Machine Learning Basics

A self-study metarials for PRML [1]

Jisung Lim[1]

[1] B.S. Candidate of Industrial Engineering
Yonsei University, South Korea.

28th January, 2017

Summary

Optical Character Recognition

OCR Problem and Some Approaches

Figure 1: The MNIST database.

- **Input**: 28×28 pixel image, represented by a vector $\boldsymbol{x}$ comprising 784 real numbers.
- **Goal**: To build a machine that will take such a vector $\boldsymbol{x}$ as input and that will produce the identity of the digit $0, \ldots, 9$ as the output.

Consider the example of recognizing handwritten digits, illustrated in Figure 1. This is the nontrivial problem due to the wide variability of handwriting. And we can tackle this problem using following approches:

1. Handcrafted rules or heuristics
2. Machine learning methods

In practice, the former leads to a proliferation of rules and invariably gives poor results. For better results, the latter can be considered an alternative.

Machine Learning Approach

Machine Learning Approach can be divided into three major stages: **training**, **testing**, and **predicting**. The overall process of a machine learning approach to classification problems such as OCR can be depicted in Figure 2.

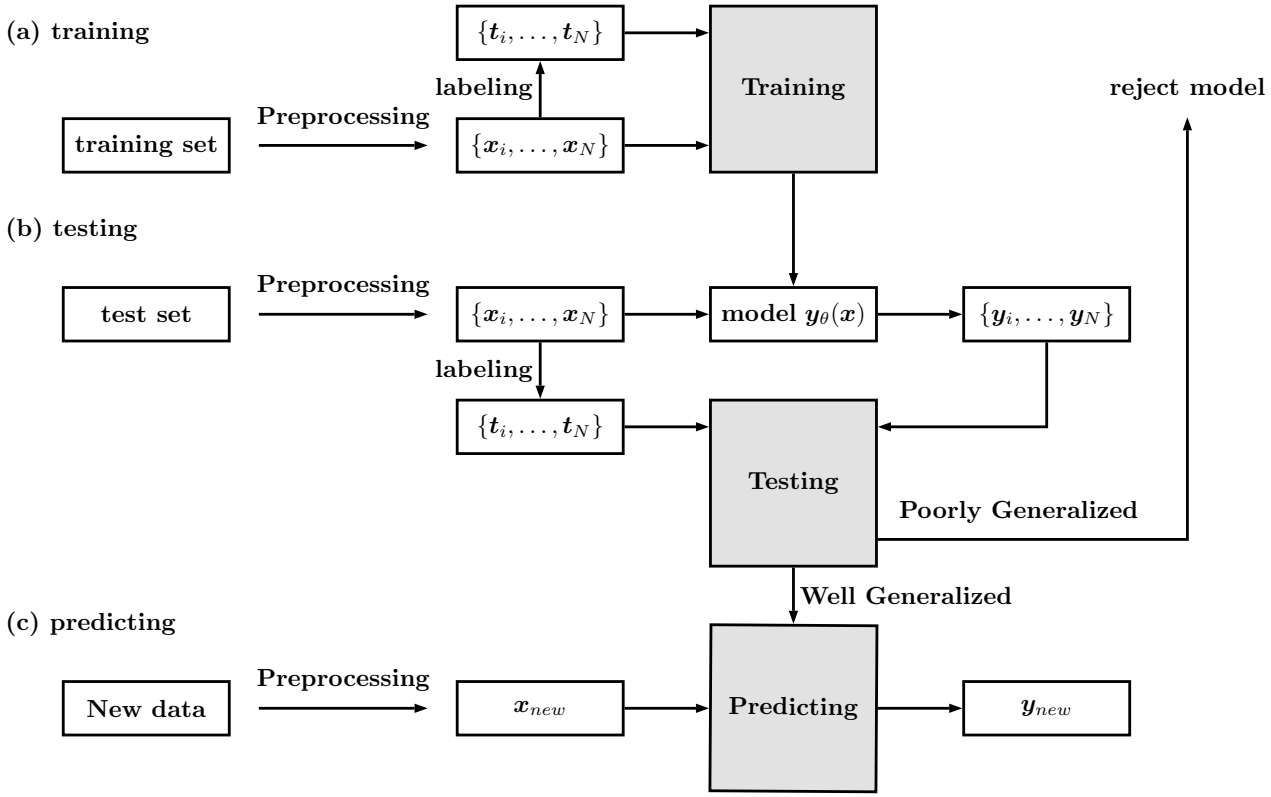

Figure 2: The overall process of machine learning approach, especially of supervised learning.

ML Approach — Input

First of all, we need to define inputs more precisely. In OCR, inputs are handwriting images of a digit. In practical applications, the original input images may have different sizes, ratios, angles, or even colors. Hence, the images are typically scaled, rotated, translated, and grey-scaled so that each digit is contained within a box of a fixed-size, says 28×28, and is toned with greyscale color. And then, the preprocessed input images can be represented as a vector $\boldsymbol{x} = (x_1, \ldots, x_{784}) \in \mathbb{R}^n$ where the greyscale color of ith pixel is a x_i. Finally, you should label the true number $\boldsymbol{t}$ for each input image $\boldsymbol{x}$.

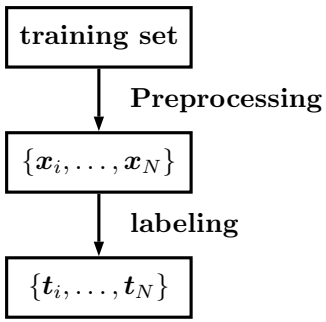

Figure 3

Preprocess: To transform the original input variables into some new space of variables so that the problem becomes easier to solve.

Example x_i: An example is a collection of features. Typically, an **example** will be represented as a vector $\boldsymbol{x}_i \in \mathbb{R}^n$ where each entry x_i is a **feature**.

Label t_i: A **label** represents the identity of the corresponding digit. Typically, the labels are hand-labelled by inspecting each image individually. Note that there is one such **label** $\boldsymbol{t}_i$ for each **example** $\boldsymbol{x}_i$.

