Mathematics for Machine Learning

Garrett Thomas

1 About

Machine learning uses tools from a variety of mathematical fields. This document is intended to summarize the mathematical background needed for an introductory class in machine learning, which at UC Berkeley is known as CS 189. We will cover topics in multivariable calculus/optimization, linear algebra, and probability. Our assumption is that the reader is already familiar with single-variable calculus and has encountered the basic concepts of multivariable calculus and matrix algebra (at the level of UCB Math 53/54). We emphasize that this document is **not** a replacement for the prerequisite classes.

 Please report any mistakes to gwthomas@berkeley.edu.

Contents

2 Notation

Notation	Meaning
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the set (vector space) of n-tuples of real numbers, endowed with the usual inner product
$\mathbb{R}^{m \times n}$	the set (vector space) of m-by-n matrices
$\nabla f(x)$	the gradient of the function f evaluated at some point x
$\nabla^2 f(x)$	the Hessian of the function f evaluated at some point x
A^{T}	the transpose of the matrix A
Ω	sample space
$\mathbb{P}(A)$	the probability of event A
$\mathbb{P}(A \mid B)$	the probability of event A, given B
A^c	the complement of event A
$\mathbb{E}[X]$	the expected value of random variable X
$\mathrm{Var}(X)$	the variance of random variable X
$\mathrm{Cov}(X, Y)$	the covariance of random variables X and Y
$\lvert S \rvert$	cardinality of set S

Other notes:

- We will restrict ourselves to real values. In many places in this document, it is entirely possible (and in fact natural) to generalize to the complex case, but we will simply state the version that applies to the reals.
- We assume that vectors are *column vectors*, that is, a vector in $\mathbb{R}^n$ can be interpreted as an n-by-1 matrix. As such, taking the transpose of a vector is well-defined (and produces a *row vector*, i.e. a 1-by-n matrix).
- We do not provide proofs of the theorems given in this document. The proofs are not the point – we wish only to review important definitions and concepts.

3 Calculus and optimization

Much of machine learning is about minimizing a *loss function* (also called an *objective function* in the optimization community), which is a scalar function of several variables that typically measures how poorly our model fits the data we have. We will not give specific examples of loss functions here (go to class for these!) but we will assume here that our loss function has the form $f : \mathbb{R}^n \to \mathbb{R}$ and is sufficiently differentiable.

3.1 Gradients

The single most important concept from calculus in the context of machine learning is the *gradient*. Gradients are a generalization of the derivative to scalar functions of several variables. The gradient of $f(x_1, \ldots, x_n)$, denoted ∇f, is given by

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Gradients have a very important property: ∇f points in the direction of *steepest ascent*. We will use this fact frequently when iteratively minimizing a function via *gradient descent*:

$$\theta_{k+1} = \theta_k - \eta_k \nabla f(\theta_k)$$

where $\eta_k > 0$ is called the *step size*.

3.2 Hessians

The *Hessian* is a matrix of second-order partial derivatives:

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

i.e. $(\nabla^2 f)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$. The Hessian is used in some optimization algorithms, such as Newton's method. It is expensive to calculate but can drastically reduce the number of iterations needed to converge to a local minimum by providing information about the curvature of f.

3.3 Extrema

Optimization is about finding *extrema*, which depending on the application could be minima or maxima. A point x is said to be a *local minimum* of f if $f(x) \leq f(y)$ for all y in some neighborhood about x, and similarly a *local maximum* of f if $f(x) \geq f(y)$ for all y in some neighborhood about x. Furthermore, if $f(x) \leq f(y)$ for all y in the entire domain of f, then x is a *global minimum* of f (similarly for global maximum).

Here is a useful criterion: if f is differentiable, then for any extremum x, $\nabla f(x) = 0$. Note that the converse does not hold in general, that is, $\nabla f(x) = 0$ does not imply that x is an extremum. It could be a *saddle point* of f.

3.4 Convexity

A function f is said to be *convex* if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

for all $x, y \in \text{dom}\, f$ and any $\lambda \in [0,1]$. Geometrically, this means that the line segment between two points on the graph of f lies on or above the graph itself.

