Applied Math and Machine Learning Basics[1]

Nic Becker

July 11, 2019

[1] Chapters 3, 4, & 5 of *Deep Learning* (Goodfellow, Bengio, and Courville)

These are reading notes to *Deep Learning*'s opening chapters on basic probability and information theory, numerical computation, and machine learning basics. It's intended to be quick reference for practicing ML. If you're looking for linear algebra this is not your doc.

Probability and Information Theory

Why Probabability?

Three possible sources of uncertainty:

- Inherent stochasticity in the system being modeled (e.g. quantum mechanics, deck of cards)
- Incomplete observability (e.g. Monty Hall problem)
- Incomplete modeling (e.g. name any social science)

Probability dealing with the rate at which events occur are known as **frequentist probability**. On the other hand, probability dealing with qualitative levels of certainty, like **degree of belief**, is known as **Bayesian probability**.

Logic provides a set of formal rules for determining what propositions are implied to be true or false given the assumption that some other set of propositions are true or false. Probability theory provides a set of formal rules for determining the likelihood of a proposition being true given the likelihood of other propositions.

Random Variables

A **random variable** is a variable that can take on dfiferent values randomly.

Can be discrete or continuous, continuous random variables are real.

Discrete Variables and Probability Mass Functions

A **probability distribution** is a description of how likely a random variable or set of random variables is to take on each of its possible states.

Probability mass functions (PMF) are denoted by a capital P. Common notations: $P(x), P(\mathrm{x}= x), x \sim P(\mathrm{x})$. They must satisfy:

- The domain of P msut be the set of all possible states of x.

- $\forall x \in \mathrm{x}, 0 \leq P(x) \leq 1$. An impossible event has a probability 0. Likewise, an event that is guaranteed to happen has probability 1.
- $\sum_{x \in \mathrm{x}} P(x) = 1$, i.e. the probability is **normalized**.

When PMF's act on many variables at the same time they are known as **joint probability distributions**.

Continuous Variables and Probability Density Functions

Integrating a **probability density function (PDF)** yields the probability of landing between two states. A function p must satisfy:

- The domain of p must be the set of all possible states of x.
- $\forall x \in \mathrm{x}, p(x) \geq 0$. We do not require $p(x) \leq 1$.
- $\int p(x)dx = 1$

Marginal and Conditional Probability

The probability over a subset of variables is known as the **marginal probability distribution**. For example, suppose we have discrete random variables x and y, and we know $P(x, y)$. We can find $P(x)$ with the **sum rule**.

$$\forall x \in x, P(\mathrm{x} = x) = \sum_{y} P(\mathrm{x} = x, \mathrm{y} = y)$$

For continuous variables:

$$p(x) = \int p(x, y)dy$$

The probability of some event given some other event has happened is **conditional probability**. "Probability of y given x" is denoted by $P(\mathrm{x} = x \mid \mathrm{y} = y)$.

$$P(\mathrm{x} = x \mid \mathrm{y} = y) = \frac{P(\mathrm{x} = x, \mathrm{y} = y)}{P(\mathrm{x} = x)}$$

Computing conditional probability is not the same as computing what would happen if some action were undertaken. Computing consequences of an action (**intervention query**) falls in the domain of **causal modeling**.

The Chain Rule of Conditional Probabilities

Joint probability distributions over many random variables can be decomposed into conditional distributions over only one variable:

$$P(x^{(1)}, ..., x^{(n)} = P(x^{(1)}) \Pi_{i=1}^{n} P(x^{(i)} \mid x^{(1)}, ..., x^{(i-1)})$$

Independence and Conditional Independence

Two random variables x and y are **independent** if their probability distribution can be expressed as a product of two factors, one involving only x and one involving only y.

Two random variables x and y are **conditionally independent** given a random variable z if the conditional probability distribution over x and y factorizes in this way for every value of z:

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

Compact notation: $\mathrm{x} \perp \mathrm{y}$ are independent. $\mathrm{x} \perp \mathrm{y} \,|\, \mathrm{z}$ are conditionally independent.