ML Approach — Train and Test

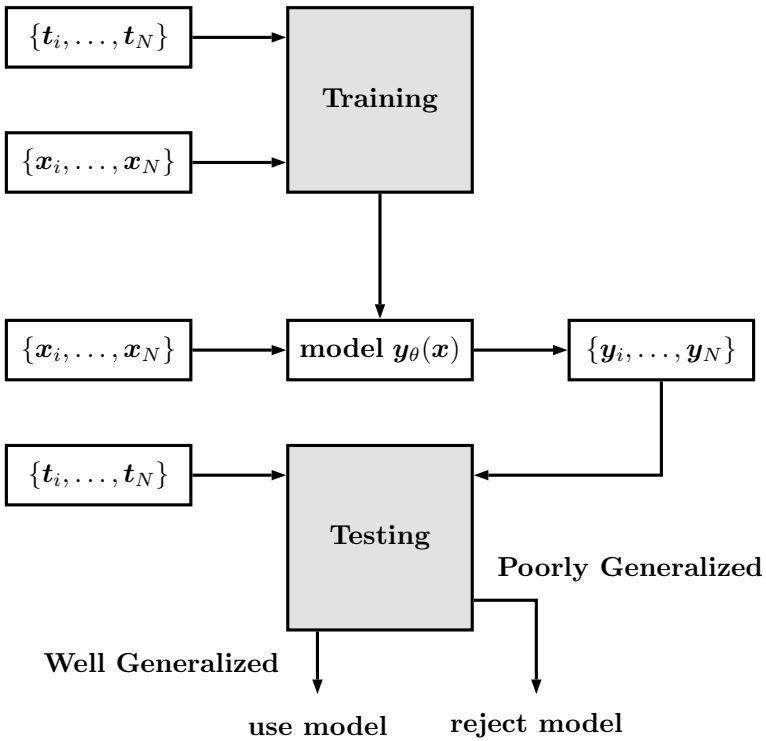

Figure 4: The description of training and test.

- **Training**

  To tune the parameters of an adaptive model, it uses a large set of N **examples** $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with its corresponding **labels** $\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N\}$. The model can be expressed as a function $\boldsymbol{y}(\boldsymbol{x})$ which takes a new digit image $\boldsymbol{x}_{new}$ as input and that generates an output vector $\boldsymbol{y}_{new}$.

- **Testing**

  Once the model is trained it can then determine the identity of new digit images, which are said to comprise a test set. The ability to categorize correctly new examples that differ from those used for training is known as **generalization**. In practical applications, the variability of the input vectors will be such that the training data can comprise only a tiny fraction of all possible input vectors, and so generalization is a central goal in pattern recognition.

ML Approach — Prediction model

Let's go back to handwriting recognition. First, the key to the handwriting recognition problem is to classify entirely new, unseen data into the appropriate class. In particular, machine learning methods learn and test through preprocessed (labeled) data as we have seen so far to generate a predictive model. Since, through the testing stage, the prediction model has been verified to be sufficiently generalized, it can respond appropriately to new data and can be described as follows.

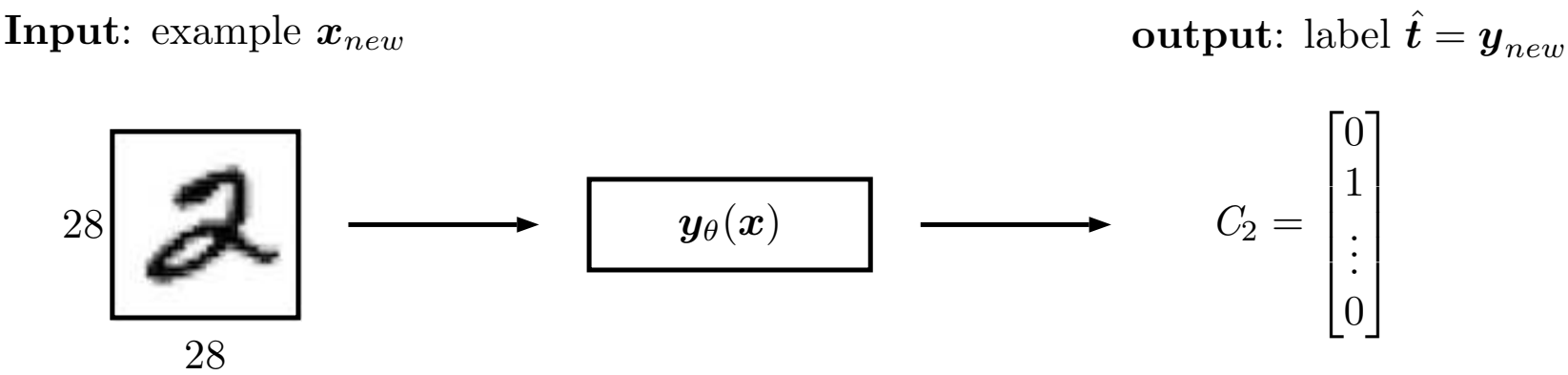

Figure 5: The description of prediction in OCR.

ML Approach — SL, UL, and RL

SL **Supervised Learning**:
Supervised learning is the machine learning task of inferring a function from labeled training data [3]. The **data** consists of a set of examples, each of which is *a pair of an input value and corresponding desired output value*.
Ex. Regression, Classification.

UL **Unsupervised Learning**:
Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution – this distinguishes unsupervised learning from supervised learning and reinforcement learning.
Ex. Clustering, Density estimation.

RL **Reinforcement Learning**:
A goal of reinforcement learning is **to find an action** which gives **greatest reward** by **interacting with a given environment**. It discovers the optimal by a process of **exploration and exploitation**. (A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent tries to find a suitable action a_t which moves the current environment s_t to next state s_{t+1} and determines a corresponding reward r_{t+1}. It discovers the optimal action by balancing the trade-off between **exploration**, in whcih the system **tries out new kinds of actions** to see how effective they are, and **exploitation**, in which the system **makes use of actions** that are known to yield a high reward.)

Precise Definition of ML

Definition of machine learning

A computer program is said to ***learn*** from **experience** ***E*** with respect to some class of **tasks** ***T*** and **performance measure** ***P***, if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E***.

Polynomial Curve Fitting

Given Situation

- N labeled observations.
- N observations $\mathbf{X} \equiv (x_1, \ldots, x_N)^T$, together with corresponding labels $\mathbf{t} \equiv (t_1, \ldots, t_N)^T$
- (x_i, t_i) possess underlying regularity, which we wish to learn.
- Individual t_i observations are corrupted by random noise, which might arise from intrinsical stochasticity due to there being source of variability.