It can be shown that if f is twice differentiable, then it is convex iff the Hessian $\nabla^2 f$ is positive semi-definite.

Convexity sounds like a very mysterious property, but it has some wonderful implications:

- Any local minimum of a convex function f is a global minimum of f.
- There are efficient algorithms to minimize convex functions.

Unfortunately, many loss functions that we would like to minimize are non-convex. This means whatever local minima our optimization algorithm finds may not be globally optimal.

4 Linear Algebra

4.1 Eigenthings

For a square matrix $A \in \mathbb{R}^{n \times n}$, there may be vectors which, when A is applied to them, are simply scaled by some constant. Mathematically we say that a nonzero vector $x \in \mathbb{R}^n$ is an *eigenvector* of A corresponding to *eigenvalue* $\lambda \in \mathbb{R}$ if $Ax = \lambda x$. (We exclude the zero vector from this definition because $A(0) = \lambda \cdot 0$ for every $\lambda \in \mathbb{R}$.)

Suppose $\lambda_1, \ldots, \lambda_k$ are the eigenvalues of A. The *trace* and *determinant* of A are the sum and product (respectively) of all A's eigenvalues:

$$\operatorname{tr}(A) = \sum_{i=1}^{k} \lambda_i, \quad \det(A) = \prod_{i=1}^{k} \lambda_i$$

4.2 Transposition

If $A \in \mathbb{R}^{m \times n}$, its *transpose* $A^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ is given by $(A^{\mathrm{T}})_{ij} = A_{ji}$ for each (i, j). In other words, the columns of A become the rows of A^{T}, and the rows of A become the columns of A^{T}.

The transpose has several nice algebraic properties that can be easily verified from the definition:

1. $(A^{\mathrm{T}})^{\mathrm{T}} = A$
2. $(A + B)^{\mathrm{T}} = A^{\mathrm{T}} + B^{\mathrm{T}}$
3. $(AB)^{\mathrm{T}} = B^{\mathrm{T}} A^{\mathrm{T}}$
4. $(\alpha A)^{\mathrm{T}} = \alpha A^{\mathrm{T}}$

4.3 Inner Products

The standard *inner product* of $x, y \in \mathbb{R}^n$ is denoted $x^{\mathrm{T}} y$ (or $x \cdot y$, hence the alternative name *dot product*) and is given by

$$x^{\mathrm{T}} y = \sum_{i=1}^{n} x_i y_i$$

Note that this is a special case of matrix multiplication where we regard the resulting 1×1 matrix as a scalar.

The inner product has several nice algebraic properties that can be easily verified from the definition:

1. $x^{\mathrm{T}} y = y^{\mathrm{T}} x$
2. $x^{\mathrm{T}}(y + z) = x^{\mathrm{T}} y + x^{\mathrm{T}} z$
3. $(\alpha x)^{\mathrm{T}}(\beta y) = \alpha \beta x^{\mathrm{T}} y$

Two vectors x and y are said to be *orthogonal* if $x^{\mathrm{T}} y = 0$.

4.4 Norms

A *norm* is a function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ that satisfies the following properties:

1. $\forall x \in \mathbb{R}^n, \|x\| \geq 0$, with $\|x\| = 0$ only if $x = 0$
2. $\forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \|\alpha x\| = |\alpha|\|x\|$
3. $\forall x, y \in \mathbb{R}^n, \|x + y\| \leq \|x\| + \|y\|$ (the so-called *triangle inequality*)

But these are general properties, and we will typically only be concerned with a few specific norms:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^{\mathrm{T}} x}$$

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Note that the 1- and 2-norms are special cases of the p-norm, and the ∞-norm is the limit of the p-norm as p tends to infinity.

Here's a fun fact: for any given finite-dimensional vector space V, all norms on V are equivalent in the sense that for two norms $\|\cdot\|_A, \|\cdot\|_B$, there exist constants $\alpha, \beta \in \mathbb{R}$ such that

$$\alpha\|x\|_A \leq \|x\|_B \leq \beta\|x\|_A$$

for all $x \in V$.