Expectation, Variance, and Covariance

The **expectation**, or **expected value**, of some function $f(x)$ with respect to a probability distribution $P(x)$ is the average, or mean value, that f takes on when x is drawn from P. For discrete variables:

$$\mathop{\mathbb{E}}_{x \sim p}[F(x)] = \sum_x P(x) f(x)$$

For continuous variables:

$$\mathop{\mathbb{E}}_{x \sim p}[F(x)] = \int p(x) f(x) dx$$

More, expectations are linear:

$$\mathop{\mathbb{E}}_{x}[\alpha f(x) + \beta f(x)] = \alpha \mathop{\mathbb{E}}_{x}[f(x)] + \beta \mathop{\mathbb{E}}_{x}[g(x)]$$

Variance measures how much the values of a function of a random variable x vary as we sample different values of x from its probability distribution. The **standard deviation** is the square root of the variance.

$$\mathrm{Var}(f(x)) = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right]$$

The **covariance**[2] gives some sense of how much two value sare linearly related to each other, as well as the scale of these variables:

$$\mathrm{Cov}(f(x), g(x)) = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])(g(x) - \mathbb{E}[g(x)])\right]$$

The notions of covariance and dependence are related but distinct concepts. Two variables that are independent have zero covariance, and two variables that have nonzero covariance are dependent. For two variables to have zero covariance, there must be no linear dependence between them. Independence is a stronger requirement than zero covariance, because independence also excludes nonlinear relationships. It is possible for two variables to be dependent but have zero covariance.

[2] High absolute values of the covariance mean that the values change very much and are both far from their respective means at the same time. If the sign of the covariance is positive, then both variables tend to take on relatively high values simultaneously. If negative, then one variable tends to take on a relatively high value at the times thta the other takes on relatively low value and vice versa. Other measures such as **correlation** normalize the contribution of each variable in order to measure only how much the variables are related.