Our Goal and Intrinsic Difficulty

- **Goal**: To exploit this given training set in order to make prediction of the value $\hat{t}$ of the target variable for some new value $\hat{x}$ of the input variable.
- **Intrinsic Difficulty**: This is a intrinsically difficult problem not only as we have to generalize from a **finite data** set but also, for a given $\hat{x}$ there is **uncertainty** as to the appropriate value for $\hat{t}$.

Polynomial Function as a Estimate $\hat{t} = y(x, \boldsymbol{w})$

Consider a simple approach based on curve fitting. We can use the polynomial function to fit the data. The value $\hat{t} = y(x, \boldsymbol{w})$ of the function is an estimate of the target value t.

$$\hat{t} = y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j \tag{1}$$

- Since M is the order of the polynomial, determining M has meaning as a **model selection**. Model selection generally makes a big difference in fitting performance and should be considered separately from the modeling data.
- The values of the coefficients w_j will be determined by **fitting** the polynomial to the training data. This can be done by minimizing difference between the target value t and and its estimate $\hat{t} = y(x, \boldsymbol{w})$.

Error Minimization

We first define an error function $E(\boldsymbol{w})$ that serve as a criteria for not only how to select the best model based on training data but also how well the model fits into the test data. We can use sum of squared errors as a error function.

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \boldsymbol{w}) - \boldsymbol{t}_n\}^2 \quad \text{where} \quad y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j \tag{2}$$

You can also use root mean square to validate the model with the test set.

$$E_{RMS}(\boldsymbol{w}^*) = \sqrt{2E(\boldsymbol{w}^*)/N} \tag{3}$$

RMS has two advantages as followings:

1. $\mathbf{1}/\boldsymbol{N}$ Normalize the size of the data set, so that it makes possible to compare different sizes of data sets on an equal footing.
2. $\sqrt{E(\cdot)}$ Remove square property of $E(\boldsymbol{w})$ so that this ensures the same measure scale between error function E_{RMS} and target value t.

Regularization

Add regularization term to error function $E(\boldsymbol{w})$ for preventing overfitting.

$$\tilde{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{4}$$

The coefficient λ governs the relative importance of the regularization term.

$$\ln\lambda = \begin{cases} -\infty & (\lambda \to 0) \quad \text{(overfitting)} \\ K & (0 < \lambda < 1) \quad \text{(well fit)} \\ 0 & (\lambda = 1) \quad \text{(underfitting)} \end{cases} \tag{5}$$

Hyperparameter

- **Hyperparameter**
  Hyperparameter is a parameter that controls the complexity of machine learning model. In polynomial curve fitting problem, the order of polynomial M and the coefficient λ which governs the relative importance of the regularizer are hyperparameters. Sometimes we can optimize or learn it, but most of time it is not appropriate to learn that hyperparameter on the training set.
- **Validation set**
  To solve above problem, we need a validation set of examples which is separately obtained from either training or test set. In polynomial curve fitting problem, by taking the available data and partitioning it into a **training set**, used to determine the coefficients $\boldsymbol{w}$, and a separate **validation set**, also called a hold-out set, used to optimize the model complexity (either M or λ).

Chapter Objectives

- **Probability Theory**
  provides a consistent framework for expressing an uncertainities in a precise and quantitative manner.
  Keywords. Two rules of probability (sum rule and product rule), Random variable, Probabilty mass and density, Frequentist vs. Bayesian, Estimation (MLE, MAP), Bias, Full Bayesian process (bayesian prediction).
- **Decision Theory**
  allows us to exploit this probabilistic representation to make optimal prediction according to appropriate criteria.
  Keywords. Inference stage and decision stage, Loss function, Reject option, Gen or Dis, Usefulness of a posterior
  Additionals. Functional, Calculus of Variations, Lagrangian and KKT conditions
- **Information Theory**
  describes, in terms of amount of information, how to encode a random event, calculate the average amount of information in a random variable, and compare two different distributions.
  Keywords. Self information and joint, conditional, or mutual information, Entrophy and differential entrophy, Maximum entrophy, KL divergence.

Where the uncertainty comes from?

- **Inherent stochasticity**
  A system could contains intrinsic random factors, so we should take those factors into account to design a model.
  e.g. Shuffle in card game, Customers or accidents come randomly
- **Incomplete observability**
  Sometimes, the system cannot be fully observed. Even if the outcome is deterministic, it might be considered a stochastic system if the outcome is not known exactly.
  e.g. Monty Hall Game, Bayesian Statistics
- **Incomplete modeling**
  To simplify, we can ignore information we already know. In this case, the information can be treated as stochastic. Sometimes, simple stochasticity is much better than complex deterministicity.
  e.g. Discretize space for robot.

Random Variable

- **Random Experiment**
  An experiment whose outcome cannot be predicted with certainty before the experiment is excuted. In classical or frequency-based probability theory, we also assume that the experiment can be repeated indefinitely under essentially the same conditions (e.g. Bernoulli trial). The repeatability assumption is important because the classical theory is concerned with the long-term behavior as the experiment is replicated. By contrast, subjective or belief-based probability theory is concerned with measures of belief about what will happen when we run the experiment. In this view, repeatability is a less crucial assumption.
- **Sample Space**
  The sample space of a random experiment is a set S that includes all possible outcomes of the experiment; the sample space plays the role of the universal set when modeling the experiment.
- **Events**
  Certain subsets of the sample space of an experiment are referred to as events.

Random Variable (conti. . .)

- **Random Variable**
  Suppose again that we have a random experiment with sample space S. A mapping x from S into another set T is called a (T-valued) random variable. If sample sapce is a countable set, then we call this random variable $discrete$. If an uncountable set, then $continuous$.
- **Probability Measure**
  A probability measure (or probability distribution) $\mathbb{P}$ for a random experiment is a real-valued function, defined on the collection of events, that satisfies the following axioms:
  1. $\mathbb{P}(A) \geq 0$ for every event A
  2. $\mathbb{P}(S) = 1$
  3. If $A_i : i \in I$ is countable, pairwise disjoint collection of events, then

$$\mathbb{P}(\cup_{i \in I} A_i) = \sum_{i \in I} \mathbb{P}(A_i) \tag{6}$$

Probability Density Function

- **Probability Density Function**
  If x is a random variable, the probability density functyion of x is the function $p(\mathrm{x} = x)$ on S that assigns probabilities p to the point $\mathrm{x} = x$ in S:
  1. $p(x) \geq 0, \forall x \in S$
  2. $\int_S p(x)\,\mathrm{dx} = 1$
  3. $\int_A p(x)\,\mathrm{dx} = \mathbb{P}(\mathrm{x} \in A), \forall A \subseteq S$
- **Probability Distribution**
  A probability distribution $p(\mathrm{x} = x)$ of random variable x is the function that assigns probabilities to the subset of S, namely:

$$A \mapsto \mathbb{P}(\mathrm{x} \in A) \text{ for } A \subseteq S \tag{7}$$

  If we use parametrized distribution, then we can use the notation such as:

$$\mathrm{x} \sim U(x; \alpha, \beta) \tag{8}$$

Basic Rules

For convenience, we will use more simple notation $\mathbb{P}(A)$ instead of $\mathbb{P}(\mathrm{x} \in A)$.

