)] < 1-\varepsilon.$$

Since $\{x_i, y_i\}_{i=1}^n$ are iid, we have

$$\mathbb{P}_{x_{1:n}\sim D}[h(x_i) = h^\star(x_i) \text{ for all } 1 \leq i \leq n] < (1-\varepsilon)^n.$$

Note that our consistent learner do not make any mistake on the training set $\{x_i, y_i\}_{i=1}^n$

$$\mathbb{P}_{x_{1:n}\sim D}[h = \widehat{h}] < (1-\varepsilon)^n.$$

It follows that

$$\begin{aligned}\mathbb{P}_{x_{1:n}\sim D}[\mathrm{err}_D(\widehat{h}) > \varepsilon] &\leq \sum_{h\in\mathscr{H}:\ \mathrm{err}_D(h)>\varepsilon} \mathbb{P}[\widehat{h} = h] \\ &\leq |\mathscr{H}|\,(1-\varepsilon)^n \\ &\leq |\mathscr{H}|\, e^{-n\varepsilon},\end{aligned}$$

where in the last step we used the inequality $(1-u)^n \leq e^{-nu}$. This completes the proof. □

Corollary. If we want $\mathbb{P}_{x_{1:n}\sim D}[\mathrm{err}_D(\widehat{h}) > \varepsilon] \leq \delta$, we must have

$$n \geq \frac{\ln\left(|\mathscr{H}|/\delta\right)}{\varepsilon}.$$

Technically speaking the implication should be the other direction, but this bound for sample size n **gurantees** (ε, δ)-PAC.

To illustrate how we should think of $|\mathscr{H}|$, we present an example.

Example. *** TO-DO ***

Now we move on to the setting of agnostic leraning.

Agnostic leraning

In agnostic learning, again we collect features $x_1, \ldots, x_n \sim D$ iid, and labels $y_1, \ldots, y_n$. This forms a data set $S_n = \{x_i, y_i\}_{i=1}^n$. The goal is to use this dataset S_n to produce an hypothesis $\widehat{h}$ such that

$$\mathrm{reg}_{D,\mathscr{H}}(\widehat{h}) = \mathrm{err}_D(\widehat{h}) - \min_{h\in\mathscr{H}} \mathrm{err}_D(h) < \varepsilon$$

with probability at least $1-\delta$, where the probability is with respect to the randomness of the dataset S_n.

For this setting, we

2 Mixture models and EM

2.1 Kullback-Leibler (KL) Divergence

In this section, we adopt the convention that $0 \log 0 = 0$.

Definition (KL divergence)**.** The Kullback-Leibler (KL) divergence of two distributions $P(X)$ and $Q(X)$ over the outcome space X is defined as follows:

$$\mathrm{KL}(P \| Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}.$$

To understand the significance of KL-divergence better, we first discuss some related concepts in information theory, starting with the definition of **entropy**.

Definition. The entropy of a distribution $P(X)$ is defined as

$$H(P) = -\sum_{x \in X} P(x) \log P(x)$$

Intuitively, entropy measures how dispersed a probability distribution is. For example, a uniform distribution is considered to have very high entropy (i.e. a lot of uncertainty), whereas a distribution that assigns all its mass on a single point is considered to have zero entropy (i.e. no uncertainty). Notably, it can be shown that among continuous distributions over $\mathbb{R}$, the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ has the highest entropy (highest uncertainty) among all possible distributions that have the given mean μ and variance σ^2.

To further solidify our intuition, we present motivation from communication theory. Suppose we want to communicate from a source to a destination, and our messages are always (a sequence of) discrete symbols over space X (for example, X could be letters {a, b, . . . , z}). We want to construct an encoding scheme for our symbols in the form of sequences of binary bits that are transmitted over the channel. Further, suppose that in the long run the frequency of occurrence of symbols follow a probability distribution $P(X)$. This means, in the long run, the fraction of times the symbol x gets transmitted is $P(x)$.

A common desire is to construct an encoding scheme such that the average number of bits per symbol transmitted remains as small as possible. Intuitively, this means we want very frequent symbols to be assigned to a bit pattern having a small number of bits. Likewise, because we are interested in reducing the average number of bits per symbol in the long term, it is tolerable for infrequent words to be assigned to bit patterns having a large number of bits, since their low frequency has little effect on the long term average. The encoding scheme can be as complex as we desire, for example, a single bit could possibly represent a long sequence of multiple symbols (if that specific pattern of symbols is very common). The entropy of a probability distribution $P(X)$ is its optimal bit rate, i.e., the lowest average bits per message that can possibly be achieved if the symbols $x \in X$ occur according to $P(X)$. It does not specifically tell us how to construct that optimal encoding scheme. It only tells us that no encoding can possibly give us a lower long term bits per message than $H(P)$.

To see a concrete example, suppose our messages have a vocabulary of $K = 32$ symbols, and each symbol has an equal probability of transmission in the long term (i.e, uniform probability distribution). An encoding scheme that would work well for this scenario would be to have $\log_2 K$ bits per symbol, and assign each symbol some unique combination of the $\log_2 K$ bits. In fact, it turns out that this is the most efficient encoding one can come up with for the uniform distribution scenario.

It may have occurred to you by now that the long term average number of bits per message depends only on the frequency of occurrence of symbols. The encoding scheme of scenario A can in theory be reused in scenario B with a different set of symbols (assume equal vocabulary size for simplicity), with the same long term efficiency, as long as the symbols of scenario B follow the same probability distribution as the symbols of scenario A. It might also have occured to you, that reusing the encoding scheme designed to be optimal for scenario A, for messages in scenario B having a different probability of symbols, will always be suboptimal for scenario B. To be clear, we do not need know what the specific optimal schemes are in either scenarios. As long as we know the distributions of their symbols, we can say that the optimal scheme designed for scenario A will be suboptimal for scenario B if the distributions are different.

