B14 Inference

Engineering Department
University of Oxford, UK

January 28, 2017

Contents

1 Introduction

These four lectures build on both the basic probability and statistics materials covered in A1 and the estimation concepts introduced in the estimation portion of B14. In particular these lectures re-introduce frequentist inference, in A1 called hypothesis testing, ground it in a slightly more theoretical framework, and explain it in terms of model-based reasoning. We continue by re-introducing maximum likelihood parameter estimation and regression, noting that parameter estimation can be used to perform inference. Following this we introduce regularization then Bayesian inference, the latter of which is contrasted to frequentist inference. We finish by introducing classification as a specialization of regression and discuss latent variable inference in this context.

The contents of these lectures are covered in many textbooks, notably the estimation and frequentist inference parts in

1. Bickel and Doksum (2015)
2. Johnson (2000)

and Bayesian inference in

1. Bishop (2006)
2. Gelman et al. (2014).

We will make make use of vector calculus in some of our derivations. The essential resource for making sense of these operations is the "Matrix Cookbook" of Petersen et al. (2008).

2 Models

A model is a simulacrum, often taking mathematical form, that stands in for some real-world system of interest. Models are used both for understanding systems and making predictions about how those systems might evolve – these two use case can be made mathematically equivalent. Statistical models are mathematical models that explicitly embrace, utilize, and compute with uncertainty. Inference has many technical definitions – these lectures will introduce two – but at a high level it can be thought of as using a model to say something about the real world.

Mathematically, by statistical model on a sample space X, we mean a set of distributions (actually measures) on X. If we write $PM(x)$ for the space of all possible distributions (measures) over X then a model is a subset $M \subset PM(X)$. The elements of M are indexed by a parameter θ with values in a parameter space T, that is

$$M = \{P_\theta | \theta \in T\}$$

where each P_θ is a member of the set $PM(x)$. A model is parametric if T is finite dimensional. Usually $T \subset \mathbb{R}^d$ for some $d \in \mathbb{N}$. If $dim(T) = \infty$ then M is a non-parametric model.

A canonical problem of statistics is to take observations $\{x_1, \ldots, x_N\}, x_n \in X$, which we model as random variables $\{X_1, \ldots, X_N\}$. which we assume are drawn from P_θ, i.e.

$$\{X_1, \ldots, X_N\} \underset{iid}{\sim} P_\theta \qquad \theta \in T$$

and use them to tell us something about the value of θ.

Inference is typically phrased in terms of mathematical operations involving a model. There are *frequentist* and *Bayesian* schools of inference and both involve estimation and interpretation of model parameters. We will start with estimators and frequentist notions of inference.

3 Estimators

Assume there is a sample $\{x_1, x_2 \ldots, x_n\}$ of n *iid* (independent and identically distributed) observations coming from a *true* but unknown distribution $P_0(\cdot)$. Let us assume the $P_0 \in M$, i.e. $\exists \theta_0 \in T$ such that $P_{\theta_0} = P_0$. Let us also assume that T is finite dimensional.

We would like an *estimator* f which produces an *estimate* $\hat{\theta}$ that is close to θ_0 (the *estimand*), i.e. $\hat{\theta} = f(x_1, \ldots, x_n) \approx \theta_0$. An estimator is a statistic, both are defined simply as functions of the sample or *data*. Some estimators can be quite complex, some quite simple.

In order to motivate estimators and their upcoming utility in frequentist inference, consider the following general idea. Let's say that we want to learn something about the world. As a first step we need to posit a model of the world and of how what we are able to measure relates to how the world is. Because the world is stochastic and measurements uncertain within tolerances this model will typically be a model that embraces randomness, i.e. a statistical model. Let's say this model has one parameter (potentially a high-dimensional structured parameter) and that given a value for this parameter a population is generated. Let's also assume that this population is large (manufactured parts, running engines in cars, heights of people, etc.) and that for practical reasons we cannot measure or observe all the individuals in the population, rather, we can only take measurements from a sub-population or sample. Now let's say that we have a function that maps from the measurements of a subpopulation (our data) to an estimate of some model parameter in which we're interested. If we took all possible random subpopulations and ran this estimator there would be a distribution of estimates that came out. Frequentist inference basically arises from using the model to establish a theoretical sampling distribution of the estimator and then checks whether or not the estimate arrived at by running the same estimator on the one actual, real, collected dataset is surprising or not. If it's surprising then that lends evidence to your model being an inaccurate explanation of how the world is. If it's not surprising then it's taken to mean that you cannot refute the given hypothesis (which came in the form of the model). Note that this does not mean that there isn't a very large number of hypotheses that also cannot be refuted, simply, that the single given hypothesis might not be a good hypothesis.