If x and y are orthogonal unit vectors (i.e. $\|x\| = \|y\| = 1$), then they are described as *orthonormal*.

4.5 The Cauchy-Schwarz Inequality

This inequality is sometimes useful in proving bounds:

$$|x^{\mathrm{T}} y| \leq \|x\|_2 \|y\|_2$$

for all $x, y \in \mathbb{R}^n$. Equality holds exactly when x and y are scalar multiples of each other.

4.6 Special Kinds of Matrices

There are several ways matrices can be classified. Each categorization implies some potentially desirable properties, so it's always good to know what kind of matrix you're dealing with.

4.6.1 Orthogonal Matrices

A matrix $Q \in \mathbb{R}^{n \times n}$ is said to be *orthogonal* if its columns are pairwise orthonormal. This definition implies that

$$Q^{\mathrm{T}} Q = Q Q^{\mathrm{T}} = I$$

or equivalently, $Q^{\mathrm{T}} = Q^{-1}$. A nice thing about orthogonal matrices is that they preserve inner products:

$$(Qx)^{\mathrm{T}}(Qy) = x^{\mathrm{T}}Q^{\mathrm{T}}Qy = x^{\mathrm{T}}Iy = x^{\mathrm{T}}y$$

A direct result of this fact is that orthogonal matrices also preserve 2-norms:

$$\|Qx\|_2 = \sqrt{(Qx)^{\mathrm{T}}(Qx)} = \sqrt{x^{\mathrm{T}}x} = \|x\|_2$$

4.6.2 Symmetric Matrices

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *symmetric* if it is equal to its own transpose ($A = A^{\mathrm{T}}$). A very important property of symmetric matrices is that they can be decomposed in the following manner:

$$A = Q\Lambda Q^{\mathrm{T}}$$

Here Q is an orthogonal matrix, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. This is referred to as the *eigendecomposition* or *spectral decomposition* of A.

4.6.3 Positive (Semi-)Definite Matrices

A symmetric matrix A is *positive definite* if for all nonzero $x \in \mathbb{R}^n$, $x^{\mathrm{T}}Ax > 0$. Sometimes people write $A > 0$ to indicate that A is positive definite. Positive definite matrices have all positive eigenvalues.

A symmetric matrix A is *positive semi-definite* if for all $x \in \mathbb{R}^n$, $x^{\mathrm{T}}Ax \geq 0$. Sometimes people write $A \geq 0$ to indicate that A is positive semi-definite. Positive semi-definite matrices have all nonnegative eigenvalues.

Positive definite and positive semi-definite matrices will come up very frequently! Note that since these matrices are also symmetric, the properties of symmetric matrices apply here as well.

As an example of how these matrices arise, the matrix $A^{\mathrm{T}}A \geq 0$ for any $A \in \mathbb{R}^{m\times n}$, since

$$x^{\mathrm{T}}(A^{\mathrm{T}}A)x = (Ax)^{\mathrm{T}}(Ax) = \|Ax\|_2^2 \geq 0$$

for any $x \in \mathbb{R}^n$.

4.7 Singular Value Decomposition

Singular value decomposition (SVD for short) is a widely applicable tool in linear algebra. Its strength stems partially from the fact that *every matrix* $A \in \mathbb{R}^{m\times n}$ has an SVD (n.b. even non-square matrices)! The decomposition goes as follows:

$$A = U\Sigma V^{\mathrm{T}}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix[1] with the *singular values* of A (denoted σ_i) on its diagonal. By convention, the singular values are ordered such that they are non-increasing as a function of the index, i.e.

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$$

[1]Some would protest that a diagonal matrix must be square. We simply mean that all the off-diagonal entries are zero.

The singular values of A are the square roots of the eigenvalues of $A^{\mathrm{T}}A$.

The columns of U are called the *left-singular vectors* of A, and they are eigenvectors of AA^{T}. (Try showing this!) The columns of V are called the *right-singular vectors* of A, and they are eigenvectors of $A^{\mathrm{T}}A$.

5 Probability

Probability theory provides powerful tools for modeling and dealing with uncertainty. In machine learning, we will use it extensively, particularly to construct and analyze classifiers.