- **Sum Rule**

$$\mathbb{P}(A+B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B) \tag{9}$$

- **Product Rule**

$$\mathbb{P}(A \cap B) = \mathbb{P}(A|B)\mathbb{P}(B) = \mathbb{P}(B|A)\mathbb{P}(A) \tag{10}$$

- **Chain Rule**
  By product rule with recursive manner, we get

$$\mathbb{P}(X_{0:N-1}) = \mathbb{P}(X_0)\mathbb{P}(X_1|X_0)\cdots\mathbb{P}(X_{N-1}|X_{0:N-2}) \tag{11}$$

- **Marginal Probability**

$$\mathbb{P}(A) = \int_A p(x)\,\mathrm{dx} = \int_A \int_Y p(x,y)\,\mathrm{dydx} \tag{12}$$

- **Conditional Probability**

$$\mathbb{P}(A|x) = \int_A p(y|x)\,\mathrm{dy} = \int_A \frac{p(x,y)}{p(x)}\,\mathrm{dy} = \int_A \frac{p(x,y)}{\int_Y p(x,y)\,\mathrm{dy}}\,\mathrm{dy} \tag{13}$$

Bayes Rule

- **Bayes Rule**
  By conditional probability, we get bayes rule

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A,B)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A,B)}{\int_Y \mathbb{P}(A,B)\,\mathrm{d}y} \tag{14}$$

- **In the perspective of machine learning**
  We may substitute x and y for H, which denotes a model hypothesis, and D, which denotes a data set.

$$\mathbb{P}(H|D) = \frac{\mathbb{P}(H,D)}{\mathbb{P}(D)} = \frac{\mathbb{P}(H,D)}{\sum_H \mathbb{P}(H,D)} = \frac{\mathbb{P}(H)\mathbb{P}(D|H)}{\sum_H \mathbb{P}(H)\mathbb{P}(D|H)} \tag{15}$$

  We call $\mathbb{P}(H)$ prior, $\mathbb{P}(H|D)$ posterior, $\mathbb{P}(D|H)$ likelihood, and $\mathbb{P}(D)$ evidence.

Independence

- **Independence**
  We says random variable x and y are independence if

$$p(x, y) = p(x)p(y), \forall x \in \mathrm{x}, y \in \mathrm{y} \tag{16}$$

  This also means that $p(x) = p(x|y), p(y) = p(y|x), \forall x \in \mathrm{x}, y \in \mathrm{y}$. We denotes independence between x and y into $\mathrm{x} \perp \mathrm{y}$.
- **Conditionally Independence**
  We says random variable x and y are conditionally independence given z, if there exist function g and h such that

$$p(x, y|z) = g(x, z)h(y, z), \forall x \in \mathrm{x}, y \in \mathrm{y}, z \in \mathrm{z} \tag{17}$$

  We denotes conditionally independence between x and y given z into $\mathrm{x} \perp \mathrm{y} \mid \mathrm{z}$.

Expactation and Variance

- **Expectation**
  Expectation means expected value of $f(x)$ when x is drawn from $P(\mathrm{x} = x)$. We can get expectation of $f(x)$ by

$$\mathbb{E}_{\mathrm{x}\sim P}[f(x)] = \int_{\mathrm{x}} f(x)P(\mathrm{x} = x)\,\mathrm{dx} \tag{18}$$

  Expectations are linear functional,

$$\mathbb{E}_{\mathrm{x}\sim P}[\alpha f(x) + \beta g(x)] = \alpha\mathbb{E}_{\mathrm{x}\sim P}[f(x)] + \beta\mathbb{E}_{\mathrm{x}\sim P}[g(x)] \tag{19}$$

- **Variance**
  Variance means how much the values of a function of a random variable x vary as we sample different values of x from its distribution:

$$Var_{\mathrm{x}\sim P}[f(x)] = \mathbb{E}_{\mathrm{x}\sim P}[\{f(x) - \mathbb{E}[f(x)]\}^2] \tag{20}$$

  The square root of the variation called $standard\ deviation$.
- **Covariance**
  Covariance gives information about how two values are linearly related, as well as the scale of these variables:

$$Cov_{\mathrm{x,y}}[f(x), g(y)] = \mathbb{E}_{\mathrm{x,y}}[\{f(x) - \mathbb{E}_{\mathrm{x}}[f(x)]\}\{g(y) - \mathbb{E}_{\mathrm{y}}[g(y)]\}^T] \tag{21}$$

Frequentist vs Bayesian

Who are the bayesian?

- **Frequentist**
  1. "Probabilities represent long run frequencies of events."
  2. probabilities are fundamentally related to frequencies of events.
  3. **Do not use** subjective information.
  4. it is meaningless to talk about the probability of the parameter Θ: the parameter Θ is (by definition) a single fixed value, and to talk about a frequency distribution for a **fixed value** is nonsense.
- **Bayesian**
  1. "Probability is used to quatify our uncertainty about something or precisely degree of belief."
  2. probabilities are fundamentally related to our own knowledge about an event.
  3. Use **subjective information**.
  4. we can meaningfully talk about the probability that the paramater Θ lies in a given range. That probability codifies our knowledge of the value based on **prior information** and/or available data.

Cox's Theorem

Cox's theorem implies that any plausibility model that meets the postulates is equivalent to the subjective probability model, i.e., can be converted to the probability model by rescaling.