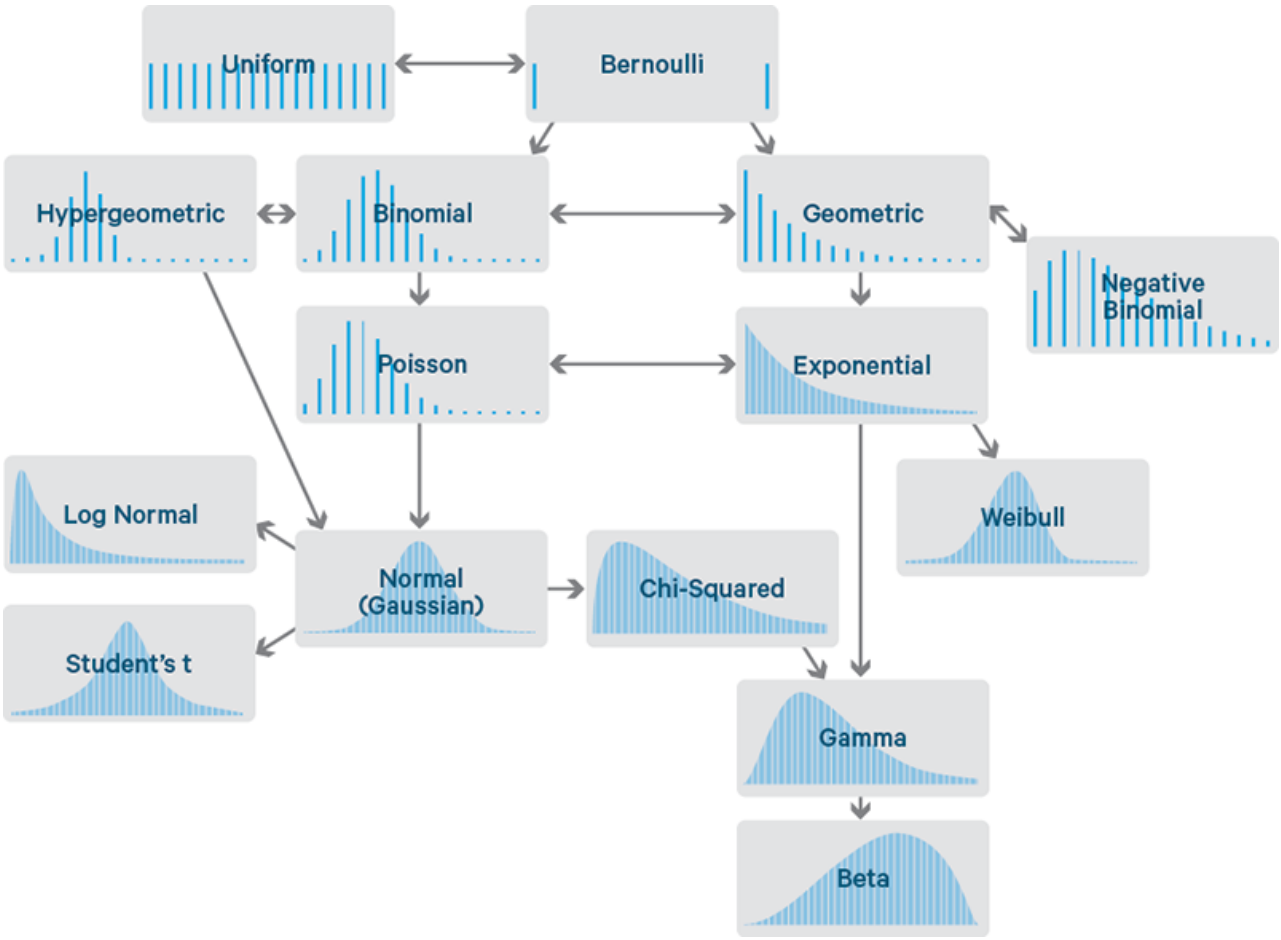

Figure 1: Relationships between common PMFs and PDFs

Common Probability Distributions

The **Bernoulli distribution** is a distribution over a single binary random variable, and is controlled by a single parameter $\phi \in [0, 1]$.

$$P(\mathrm{x} = 1) = \phi$$
$$P(\mathrm{x} = 0) = 1 - \phi$$
$$P(\mathrm{x} = x) = \phi^x (1 - \phi)^{1-x}$$
$$\mathop{\mathbb{E}}_{x}[x] = \phi$$
$$\mathrm{Var}_x(x) = \phi(1 - \phi)$$

The **multinoulli**, or **categorical, distribution**[3] is a distribution over a single variable with k different states, where k is finite.

The good ol' **Gaussian distribution**[4] or **normal distribution** is the most commonly used distribution over real numbers and a default choice in the absence of prior knowledge.

$$\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\Big(-\frac{1}{2\sigma^2}(x - \mu)^2\Big)$$

[3] See *Deep Learning* page 60 for more on this one.

[4] μ gives the coordinate of the central peak, and is the mean of the distribution $\mathbb{E}[x] = \mu$. The standard deviation is given by σ, and the variance by σ^2.

When we evaluate a PDF, we need to square and invert σ. A more efficient way of parameterizing the distribution is to use a parameter $\beta \in (0, \inf)$ to control the **precision**, or inverse variance, of the distribution:

$$\mathcal{N}(x; \mu, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\Big(-\frac{1}{2}\beta(x-\mu)^2\Big)$$

The **central limit theorem** shows that the sum of many independent random variables is approximately randomly distributed. Consequently, many complicated systems can be modeled successfully as normal distributed noise, even if the system can be decomposed into parts with more structured behavior. More, out of all the possible probability distributions with the same variance, the normal distribution encodes the maximum amount of uncertainty over the real numbers (think of it as inserting the least amount of prior knowledge into a model).

The **multivariate normal distribution**[5] is the generalization to $\mathbb{R}^n$:

[5] $\boldsymbol{\mu}$ gives the vector-valued mean of the distribution, $\boldsymbol{\Sigma}$ gives the covariance matrix of the distribution.

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^n \det \Sigma}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big)$$

Similar to the 1D example, we can use a **precision matrix** β since the covariance is not an efficient way to parameterize the disbtribution (since we would need to invert it):

We often fix the covariance matrix to be a diagonal matrix. An even simpler version is the **isotropic** Gaussian distribution, whose covariance matrix is a scalar times the identity matrix.

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\beta}^{-1}) = \sqrt{\frac{\det \boldsymbol{\beta}}{(2\pi)^n}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\beta}(\mathbf{x}-\boldsymbol{\mu})\Big)$$

Exponential distributions have a sharp point at $x = 0$, something often desired in deep learning.