5.1 Basics

Suppose we have some sort of randomized experiment (e.g. a coin toss, die roll) that has some fixed set of possible outcomes. We call this set the *sample space* and denote it Ω. Any subset $A \subseteq \Omega$ is called an *event*. A *probability distribution* over Ω specifies how likely each event is to occur. We write $\mathbb{P}(A)$ for the probability of event A.

Here are the basic rules of probability:

1. $\forall A \subseteq \Omega,\ 0 \leq \mathbb{P}(A) \leq 1$, with $\mathbb{P}(\varnothing) = 0$ and $\mathbb{P}(\Omega) = 1$
2. $\forall A, B \subseteq \Omega,\ \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$. Note that if A and B are mutually exclusive, then $\mathbb{P}(A \cap B) = 0$, giving $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$.

As a corollary to these, we have also that

$$\mathbb{P}(A) + \mathbb{P}(A^c) = 1$$

where A^c denotes the *complement* of A, that is, $A^c = \Omega - A$.

5.1.1 Conditional Probability

The *conditional probability* of event A given that event B has occurred is written $\mathbb{P}(A \mid B)$ and defined as

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

assuming $\mathbb{P}(B) > 0$.

5.1.2 Chain Rule

Another very useful tool, the *chain rule*, follows immediately from this definition:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A \mid B)\mathbb{P}(B) = \mathbb{P}(B \mid A)\mathbb{P}(A)$$

5.1.3 Bayes' Rule

Taking the equality from above one step further, we arrive at the simple but crucial *Bayes' rule*:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\mathbb{P}(A)}{\mathbb{P}(B)}$$

It is sometimes beneficial to omit the normalizing constant and write

$$\mathbb{P}(A \mid B) \propto \mathbb{P}(A)\mathbb{P}(B \mid A)$$

Under this formulation, $\mathbb{P}(A)$ is often referred to as the *prior* and $\mathbb{P}(B \mid A)$ as the *likelihood*.

In the context of machine learning, we can use Bayes' rule to update our "beliefs" (e.g. values of our model parameters) given some data that we've observed.

5.2 Random Variables

A *random variable* (r.v.) is some uncertain quantity with an associated probability distribution over the values it can assume. We write $X \sim p(\cdot)$ to indicate that the random variable X is distributed according to some probability mass/density function p (more on these functions below).

Formally, a random variable is a function from the sample space Ω to some other set of values, which we denote $X(\Omega) = \{X(\omega) \mid \omega \in \Omega\}$. To give a concrete example (taken from Pitman), suppose X is the number of heads in two tosses of a fair coin. The sample space is

$$\Omega = \{hh, tt, ht, th\}$$

and X is determined completely by the outcome ω, i.e. $X = X(\omega)$. Moreover, we see the fundamental way that a random variable relates back to its underlying sample space:

$$\mathbb{P}(X = x) = \mathbb{P}(\{\omega \in \Omega \mid X(\omega) = x\})$$

Typically we work only with random variables and don't concern ourselves with the original sample space Ω, but it's worth knowing that it really is there behind the scenes.

A word on notation: we write $p(X)$ to denote the entire probability distribution of X and $p(x)$ for the evaluation of the function p at a particular value $x \in X(\Omega)$. Hopefully this (reasonably standard) abuse of notation is not too distracting. If p is parameterized by some parameters θ, we write $p(X \mid \theta)$ or $p(x \mid \theta)$.

5.2.1 Discrete Random Variables

Discrete r.v.'s are usually specified by a nonnegative *probability mass function* (p.m.f.) p which satisfies

$$\sum_{x \in X(\Omega)} p(x) = 1$$

For a discrete X, the probability of a particular value is given exactly by its p.m.f.:

$$\mathbb{P}(X = x) = p(x)$$

5.2.2 Continuous Random Variables

Continuous r.v.'s are usually specified by a nonnegative *probability density function* (p.d.f.) p which satisfies