Prior, Likelihood, and Posterior

Bayes' Theorem was used to

- convert **a prior belief** $\mathbb{P}(H)$
- into **a posterior belief** $\mathbb{P}(H|D)$
- by incorporating **observed data**. $\mathbb{P}(D|H)$

$$\mathbb{P}(H|\mathcal{D}) = \frac{\mathbb{P}(H)\mathbb{P}(D|H)}{\mathbb{P}(D)} = \frac{\mathbb{P}(H)\mathbb{P}(D|H)}{\sum_H \mathbb{P}(H)\mathbb{P}(D|H)} \tag{22}$$

Let's specify the hypothesis H as a hypothesis about model parameter θ. More intuitively, we can describe each term as follows:

- **Prior** $\mathbb{P}(\theta)$: Assumption about parameter θ.
- **Posterior** $\mathbb{P}(\theta|\mathcal{D})$: Evaluation of uncertainty in θ after we have seen data $\mathcal{D}$.
- **Likelihood** $\mathbb{P}(\mathcal{D}|\theta)$: The effect of observed data $\mathcal{D}$ given setting of the parameter θ.

$$(Posterior) \propto (Likelihood) \times (Prior) \tag{23}$$

Estimation and Likelihood Function

What is likelihood function?

The quantity $\mathbb{P}(\mathcal{D}|\theta)$ on the right-hand side of Bayes' theorem is evaluated for the observed data set $\mathcal{D}$ and can be viewed as a function of the parameter θ, in which case it is called the likelihood function.

$$L_{\mathcal{D}}(\theta) = \mathbb{P}(\mathcal{D}|\theta) \tag{24}$$

The likelihood function expresses how probable the observed data set is for different settings of the parameter θ. Note that the likelihood is not a probability distribution over θ, and its integral with respect to θ does not (necessarily) equal one.

Random Sample (i.i.d condition)

Consider the case where our data $\mathcal{D} = (x_1, \ldots, x_N)$ is a random sample of size N from the distribution of a random variable x taking values in $\mathbb{R}$, with probability density function $p(x|\theta)$.

$$L_{\mathcal{D}}(\theta) = \prod_{i=1}^{N} p(x_i|\theta) \quad \text{where} \quad \mathcal{D} = (x_1, \ldots, x_N) \tag{25}$$

or equivalently,

$$\ln[L_{\mathcal{D}}(\theta)] = \sum_{i=1}^{N} \ln[p(x_i|\theta)] \quad \text{where} \quad \mathcal{D} = (x_1, \ldots, x_N) \tag{26}$$

Frequentist and Maximum Likelihood

Maximum Likelihood Estimation

A widely used frequentist estimator is maximum likelihood, in which θ is set to the value that maximizes the likelihood function $L_{\mathcal{D}}(\theta)$. This corresponds to choosing the value of θ for which the probability of the observed data set is maximized.

$$\theta^* = \arg\max_{\theta}[L_{\mathcal{D}}(\theta)] \tag{27}$$

In the machine learning literature, the negative log of the likelihood function is called an error function.

$$\theta^* = \arg\min_{\theta} E(\theta) = \arg\min_{\theta}[-\ln L_{\mathcal{D}}(\theta)] \tag{28}$$

e.g. Univariate Gaussian Distribution

Suppose that $\mathrm{x} \sim \mathcal{N}(x|\mu, \sigma^2)$ with $\mathrm{i.i.d\ condition}$. Then likelihood function in this case becomes

$$L_{\mathcal{D}}(\theta) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \sigma^2) \tag{29}$$

and hence the loglikelihood function becomes

$$\ln[L_{\mathcal{D}}(\theta)] = \sum_{i=1}^{N}[\mathcal{N}(x_i|\mu, \sigma^2)] = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 - \frac{N}{2}\ln[\sigma^2] - \frac{N}{2}\ln[2\pi] \tag{30}$$

Frequentist and Maximum Likelihood (conti. . .)

Significant Limitation of Maximum Likelihood

Maximizing (30) with repect to μ and σ^2, respectively, we obtain the maximum likelihood solutions given by

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{31}$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu_{\mathrm{ML}})^2 \tag{32}$$

which is the $sample\ mean$ and $sample\ variance$, respectively.

So, what's the problem?

Here we give an indication of the limitation of the MLE in the context of our solutions for the ML parameters. At first consider the expectations of our solutions.

$$\mathbb{E}[\mu_{\mathrm{ML}}] = \mu \tag{33}$$

$$\mathbb{E}[\sigma^2_{\mathrm{ML}}] = (\frac{N-1}{N})\sigma^2 \tag{34}$$

so that on average the maximum likelihood estimate will obtain the correct mean but will underestimate the true variance by a factor $(N-1)/N$. In fact, the issue of bias in maximum likelihood lies at the root of the overfitting problem that we encountered earlier in the context of polynomial curve fitting.

Frequentists' Prediction

Having determined the parameters θ, we can now make predictions for new values of x. Because we now have a probabilistic model, these are expressed in terms of the predictive distribution that gives the probability distribution over x, rather than simply a point estimate, and is obtained by substituting the maximum likelihood parameters into μ and σ^2:

$$p(x|\mu_{\mathrm{ML}}, \sigma^2_{\mathrm{ML}}) = \mathcal{N}\left(x \middle| \frac{1}{N}\sum_{i=1}^{N} x_i, \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\mathrm{ML}})^2\right) \tag{35}$$

Bayesian: Prior and Posterior

Suppose that we have an observable random variable x for an experiment, and the distribution of x depends on a parameter θ. We will denote the p.d.f of x for a given value of θ by $f(x|\theta)$.

$$data: \quad \mathrm{x} \sim f(x|\theta) \tag{36}$$

In $bayesian\ analysis$, we treat the parameter θ as a random variable, with a given probability density function $h(\theta)$ for random variable θ. The corresponding distribution is called the $priordistribution$ of θ and is intended to reflect our prior knoweledge (if any) of the parameter, before we gather data.

$$prior: \quad \theta \sim h(\theta) \tag{37}$$

After observing the data $\mathrm{x} = x$, we then use Bayes' theorem, to compute the conditional probability density function of θ given $\mathrm{x} = x$.

Bayesian: Prior and Posterior (conti. . .)