The exponential distribution uses the **indicator function** $\mathbf{1}_{x \geq 0}$ to assign probability zero to all negative values of x.

$$p(x; \lambda) = \lambda \mathbf{1}_{x \geq 0} \exp(-\lambda x)$$

Laplace distribution is closely related and allows us to place a sharp peak of probability mass at an arbitrary point μ.

$$\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\Big(-\frac{|x-\mu|}{\gamma}\Big)$$

The **Dirac delta distribution**[6] is what you would expect:

[6] The Dirac delta distribution is only necessary to define the empirical distribution over continuous variables. For discrete variables, conceptualize an empirical distribution as a multinoulli distribution, with a probability associated with each possible input value that is simply equal to the **empirical frequency** of that value in the training set.

$$p(x) = \delta(x - \mu)$$

A common use of the Dirac delta distribution is as a component of an **empirical distribution**,

$$\hat{p}(\mathbf{x}) = 1/m \sum_{i=1}^{m} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

which puts probability mass $1/m$ on each of the m points $\mathbf{x}^{(1)}...\mathbf{x}^{(i)}$, forming a given data set or collection of samples.

We can combine distributions to construct a **mixture distribution**. This can be useful in thinking about **latent variables**, or random variables that we cannot observe directly.[7]

[7] See *Deep Learning* page 65.

Useful Properties of Common Functions

This section covers the **logistic sigmoid** function which can be used to produce the ϕ parameter of a Bernoulli distribution and the **softplus function** $\zeta(x) = \log(1 + \exp(x))$ which can be useful for producing σ or β of a normal distribution. These properties are handy:

$$\sigma(x) = \frac{\exp(x)}{\exp(x) + \exp(0)}$$

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$\sigma(x)(1 - \sigma(x)) = /sigma(-x)$$

$$\log\sigma(x) = -\zeta(-x)$$

$$\frac{d}{dx}\zeta(x) = \sigma(x)$$

$$\forall x \in (0, 1), \sigma^{-1}(x) = log(\frac{x}{1 - x})$$

$$\forall x \geq 0, \zeta^{-1}(x) = \log(\exp(x) - 1)$$

$$\zeta(x) = \int_{-\inf}^{x} \sigma(y)dy$$

$$\zeta(x) - \zeta(-x) = x$$

Justification for the name *softplus*. The function is a smoothed version of the **positive part function,** $x^+ = \max\{0, x\}$

$\sigma^{-1}(x)$ is called the **logit** in statistics, but this is rarely used in ML.

Bayes' Rule

$$P(\mathrm{x}|\mathrm{y}) = \frac{P(\mathrm{x})P(\mathrm{y}|\mathrm{x})}{P(\mathrm{y})}$$

Information Theory

Information theory is a branch of applied mathematics that revolves around quantifying how much information is present in a signal. It is useful in designing optimal codes and calculating the expected length of messages sampled from specific probability distributions using various encoding schemes. In the context of ML, we can also apply information theory to continuous variables where some of these message length interpretations do not apply. Learning that an unlikely event has occurred is more informative than learning that a likely event has occurred.

- Likely events should have low information content, and in the extreme case, events that are guaranteed to happen should have no information content whatsoever.
- Less likely events should have higher information content.
- Independent events should have additive information. For example, finding out that a tossed coin has come up as heads twice should convey twice as much information as finding out that a tossed coin has come up as heads once.

We define the **self-information**[8] of an event $\mathrm{x} = x$ to be

$$I(x) = -\ln P(x)$$

When x is continuous, we use the same definition of information by analogy, but some of the properties from the discrete case are lost. For example, an event with unit density still has zero information, despite not being an event that is guaranteed to occur.

Self-information deals only with a single outcome. **Shannon entropy**[9] quantifies the amount of uncertainty in an entire probability distribution:

$$H(x) = \mathop{\mathbb{E}}_{x\sim P}[I(x)] = -\mathop{\mathbb{E}}_{x\sim P}[\log P(x)]$$

Distributions that are nearly deterministic have low entropy; distributions that are closer to uniform have high entropy. When x is continuous, the Shannon entropy is known as the **differential entropy**.

If we have two separate probability distributions $P(x)$ (classically, the true distribution) and $Q(x)$ (approximate distribution) over the same random variable x, we can measure how different these distributions are using the **Kullback-Leiber (KL) Divergence**:

$$D_{KL}(P||Q) = \mathop{\mathbb{E}}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathop{\mathbb{E}}_{x\sim P}[\log P(x) - \log Q(x)]$$

In the case of discrete variables, it is the extra amount of information (measured in bits if we use the base-2 logarithm, but in ML we usually use nats and the natural log) 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 the messages drawn from probability distribution Q.

- KL divergence is non-negative.
- KL divergence is 0 if and only if P and Q are the same distribution in the case of discrete variables, or equal "almost everywhere" in the case of continuous variables.