To be more mathematically concrete, consider drawing from some large, possibly infinite population a finite sub-population (called a *sample*) of size N. Call this sample X_1. Then consider evaluating an estimator on this population to produce an estimate $\hat{\theta}_1$. Repeat this procedure M times to produce M estimates. Clearly as each sample is different so too will be the M different estimates, as illustrated in the following table

Datasets/samples		Estimates
$X_1 = \{x_{11} = \dots, x_{12} = \dots,$	$x_{1N} = \dots\}$	$\hat{\theta}_1$
$X_2 = \{x_{21} = \dots, x_{22} = \dots,$	$x_{2N} = \dots\}$	$\hat{\theta}_2$
.		.
.		.
$X_M = \{x_{M1} = \dots, x_{M2} = \dots,$	$x_{MN} = \dots\}$	$\hat{\theta}_M$

where $x_{mn} \underset{iid}{\sim} P_0$, $P_0(X) = \prod P_0(x)$. Further, one could calculate both the sample mean and variance of these M estimates as shown in the Figure 1.

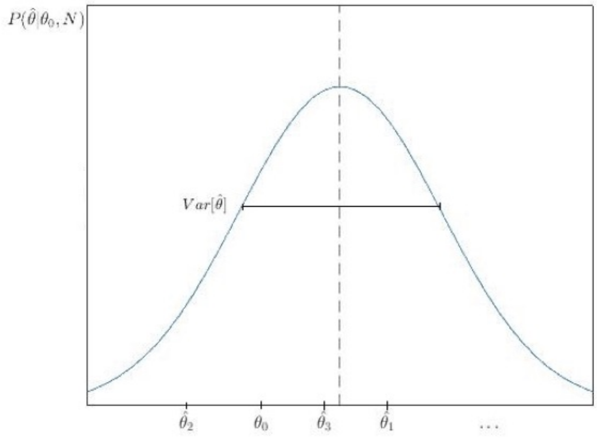


Figure 1: Graphical representation of $\mathrm{Var}(\hat{\theta})$

Such a procedure is not practical in any real science or engineering application because there is always a cost associated with making M measurements, so, instead, in practice, one replaces the original large population with a hypothesis about its distribution, i.e. assumes a statistical model of this population and then runs such a procedure. In such a case there is no reason to stop with M estimates. Instead the sampling distribution of an estimator under a model then can either analytically derived or nearly exhaustively computed even when $M \to \infty$.

3.1 Properties of estimators

Estimators have mathematical properties. A familiarity with mean squared error, variance, bias, and the relationship between the three is essential for understanding decisions we will make later when constructing estimators.

Let the data $X = \{x_1, \dots, x_n\}$ and the estimator $\hat{\theta} = f(X)$ with $\hat{\theta} \in \mathbb{R}$; i.e. the estimate is assumed to one-dimensional. Continue assuming that the true data distribution is P_0 and there is a parameter θ_0 such that P_{θ_0} can equal P_0.

1. Mean Squared Error:

$$MSE(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta_0)^2\right]$$

   note that this expectation is with respect to samples (populations of some size) drawn from P_0. This is the mean or expected squared error; small if the estimator is good.

2. Variance:

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

   this is, for a given sample size (and P_0), what is the characteristic “spread” of the estimate?

3. Bias:

$$B(\hat{\theta}) = \mathbb{E}(\hat{\theta}) - \theta_0$$

is the distance between the average of estimates over all samples of a given size and θ_0. Statisticians like unbiased estimators, i.e. $B(\hat{\theta}) = 0$.