Before going through the derivation of this theorem, first recall that the joint probability density function of (x, θ) is the mapping given by

$$(x, \theta) \mapsto h(\theta) f(x|\theta) \tag{38}$$

Next, recall that the marginal probability density function f of x is given by

$$f(x) = \sum_{\theta \in \Theta} h(\theta) f(x|\theta) \tag{39}$$

or equivalently,

$$f(x) = \int_{\Theta} h(\theta) f(x|\theta) \mathrm{d}\theta \tag{40}$$

Finally, the conditional probability density function of θ given $\mathrm{x} = x$ is

$$posterior: \quad h(\theta|x) = \frac{h(\theta) f(x|\theta)}{f(x)} \tag{41}$$

called the $posterior\ distribution$, and is an updated distribution, given the information in the data. Note that $f(x)$ is simply the *normalizing constant* for the function $(x, \theta) \mapsto h(\theta) f(x|\theta)$.

Bayesian and Maximum A Posteriori

Maximum A Posteriori Estimation

Now, let's consider the case where the input data $\mathcal{D} = (x_1, \ldots, x_N)$ is a *random sample* of size N, each entry of which is from the distribution of a random variable x.

$$\mathcal{D} = (x_1, \ldots, x_N) \tag{42}$$

Specifically, suppose that x takes values in a set, says, $\mathbb{R}$ and has probability density function $g(x|\theta)$ for $x \in \mathbb{R}$, given $\theta = \theta$.

$$\mathrm{x} \sim g(x|\theta) \tag{43}$$

We can now determine θ by finding the most probable value of θ given the data, in other words by maximizing the posterior distribution. This technique is called maximum posterior, or simply MAP.

$$\theta^* = \arg\max_{\theta}[h(\theta|\mathcal{D})] = \arg\max_{\theta}[\prod_{i=1}^{N} g(\theta|x_i)] \tag{44}$$

or equivalently,

$$\theta^* = \arg\min_{\theta}[-\ln \sum_{i=1}^{N} g(\theta|x_i)] \tag{45}$$

Fully Bayesian Approach for Prediction

Although we have included a prior distribution $p(\theta|\mathcal{D})$, we are so far still making a point estimate of θ and so this does not yet amount to a Bayesian treatment. At first, our goal is to get probability distribution of x given the data $\mathcal{D} = (x_1, \ldots, x_N)$

$$predictive\ distribution: \quad \mathrm{x} \sim p(x|\mathcal{D}) \tag{46}$$

Now we should consistently apply the sum and product rules of probability, which requires that we integrate over all values of random variable θ. Such marginalizations lie at the heart of Bayesian methods for pattern recognition.

$$\begin{aligned} p(x|\mathcal{D}) &= \int_{\Theta} p(x, \theta|\mathcal{D}) \mathrm{d}\theta = \int_{\Theta} p(x|\theta, \mathcal{D}) p(\theta|\mathcal{D}) \mathrm{d}\theta \\ &= \int_{\Theta} p(x|\theta) p(\theta|\mathcal{D}) \mathrm{d}\theta \end{aligned} \tag{47}$$

Now, in (47), we can see that the predictive distribution $p(x|\mathcal{D})$ is the marginal probability distribution of posterior $p(\theta|\mathcal{D})$ with repect to the model parameter θ. And the probability distribution $p(x|\theta)$ is our assumption about probability distribution of data.

Inference Stage and Decision Stage

Before we discuss the Decision theory, recall the key role of Probability theory and Decision theory.

- **Probability Theory** provides a consistent framework for expressing an uncertainities in a precise and quantitative manner.
- **Decision Theory** allows us to exploit this probabilistic representation to make optimal prediction according to appropriate criteria.

As you can see here, Probability theory works as a consistent mathematical framework, which allows us to substitute a frequently observed event or a specific belief of uncertainty with a precise and quantitative mathematical expression. For example, we may determine the joint probability distribution $p(x, t)$ exploiting a set of training data. This is what we called $Inference\ stage$.

In practical sense, after the inference stage, we must often make a specific prediction, says, for the target value t, or, more generally, take a specific action based on our understanding of those probabilistic representations that has been obtained through the inference stage. For example, if we have the joint probability distrubtion $p(x, \mathcal{C}_k)$ for classification problem, although this can be very useful and informative quantity, we must decide which class $\mathcal{C}_k$ the value x belongs to. This is what we called $Decision\ stage$ and, thanks to Decision theory, we can make optimal decision given the appropriate probabilities from the situations involving uncertainties.

Inference Stage and Decision Stage (conti. . .)

Inference stage
provides probabilistic representation which discribes the situation in a stochastic way.

- By statistical inferences, we may make a joint or conditional probability distribution from the data set $\mathcal{D}$.

$$p(t|x,\theta) \quad \text{or} \quad p(t,x|\theta) \tag{48}$$

- Also, we may find a prediction distribution for a new target value t_{new} which is corresponding to the new input vaule x_{new} given the training data set $\mathcal{D}$.

$$p(t|x,\mathcal{D}) \quad \text{or} \quad p(t,x|\mathcal{D}) \tag{49}$$

Decision stage
tells us how to makes optimal decision, which is usally in terms of loss function, based on those probabilistic representation.

- It usally define loss function, and do the math based on our understanding of the value that t is likely to take.
- Through that mathematical reasoning, we may makes optimal decision and action to take.

More Mathematical Interpretation of Decision Theory

In general sense, we also can view the decision stage as:
On a unknown, hidden state of 'nature' $s \in \mathcal{S}$, choosing the action a from a set of possible actions $\mathcal{A}$ we can take, given observed data $x \in \mathcal{X}$ to minimize the total loss $\mathbb{E}(L)$ which is expactation value of every loss $L(y, a) \forall a \in \mathcal{A}$ in such state.

$$\begin{aligned} \text{state}: &\quad y \in \mathcal{Y} \\ \text{new data}: &\quad x \in \mathcal{X} \\ \text{action}: &\quad a \in \mathcal{A} \\ \text{loss}: &\quad L(y, a) \end{aligned} \tag{50}$$

And the goal can be expressed as:

$$\text{Goal}: \quad \delta(\mathrm{x}) = \arg\min_{a \in \mathcal{A}} \mathbb{E}[L(y, a)] \quad \text{where} \quad \delta: \begin{array}{c} \mathcal{X} \to \mathcal{A} \\ x \mapsto a \end{array} \tag{51}$$