[8] $I(x)$ is written in units of **nats**. One nat is the amount of information gained by observing an event of probability 1/e. Other texts use base-2 logarithms and units called **bits** or **shannons**. This is equivalent to reducing the amount of uncertainty by a factor of 2.

[9] The expected amount of information in an event drawn from that distribution. **It gives a lower bound on the number of nats needed on average to encode symbols drawn from a distribution** P.

- It is often conceptualized as measuring some sort of distance between two distributions.
- It is not a true distance measure because it is not symmetric: $D_{KL}(P||Q) \neq D_{KL}(Q||P)$[10]
- Optimizing $D_{KL}(P||Q)$: Wherever P has high probability, Q must have high probability (Forward KL, mean-seeking behavior)
- Optimizing $D_{KL}(Q||P)$: Wherever Q has high probability, P must have high probability (Reverse KL, mode-seeking behavior)

[10] Check figure 3.6 in *Deep Learning*.

More: https://dibyaghosh.com/blog/probability/kldivergence.html

A quantity that is closely related to KL divergence is the **cross-entropy** $H(P, Q) = H(P) + D_{KL}(P||Q)$, which is similar to the KL divergence but lacking the term on the left:

$$H(P, Q) = - \mathop{\mathbb{E}}_{x \sim P} \log Q(x)$$

We can rewrite the objective as cross-entropy - entropy of P:

$$D_{KL}(P||Q) = H(P, Q) - H(P)$$

» Quora: Intuitive way to think about cross-entropy «

Minimizing the cross-entropy with respect to Q is equivalent to minimizing the KL divergence, because Q does not participate in this omitted term. When computing many of these quantities, it is common to encounter expressions of the form $0 \log(0)$. By convention, we treat these expressions as $\lim_{x \to 0} x \log x = 0$.

» Chris Olah: Visual Information Theory«

Structured Probabilistic Models

Not sure how helpful this section is for reference.

ML algorithms often involve probability distributions over a very large number of random variables. Often, these probability distributions involve direct interactions between relatively few variables. Instead of using a single function to represent the entire joint probability distribution, we split the distribution into factors that we multiply together to greatly reduce the number of parameters needed to describe the distribution:

Suppose a influences the value of b, and b influences the value of c, but that a and c are independent given b.

$$p(a, b, c) = p(a)p(b \mid a)p(c \mid b)$$

Each factor uses a number of parameters that is exponential in the number of variables in the factor. We can greatly reduce the cost of representing a distribution if we are able to find a factorization into distributions over fewer variables.

We can describe these factorization using graphs.[11] When we

[11] "graph" in the sense of graph theory: a set of vertices that may be connected to each other with edges

represent the factorization of a probability distribution with a graph, we call it a **structured probabilistic model**, or a **graphical model**.

There are two kinds of graphical models: directed and undirected. Both kinds use a graph $\mathcal{G}$ in which each node corresponds to a random variable, and an edge connecting two random variables means that the probability distribution is able to represent direct interactions between those two random variables.

Pictures are really helpful here, but I'm too lazy to include them. Check DL page 76 or google.

Directed models use graphs with directed edges (arrows) and represent factorizations into conditional probability distributions (example above). A directed model contains one factor for every random variable x_i in the distribution, adn that factor consists of the conditional distribution over x_i given the parents of x_i, denoted $Pa_{\mathcal{G}}(\mathrm{x}_i)$:

$$p(\mathrm{x}) = \Pi_i p(\mathrm{x}_i | Pa_{\mathcal{G}}(\mathrm{x}_i))$$

Undirected models use graphs with undirected edges and represent factorizations into a set of functions; unlike in the directed case, these functions are usually not probability distributions of any kind. Any set of nodes that are all connected to each other in $\mathcal{G}$ is called a clique. Each clique $\mathcal{C}^{(i)}$ in an undirected model is associated with a factor $\phi^{(i)}(\mathcal{C}^{(i)})$. These factors are just functions, not probability distributions. The output of each factor must be non-negative, but there is no constraint that the factor must sum or integrate to 1 like a probability distribution.

The probability of a configuration of random variables is **proportional** to the product of all these factors—assignments that result in larger factor values are more likely. Divide by normalization constant in order to obtain normalized probability distribution:

$$p((x)) = \frac{1}{Z} \Pi_i \phi^{(i)}(\mathcal{C}^{(i)})$$

Numerical Computation

link to chapter in Deep Learning

Machine Learning Basics

link to chapter in Deep Learning