Theorem 3.1.

$$MSE(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + (B(\hat{\theta}))^2$$

Proof.

$$\begin{aligned}
MSE(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \theta_0)^2] \\
&= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta_0)^2] \\
&= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] \\
&+ \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])(\mathbb{E}[\hat{\theta}] - \theta_0)] \qquad (1) \\
&+ \mathbb{E}[(\mathbb{E}[\hat{\theta}] - \theta_0))(\hat{\theta} - \mathbb{E}[\hat{\theta}])] \qquad (2) \\
&+ \mathbb{E}[\mathbb{E}[\hat{\theta}] - \theta_0)^2]
\end{aligned}$$

Noting that $\mathbb{E}[\hat{\theta}], \theta_0$ are constant $\therefore$ $(1) = (2) = (\mathbb{E}[\hat{\theta}] - \mathbb{E}[\hat{\theta}])(\mathbb{E}[\hat{\theta}] - \theta_0) = 0$

$$\begin{aligned}
\therefore MSE(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] + (\mathbb{E}[\hat{\theta}] - \theta_0)^2 \\
&= \mathrm{Var}(\hat{\theta}) + (B(\hat{\theta}))^2
\end{aligned}$$

□

We will return to this later.

3.2 Example estimators

Let's look at three simple estimators of a population mean. Assume the true distribution (our modeling assumption) is $\mathcal{N}(\mu_0, \sigma_0^2)$. Let N be the sample size and $X = \{x_1, x_2 \ldots, x_n\}, x_n \underset{iid}{\sim} \mathcal{N}(\mu_0, \sigma_0^2)$.

Let's consider some estimators:

$$\begin{aligned}
&\mathbf{A}) \qquad \hat{\mu}_A = f_A(X) = f_A(x_1, x_2 \ldots, x_n) = \mu_g \\
&\mathbf{B}) \qquad \hat{\mu}_B = f_B(X) = \arg\max_{\mu} \mathcal{L}(X; \mu) \\
&\mathbf{C}) \qquad \hat{\mu}_C = f_c(X) = \lambda\mu_g + (1 - \lambda)\hat{\mu}_B.
\end{aligned}$$

Note that $\hat{\mu}_B$ is the maximum likelihood estimator for μ, μ_g is some "guess" $\mu_g \approx \mu_0$, and λ is a parameter related to how much we "believe" our guess.

Estimator A

$$\begin{aligned}
Var(\hat{\mu}_A) &= 0 \\
Bias(\hat{\mu}_A) &= \hat{\mu}_A \\
MSE(\hat{\mu}_A) &= \hat{\mu}_A^2
\end{aligned}$$

Estimator B

This is the maximum likelihood estimator where we have a parametric model

$$X \underset{iid}{\sim} \mathcal{N}(\mu_0, \sigma_0^2), \quad \theta = \{\mu_0, \sigma_0^2\}$$

and

$$\begin{aligned}\mathcal{L}(X;\theta) &= \prod_{n=1}^{N} p(x_n|\theta) \\ &= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{\frac{(x_n-\mu_0)^2}{2\sigma_0^2}}\end{aligned}$$

By way of reminder the figure below illustrates the maximum likelihood principle whereby parameter values that give data high likelihood under a model are preferred.

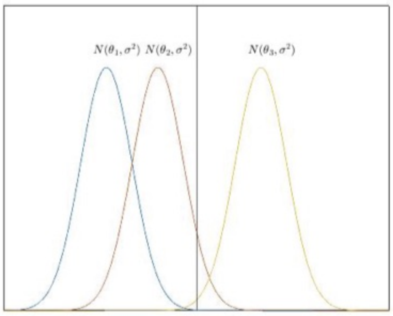


Figure 2: Which θ is preferred?

Review: Maximum Log-Likelihood *Estimation*

If we want to maximise $\mathcal{L}(X;\theta)$ w.r.t. θ we look for the highest value of $\mathcal{L}(X;\theta)$ as a function of θ where

$$\frac{\partial \mathcal{L}(X;\theta)}{\partial \theta} = 0 \quad \text{and} \quad \frac{\partial^2 \mathcal{L}(X;\theta)}{\partial \theta^2} \leq 0$$

Important $\frac{\partial \mathcal{L}(X;\theta)}{\partial \theta}$ is almost always nasty so it is usual to work with $\frac{\partial \log \mathcal{L}(X;\theta)}{\partial \theta}$ which has the same max because log is monotonically increasing. Often the resulting maximisation is easy.