Loss Function

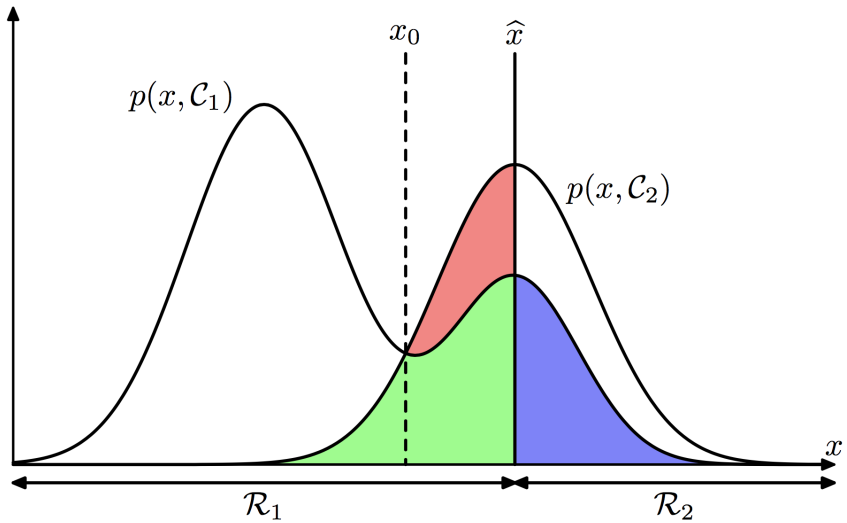

Figure 6: Schematic illustration of the joint probability distribution $p(x, \mathcal{C}_k)$.

Our goal in decision stage is to make a decision to minimize the $loss\ function$ given a new input x based on the probabilistic representation. The loss function may be denoted in several ways as follows:

$$\text{loss function}: \quad L(t, \hat{t}) \quad \text{or} \quad L(t, x) \quad \text{or} \quad L(t, y(x)) \tag{52}$$

At these notations, t is the true target value corresponding to x, and $\hat{t}$ is the decision based on a specific rule y given the new observation x, so that the function L is a single, overall measure of loss incured in taking the decision $\hat{t}$ but the true value being t given x. Let's look at two examples to help you understand.

Loss Function (conti. . .)

e.g. classification problem

In classification problem, we should classify $x \in S$ into some class $\mathcal{C}_k \; \forall k \in \{1, \ldots, K\}$. Let a rule $\mathcal{R}_k$ be the subset of S and the rule assigns all points $x \in \mathcal{R}_k$ into $\mathcal{C}_k$, then we can define the loss function as follows:

$$\forall k, j \in \{1, \ldots, K\}; \quad L(t, x) = L(\mathcal{C}_k, x \in \mathcal{R}_j) = \begin{cases} 0 & (k = j) \\ L_{kj} & (k \neq j) \end{cases} \tag{53}$$

And we may also represent this function in matrix form, which is called $loss\ matrix$.

$$\boldsymbol{L} = \begin{cases} 0 & (i = j) \\ L_{kj} & (i \neq j) \end{cases} \tag{54}$$

e.g. regression problem

In regression problem, we should find some value $\hat{t}$ based on a specific rule y given the new observation $x \in S$. The loss incurred in making decision based on rule y can be defined as follows:

$$L(t, y(x)) \tag{55}$$

in some cases, we can use square term or absolute value term:

$$L(t, y(x)) = |t - y(x)| \quad \text{or} \quad (t - y(x))^2 \tag{56}$$

Minimizing Expected Loss

In the classification example, the optimal solution is the one which minimizes the loss function. However, the loss function depends on the true class $\mathcal{C}_k$, which is unknown. For a given input x, our uncertainty in the true class is expressed through the joint probability distribution $p(x, \mathcal{C}_k)$, which is evaluated through inference stage, and so we seek instead to minimize the average loss, where the average is computed with respect to this distribution, which is given by

$$\begin{aligned}\mathbb{E}[L] &= \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(x, \mathcal{C}_k)\, \mathrm{dx} \\ &= \sum_j \int_{\mathcal{R}_j} \sum_k \{L_{kj} p(x, \mathcal{C}_k)\}\, \mathrm{dx}\end{aligned} \tag{57}$$

Now, our goal is to choose the rule $\mathcal{R}_j$ which minimize $\sum_k \{L_{kj} p(x, \mathcal{C}_k)\}$ for each x. And we may use the product rule $p(x, \mathcal{C}_k) = p(\mathcal{C}_k|x)p(x)$ to eliminate the common factor of $p(x)$, then we get:

$$x \in \mathcal{R}_{j^*} \quad \text{s.t.} \quad j^* = \arg\min_j \sum_k \{L_{kj} p(x, \mathcal{C}_k)\} = \arg\min_j \sum_k \{L_{kj} p(\mathcal{C}_k|x)\} \tag{58}$$

where $p(\mathcal{C}_k, x)$ is the posterior. Hence, it is clear that the decision rule that minimizes the expected loss is the one that assigns each new x to the class j to minimize (58).

Reject Option

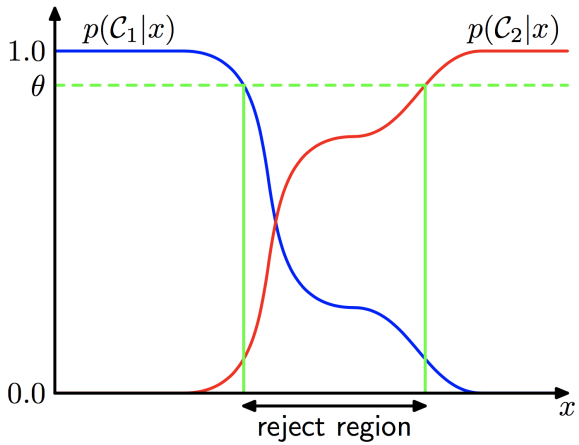

Figure 7: Schematic illustration of the joint probability distribution $p(x, \mathcal{C}_k)$.