Concretely, if we reuse the optimal scheme designed for a scenario having symbol distribution $Q(X)$, into a scenario that has symbol distribution $P(X)$, the long term average number of bits per symbol achieved is called the cross entropy, denoted by $H(P, Q)$:

Definition. The cross-entropy of two distributions $P(X)$ and $Q(X)$ is defined as

$$H(P, Q) = -\sum_{x \in X} P(X) \log Q(x)$$

To recap, the entropy $H(P)$ is the best possible long term average bits per message (optimal) that can be achived under a symbol distribution $P(X)$ by using an encoding scheme (possibly unknown) specifically designed for $P(X)$. The cross entropy $H(P, Q)$ is the long term average bits per message (suboptimal) that results under a symbol distribution $P(X)$, by reusing an encoding scheme (possibly unknown) designed to be optimal for a scenario with symbol distribution $Q(X)$.

Now, KL divergence is the penalty we pay, as measured in average number of bits, for using the optimal scheme for $Q(X)$, under the scenario where symbols are actually distributed as $P(X)$. It is straightforward to see this:

$$\begin{aligned}\mathrm{KL}(P\|Q) &= \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \\ &= \sum_{x \in X} P(x) \log P(x) - \sum_{x \in X} P(x) \log Q(x) \\ &= H(P, Q) - H(P). \quad \text{(difference in average number of bits)}\end{aligned}$$

If the cross entropy between P and Q is zero (and hence $KL(P\|Q) = 0$) then it necessarily means $P = Q$. In Machine Learning, it is a common task to find a distribution Q that is "close" to another distribution P. To achieve this, we use $\mathrm{KL}(P\|Q)$ to be the loss function to be optimized. As we will see in this below, Maximum Likelihood Estimation, which is a commonly used optimization objective, turns out to be equivalent minimizing KL divergence between the training data (i.e. the empirical distribution over the data) and the model.

Now we present some useful properties of KL divergence.

Theorem (Non-negativity)**.** For any distribution P and Q, we have

$$\mathrm{KL}(P\|Q) \geq 0,$$

and $\mathrm{KL}(P\|Q) = 0$ if and only if $P = Q$ (almost everywhere).

Proof. By definition,

$$\mathrm{KL}(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} = -\sum_{x \in X} P(x) \log \frac{Q(x)}{P(x)}.$$

Since $-\log x$ is strictly convex, by Jensen's inequality, we have

$$\mathrm{KL}(P\|Q) = -\sum_{x \in X} P(x) \log \frac{Q(x)}{P(x)} \geq -\log \sum_{x \in X} P(x) \frac{Q(x)}{P(x)} = 0.$$

When the equality holds,

$$\log \frac{Q(x)}{P(x)} = 0$$

almost everywhere. That is, $Q = P$ almost everywhere. This completes the proof. □

Definition (KL divergence of conditional distribution)**.** The KL divergence between 2 conditional distributions $P(X \mid Y)$, $Q(X \mid Y)$ is defined as follows:

$$\mathrm{KL}(P(X \mid Y) \| Q(X \mid Y)) = \sum_y P(y) \left(\sum_x P(x \mid y) \log \frac{P(x \mid y)}{Q(x \mid y)} \right).$$

This can be thought of as the expected KL divergence between the corresponding conditional distributions on x. That is, between $P(X \mid Y = y)$ and $Q(X \mid Y = y)$, where the expectation is taken over the random y.

Theorem (Chain rule for KL divergence)**.** The following equality holds:

$$\mathrm{KL}(P(X,Y) \| Q(X,Y)) = \mathrm{KL}(P(X) \| Q(X)) + \mathrm{KL}(P(Y \mid X) \| Q(Y \mid X)).$$

Proof.

$$\begin{aligned}
\mathrm{LHS} &= \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{Q(x,y)} \\
&= \sum_x \sum_y P(y \mid x) P(x) \left[\log \frac{P(y \mid x)}{Q(y \mid x)} + \log \frac{P(x)}{Q(x)} \right] \\
&= \sum_x \sum_y P(y \mid x) P(x) \log \frac{P(y \mid x)}{Q(y \mid x)} + \sum_x P(x) \log \frac{P(x)}{Q(x)} \sum_y P(y \mid x) \\
&= \sum_x \sum_y P(y \mid x) P(x) \log \frac{P(y \mid x)}{Q(y \mid x)} + \sum_x P(x) \log \frac{P(x)}{Q(x)} \\
&= \mathrm{KL}(P(X) \| Q(X)) + \mathrm{KL}(P(Y \mid X) \| Q(Y \mid X)) \\
&= \mathrm{RHS}.
\end{aligned}$$

□

2.2 The EM Algorithm in General

3 Approximate inference

A central task in the application of probabilistic models is the evaluation of the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ of the latent variables $\mathbf{Z}$ given the observed (visible) data variables $\mathbf{X}$, and the evaluation of expectations computed with respect to this distribution. For many models of practical interest, it will be infeasible to evaluate the posterior distribution or indeed to compute expectations with respect to this distribution.

In this chapter, we introduce a range of deterministic approximation schemes, some of which scale well to large applications. These are based on analytical approximations to the posterior distribution, for example by assuming that it factorizes in a particular way or that it has a specific parametric form such as a Gaussian. As such, they can never generate exact results, and so their strengths and weaknesses are complementary to those of sampling methods.