$$\int_{X(\Omega)} p(x)\,\mathrm{d}x = 1$$

For a continuous X, the probability of a particular value is zero ($\forall x \in X(\Omega)$, $\mathbb{P}(X = x) = 0$), so to get a positive probability we must integrate the p.d.f. over some range of values:

$$\mathbb{P}(a \leq X \leq b) = \int_a^b p(x)\,\mathrm{d}x$$

5.2.3 The Cumulative Distribution Function

The *cumulative distribution function* (c.d.f.) gives the probability that a random variable is at most a certain value:

$$F_X(x) = \mathbb{P}(X \leq x)$$

The c.d.f. can be used to give the probability that a variable lies within a certain range:

$$\mathbb{P}(a < X \leq b) = F_X(b) - F_X(a)$$

5.3 Joint Distributions

Often we have several random variables and we would like to get a distribution over some combination of them. A *joint distribution* is exactly this. For some r.v.'s $X_1, \ldots, X_n$, the joint distribution is written $p(X_1, \ldots, X_n)$ and gives probabilities over entire assignments to all the X_i simultaneously.

We say that two variables X and Y are *independent* if their joint distribution factors into their respective distributions, i.e.

$$p(X, Y) = p(X)p(Y)$$

It is often convenient (though perhaps questionable) to assume that a bunch of random variables are *independent and identically distributed* (i.i.d.) so that their joint distribution can be factored entirely:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i)$$

where $X_1, \ldots, X_n$ all share the same p.m.f./p.d.f.

5.4 Great Expectations

If we have some random variable X, we might be interested in knowing what is the "average" value of X. This concept is captured by the *expected value* (or *mean*) $\mathbb{E}[X]$, which is defined as

$$\mathbb{E}[X] = \sum_{x \in X(\Omega)} xp(x)$$

for discrete X and as

$$\mathbb{E}[X] = \int_{X(\Omega)} xp(x)\,\mathrm{d}x$$

for continuous X.

In words, we are taking a weighted sum of the values that X can take on, where the weights are the probabilities of those respective values. The expected value has a physical interpretation as the "center of mass" of the distribution.

5.4.1 Properties of Expected Value

A very useful property of expectation is that of linearity:

$$\mathbb{E}\left[\sum_{i=1}^{n} \alpha_i X_i + \beta\right] = \sum_{i=1}^{n} \alpha_i \mathbb{E}[X_i] + \beta$$

Note that this holds even if the X_i are not independent!

But if they are independent, the product rule also holds:

$$\mathbb{E}\left[\prod_{i=1}^{n} X_i\right] = \prod_{i=1}^{n} \mathbb{E}[X_i]$$

5.5 Variance

Expectation provides a measure of the "center" of a distribution, but frequently we are also interested in what the "spread" is about that center. We define the variance $\text{Var}(X)$ of a random variable X by

$$\text{Var}(X) = \mathbb{E}\left[\big(X - \mathbb{E}[X]\big)^2\right]$$

In words, this is the average squared deviation of the values of X from the mean of X. Using a little algebra and the linearity of expectation, it is straightforward to show that

$$\text{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

5.5.1 Properties of Variance

Variance is not linear (because of the squaring in the definition), but one can show the following:

$$\text{Var}(\alpha X + \beta) = \alpha^2 \,\text{Var}(X)$$

Basically, multiplicative constants become squared when they are pulled out, and additive constants disappear (since the variance contributed by a constant is zero).

Furthermore, if $X_1, \ldots, X_n$ are uncorrelated[2], then

$$\text{Var}(X_1 + \cdots + X_n) = \text{Var}(X_1) + \cdots + \text{Var}(X_n)$$

5.5.2 Standard Deviation

Variance is a useful notion, but our definition suffers from that fact the units of variance are not the same as the units of the random variable (again because of the squaring). To overcome this problem we can use *standard deviation*, which is defined as $\sqrt{\text{Var}(X)}$. The standard deviation has the same units as X.

5.6 Covariance

Covariance is a measure of the linear relationship between two random variables. We denote the covariance between X and Y as $\text{Cov}(X, Y)$, and it is defined to be

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Again, the linearity of expectation allows us to rewrite this as

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

Comparing these formulas to the ones for variance, it is not hard to see that $\text{Var}(X) = \text{Cov}(X, X)$.