In some cases, the largest of the posterior probabilities $p(\mathcal{C}_k|x)$ is significantly less than unity,

$$\frac{p(\mathcal{C}_k|x)}{\sum_k p(\mathcal{C}_k|x)} << 1 \tag{59}$$

or equivalently, the joint distribution $p(x, \mathcal{C}_k)$ have comparable values. For these cases where the optimal decision is not significantly superior to the alternative, it is relatively uncertain which class the value x is assigned to. Hence, in some applications, it will be appropriate to avoid making decisions on the difficult cases in anticipation of a lower error rate on those examples for which a classification decision is made. This is known as the *rejection option*.

$$\frac{\max_k p(\mathcal{C}_k|x)}{\sum_k p(\mathcal{C}_k|x)} \ll \theta \quad \text{where} \quad \frac{1}{k} \leq \theta \leq 1 \tag{60}$$

Gen or Dis

Now, step back and see the big picture, "the whole process of machine learning." We have been divide the machin learning process into two sub stages, inference step in which we use training data to learn a probabilistic model, says, $p(t|x)$, and decision step in which we use these probabilistic representation to make optimal decision. There is an ambiguity arises, how to learn probabilistic model, and what it looks like? Furthermore, we may consider the case where do not use probabilistic inference.
In fact, we can identify three distinct approaches to solving decision problems, all of which have been used in practical applications.

Generative Method
models how the data was generated in order to categorize a signal. It asks the question: based on my generation assumptions, which category is most likely to generate this signal?

Discriminative Method
A discriminative algorithm does not care about how the data was generated, it simply categorizes a given signal. This category can be subdivided again into two subcategories:

- **Probabilistic Discriminative Method** is a stochastic way, which may directly model $p(t|x)$ (e.g. logistic regression).
- **Deterministic Discriminative Method** is a non-stochastic way, which directly find function that directly maps x to target value t without any statistical inference.

Gen or Dis (cont. . .)

Now for your understanding, let's re-visit the classification problem. **Generative**

1. Solve the inference problem of determining the class-conditional densities $p(x|\mathcal{C}_k)$ for each class $\mathcal{C}_k$ individually.
2. Separately infer the prior $p(\mathcal{C}_k)$.
3. Use Bayes' theorem to find the posterior $p(\mathcal{C}_k|x)$.
4. Equivalently, we can model the joint distribution $p(x, \mathcal{C}_k)$ directly and then normalize to obtain the posterior $p(\mathcal{C}_k)$.
5. Having found the posterior $p(\mathcal{C}_k)$, we use decision theory to determine class membership for each new input x.

Probabilistic Discriminative

1. First solve the inference problem of determining the posterior $p(\mathcal{C}_k|x)$,
2. And then, use decision theory to assign each new x to one of the classes.

Deterministic Discriminative

1. Find a function $f(x)$, called a discriminant function, which maps each input x directly onto a class label.

Usefulness of Posterior

Although the stochastic inference stage requires much more cost, we will use them since there are many powerful reasons for wanting to compute the posterior.

Minimizing Risk

$$x \in \mathcal{R}_{j^*} \quad \text{s.t.} \quad j^* = \arg\min_j \sum_k \{L_{kj} p(\mathcal{C}_k|x)\} \tag{61}$$

Reject Option

$$\frac{\max_k p(\mathcal{C}_k|x)}{\sum_k p(\mathcal{C}_k|x)} \ll \theta \tag{62}$$

Compensating for class priors
sometimes we need to modify a data set to adjust an innate rarity of a specific event, such as cancer.

$$\text{True prior: } p(\mathcal{C}_k), \text{ Balanced posterior: } q(\mathcal{C}_k|x) \propto q(x|\mathcal{C}_k) q(\mathcal{C}_k) \tag{63}$$

Although the balanced dataset would allow us to find a more accurate model, we then have to compensate for the effects of our modifications to the training data.

$$\text{Adjusted posterior:} \quad \tilde{p}(\mathcal{C}_k|x) \propto q(\mathcal{C}_k|x) \frac{p(\mathcal{C}_k)}{q(\mathcal{C}_k)} \tag{64}$$

Usefulness of Posterior (conti. . .)

Combining models
For complex applications, we may wish to break the problem into a number of smaller subproblems each of which can be tackled by a separate module. In some cases where more than a type of data is available, rather than combine all of this heterogeneous information into one huge input space, it may be more effective to build more than one system to interprete each type of data.

One simple way to do this is to assume that, for each class separately, the distributions of inputs, denoted by x_1 and x_2, are independent, so that

$$P(x_1, x_2|\mathcal{C}_k) = P(x_1|\mathcal{C}_k)P(x_2|\mathcal{C}_k) \quad \text{where} \quad x_1 \perp x_2 \,|\, \mathcal{C}_k \tag{65}$$

The posterior is then given by

$$\begin{aligned} P(\mathcal{C}_k|x_1, x_2) &\propto p(x_1|\mathcal{C}_k)p(x_2|\mathcal{C}_k)p(\mathcal{C}_k) \\ &\propto \frac{p(\mathcal{C}_k|x_1)p(\mathcal{C}_k|x_2)}{p(\mathcal{C}_k)} \end{aligned} \tag{66}$$

Self Information

Principle of quantifying Information

1. Likely events should have low information content. Less likely events should have higher information content.
2. Independent events should have additive information.

Self-information
By above principle, we can define self-information like:

$$h(x) = -\ln P(x) \tag{67}$$

Shannon Entropy
We can quantify uncertainty of whole distribution like:

$$H[x] = \mathbb{E}_{x\sim P}[I(x)] = -\mathbb{E}_{x\sim P}[\ln P(x)] \tag{68}$$

Shonnon entrophy means expected amount of information in an event drawn from that distribution.

Entrophy (discrete)

Kullbeck-Leibler(KL) Divergence
We sometimes want to know how different two distributions are. In this case, we use KL divergence defined by:

$$KL(P||Q) = \mathbb{E}_{x\sim P}[\ln \frac{P(x)}{Q(x)}] = \mathbb{E}_{x\sim P}[\ln P(x) - \ln Q(x)] \tag{69}$$

And it means extra amount of information needed to send a message containing symbols drawn from probability distribution P, when we use a code that was designed to minimize the length of messages drawn from probability distribution.
Cross-Entrophy [2]
Cross-entrophy which means the average number of information needed to encode data coming from a source with distribution P when we use model q to define our codebook. (not extra)

We can define cross-entrophy like:

$$H(P,Q) = H(P) + KL(P||Q) = -\mathbb{E}_{x\sim P}[\ln Q(x)] \tag{70}$$


Christopher M. Bishop. **Pattern Recognition and Machine Learning**. Springer, 2006.


Ian Goodfellow, Yoshua Bengio, and Aaron Courville. **Deep Learning**. `http://www.deeplearningbook.org`. MIT Press, 2016.


Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. **Foundations of Machine Learning**. The MIT Press, 2012. ISBN: 026201825X, 9780262018258.