NB: for simplicity we let $\mathcal{L}(\theta) = \mathcal{L}(X;\theta)$.

Theorem 3.2.

$$\arg\max_{\theta} \log(\mathcal{L}(\theta)) = \arg\max_{\theta}(\mathcal{L}(\theta))$$

Proof. Check extrema:

$$\begin{aligned}0 &= \frac{\partial \log \mathcal{L}(X;\theta)}{\partial \theta} \\ &= \frac{1}{\mathcal{L}(\theta)} \frac{\partial \mathcal{L}(\theta)}{\partial \theta} \\ &= \frac{\partial \mathcal{L}(\theta)}{\partial \theta} \\ \therefore \frac{\partial \log \mathcal{L}(X;\theta)}{\partial \theta} = 0 &\iff \frac{\partial \mathcal{L}(\theta)}{\partial \theta} = 0\end{aligned}$$

Check curvature

$$\begin{aligned}sign\left(\frac{\partial^2}{\partial \theta^2} \log(\mathcal{L}(\theta))\right) &= sign\left(\frac{\partial}{\partial \theta} \frac{1}{\mathcal{L}(\theta)} \frac{\partial \mathcal{L}(\theta)}{\partial \theta}\right) \\ &= sign\left(-\frac{1}{\mathcal{L}(\theta)^2} \frac{\partial \mathcal{L}(\theta)}{\partial \theta} + \frac{1}{\mathcal{L}(\theta)} \frac{\partial^2 \mathcal{L}(\theta)}{\partial \theta^2}\right) \\ &= sign\left(0 + \frac{1}{\mathcal{L}(\theta)} \frac{\partial^2 \mathcal{L}(\theta)}{\partial \theta^2}\right)\end{aligned}$$

$\therefore$ True as $\mathcal{L}(\theta)$ is a likelihood thus always positive and log is monotonic. □

Back to estimator B (maximum likelihood)

$$\arg\max_{\mu_0} \log \mathcal{L}(X;\mu_0) \quad \text{occurs at} \quad \frac{\partial}{\partial \mu_0} \log \mathcal{L}(X;\mu_0) = 0$$

$$\begin{aligned}
\text{i.e.} \quad \frac{\partial}{\partial \mu_0} \log \mathcal{L}(X;\mu_0) &= \sum_n \frac{\partial}{\partial \mu_0} \log \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{-\frac{(x_n-\mu_0)^2}{2\sigma_0^2}} \\
&= \sum_n \frac{\partial}{\partial \mu_0} \Big(const - \frac{(x_n-\mu_0)^2}{2\sigma_0^2} \Big) \\
&= \sum_n \frac{2(x_n-\mu_0)}{2\sigma_0^2} \\
&= \frac{(\sum_{n=1}^N x_n - N\mu_0)}{\sigma_0^2} = 0
\end{aligned}$$

$$\implies \hat{\mu}_B = \frac{\sum_{n=1}^N x_n}{N}$$

Bias of $\hat{\mu}_B$

$$\begin{aligned}
\mathbb{E}[\hat{\mu}_B] = \mathbb{E}\left[\frac{\sum_{n=1}^N x_n}{N}\right] &= \frac{1}{N}\sum_{n=1}^N \mathbb{E}[x_n] \\
&= \frac{1}{N} \times N \times \mu_0 = \mu_0
\end{aligned}$$

i.e. $\hat{\mu}_B$ is an *unbiased* estimator and

$$\therefore B(\hat{\mu}_B) = 0.$$

Variance of $\hat{\mu}_B$

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_B) &= \mathrm{Var}\left(\frac{\sum_{n=1}^N x_n}{N}\right) \\
&= \frac{1}{N^2}\mathrm{Var}\Big(\sum_{n=1}^N x_n\Big) \\
&= \frac{1}{N^2}\sum_{n=1}^N \mathrm{Var}(x_n) \qquad iid \\
&= \frac{\sigma_0^2}{N}
\end{aligned}$$

MSE of $\hat{\mu}_B$

$$MSE(\hat{\mu}_B) = \mathrm{Var}(\hat{\mu}_B) - B(\hat{\mu}_B) = \frac{\sigma_0^2}{N} - 0 = \frac{\sigma_0^2}{N}.$$

Note that in the limit

$$\lim_{N\to\infty} MSE(\hat{\mu}_B) = \lim_{N\to\infty} \mathrm{Var}(\hat{\mu}_B) = 0.$$

4 Inference via Hypothesis Testing

Estimation is the core of stats and goes hand in had with frequentist inference. The frequentist inference procedure is hypothesis testing, the procedure for drawing inferences by not finding surprise in estimates already outlined in Section 3.