5.6.1 Correlation

Normalizing the covariance gives the *correlation*:

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\,\text{Var}(Y)}}$$

[2]We haven't defined this yet; see the Correlation section below

Correlation also measures the linear relationship between two variables, but unlike covariance always lies between -1 and 1.

Two variables are said to be *uncorrelated* if $\text{Cov}(X, Y) = 0$ because $\text{Cov}(X, Y) = 0$ implies that $\rho(X, Y) = 0$. If two variables are independent, then they are uncorrelated, but the converse does not hold in general.

5.7 Random Vectors

So far we have been talking about *univariate distributions*, that is, distributions of single variables. But we can also talk about *multivariate distributions* which give distributions of *random vectors*:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

The metrics we have discussed for single variables have natural generalizations to the multivariate case.

Expectation of a random vector is simply the expectation applied to each component:

$$\mathbb{E}[\mathbf{X}] = \begin{bmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_n] \end{bmatrix}$$

The variance is generalized by the *covariance matrix*:

$$\Sigma = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^{\mathrm{T}}] = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \dots & \text{Cov}(X_1, X_n) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \dots & \text{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \text{Cov}(X_n, X_2) & \dots & \text{Var}(X_n) \end{bmatrix}$$

We see $\Sigma_{ij} = \text{Cov}(X_i, X_j)$. Since covariance is symmetric in its arguments, the covariance matrix is also symmetric. (It's also positive semi-definite, though we won't prove this.)

The inverse of the covariance matrix, Σ^{-1}, is sometimes called the *precision matrix*.

5.8 Estimation of Parameters

Now we get into some basic topics from statistics. We make some assumptions about our problem by prescribing a *parametric* model (e.g. a distribution that describes how the data were generated), then we fit the parameters of the model to the data. How do we choose the values of the parameters?

5.8.1 Maximum Likelihood Estimation

A common way to fit parameters is *maximum likelihood estimation* (MLE). The basic principle of MLE is to choose values that "explain" the data best by maximizing the probability of the data we've seen, conditioned on the parameters. Suppose we have random variables $X_1, \dots, X_n$ and corresponding observations $x_1, \dots, x_n$. Then

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \mathcal{L}(\theta)$$

where $\mathcal{L}$ is the *likelihood function*

$$\mathcal{L}(\theta) = p(x_1, \ldots, x_n \mid \theta)$$

Often, we assume that $X_1, \ldots, X_n$ are i.i.d. Then we can write

$$p(x_1, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta)$$

As this point, it is usually convenient to take logs, giving rise to the *log-likelihood*

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

This is a valid operation because the probabilities/densities are assumed to be positive, and since log is a monotonically increasing function, it preserves ordering. In other words, any maximizer of $\log \mathcal{L}$ will also maximize $\mathcal{L}$.

For some distributions, it is possible to analytically solve for the maximum likelihood estimator. If $\log \mathcal{L}$ is differentiable, setting the derivatives to zero and trying to solve for θ is a good place to start.

5.8.2 Maximum a Posteriori Estimation

A more Bayesian way to fit parameters is through *maximum a posteriori estimation* (MAP). In this technique we assume that the parameters are a random variable, and we specify a prior distribution $p(\theta)$. Then we can employ Bayes' rule to compute the posterior distribution of the parameters given the observed data:

$$p(\theta \mid x_1, \ldots, x_n) \propto p(\theta) p(x_1, \ldots, x_n \mid \theta)$$

Computing the normalizing constant is often intractable, because it involves integrating over the parameter space, which may be very high-dimensional. Fortunately, if we just want the MAP estimate, we don't care about the normalizing constant! It does not affect which values of θ maximize the posterior. So we have

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\theta) p(x_1, \ldots, x_n \mid \theta)$$

A particularly nice case is when the prior is chosen carefully such that the posterior comes from the same family as the prior. In this case the prior is called a *conjugate prior*. For example, if the likelihood is binomial and the prior is beta, the posterior is also beta. There are many conjugate priors; the reader may find this table of conjugate priors useful.