Assuming that we know σ_0^2 let's hypothesize $H_0 : x_n \underset{iid}{\sim} \mathcal{N}(\phi, \sigma_0^2)$. Usually symbolized by H_0, this "null hypothesis" is effectively a model-based statement about the world. Note that this, here, is a simple parametric model of a population. To test this hypothesis we conduct the following thought experiment. Assume that we use estimator $\mathbf{B}$ and that we draw an infinite number of sample populations of size N from the model above. We already know that, in this case

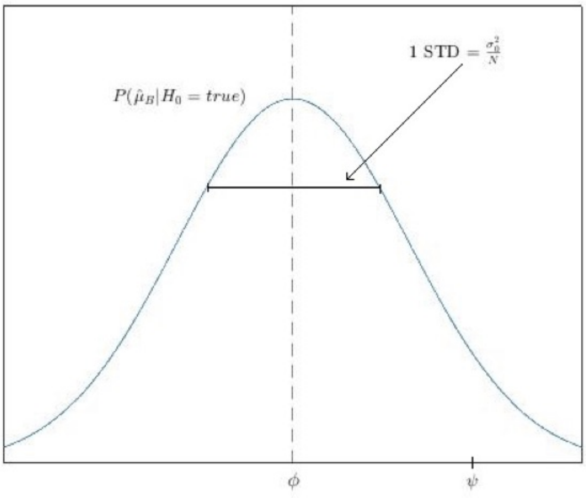
$$\mathbb{E}[\hat{\mu}_B] = \phi \quad \text{and} \quad \mathrm{Var}(\hat{\mu}_B) = \frac{\sigma_0^2}{N}$$

What can be demonstrated in this case exactly is that

$$\hat{\mu}_B \sim \mathcal{N}\left(\phi, \frac{\sigma_0^2}{N}\right).$$

The central limit theorem can be used to show that estimators that involve an average, for sufficiently large N and under mild regularity assumptions, having sampling distributions that are also are well described by a normal distribution asymptotically too.

Here though, given a particular dataset X_D we compute $\hat{\mu}_B(X_D) = \psi$. We have a picture like this



Where ψ is the result of applying estimator **B** to one dataset. To think about confidence intervals and drawing conclusions about a null hypothesis (H_0) it will help, in this and many cases to "normalise" our estimator to a z-score, i.e. a $\mathcal{N}(0, 1)$ variable.

If $\hat{\mu}_B \sim \mathcal{N}(\phi, \frac{\sigma_0^2}{N})$ then

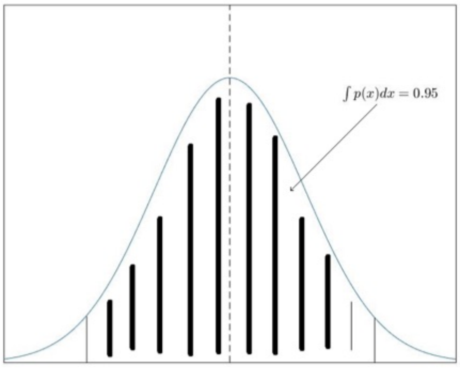
$$\frac{(\hat{\mu}_B - \phi)}{\sigma \backslash \sqrt{N}} \sim \mathcal{N}(0, 1) \qquad \text{Proof trivial}$$

We thus map from the ψ domain to the z domain i.e.

$$z(\psi) = \frac{(\psi - \phi)}{\sigma \backslash \sqrt{N}}$$

Frequentist inference rejects a null hypothesis when the value of an estimator computed on a real sample is surprising (i.e. low probability) under the null hypothesis. If, for instance, $z(\psi)$ falls within the 95% confidence interval then we fail to reject H_0. What is the 95% confidence interval?

It is, under the standard normal



the value z for which $p(-z \leq Z \leq z) = 1 - \alpha = 0.95$ for $Z \sim \mathcal{N}(0,1)$. This occurs at $\Phi(z) = P(Z \leq z) = 1 - \frac{\alpha}{2} = 0.975$, where $\Phi(z)$ is the CDF of $\mathcal{N}(0,1)$.

$$\Phi(z) \triangleq \int_{-\infty}^{z} \mathcal{N}(x;0,1)dx$$

i.e. $\Phi^{-1}\big(\Phi(z) = 0.975\big) = 1.96$ so

$$0.95 = 1 - \alpha = P(-z \leq Z \leq z) = P(-1.96 \leq \frac{\psi - \phi}{\sigma\backslash\sqrt{N}} \leq 1.96)$$

and, thusly

$$0.95 = P\big(\psi - 1.96(\frac{\sigma}{\sqrt{N}}) \leq \phi \leq \psi + 1.96(\frac{\sigma}{\sqrt{N}}))\big)$$

What does this mean? It means that, under our assumptions, both ϕ is likely to be in this band and, perhaps more importantly, that ϕ and ψ can be flipped, saying in effect that if the true parameter is ϕ, ψ outside of

$$\big(\phi - 1.96(\frac{\sigma}{\sqrt{N}})\ ,\ \phi + 1.96(\frac{\sigma}{\sqrt{N}})\big)$$

would be surprising and could be taken as evidence (at 5% confidence) to reject H_0.

Note that a p-value is the value of α one would have to choose to reject the null hypothesis given the observed sample, i.e. p-value is given by (in this case)

$$\Phi^{-1}(1 - \frac{\alpha}{2}) = \frac{\psi - \phi}{\sigma\backslash\sqrt{N}}$$

or

$$1 - \frac{\alpha}{2} = \Phi(\frac{\psi - \phi}{\sigma\backslash\sqrt{N}})$$

$$\text{p-value} = \alpha = 2\Big(1 - \Phi(\frac{\psi - \phi}{\sigma\backslash\sqrt{N}})\Big)$$

Back to our Estimators

Rewinding all the way back we have

	A	B	C
Var	0	$\frac{\sigma_0^2}{N}$	
Bias	$\hat{\mu}_A$	0	
MSE	$\hat{\mu}_A^2$	$\frac{\sigma_0^2}{N}$	

Which suggests that a) no one estimator is always best and b) that we might be able to explore the spectrum of estimators and engineer estimators that work well for our problem. For instance, what if we think we know the true "answer," i.e. $\mu_g \approx \mu_0$, like estimator C.

Estimator C

$$\hat{\mu}_C = \lambda\mu_g + (1-\lambda)\hat{\mu}_B$$

Bias of $\hat{\mu}_C$

$$\begin{aligned}
B(\hat{\mu}_C) &= B(\lambda\mu_g + (1-\lambda)\hat{\mu}_B) \\
&= \mathbb{E}[\lambda\mu_g + (1-\lambda)\hat{\mu}_B)] - \mu_0 \\
&= \lambda\mu_g + (1-\lambda)\mu_0 - \mu_0 = \lambda(\mu_g - \mu_0)
\end{aligned}$$

Variance of $\hat{\mu}_C$

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_C) &= \mathrm{Var}(\lambda\mu_g + (1-\lambda)\hat{\mu}_B) \\
&= (1-\lambda)^2 \,\mathrm{Var}(\hat{\mu}_B) = (1-\lambda)^2 \frac{\sigma^2}{N}
\end{aligned}$$

All Together

	A	B	C
Var	0	$\frac{\sigma_0^2}{N}$	$(1-\lambda)^2\frac{\sigma_0^2}{N}$
Bias	$\hat{\mu}_A$	0	$\lambda(\mu_g-\mu_0)$
MSE	$\hat{\mu}_A^2$	$\frac{\sigma_0^2}{N}$	$(1-\lambda)^2\frac{\sigma_0^2}{N} + \lambda^2(\mu_g-\mu_0)^2$

But, wait, this suggests that bias could lead to lower variance (when $\frac{\sigma_0^2}{N} > (1-\lambda)^2\frac{\sigma_0^2}{N}$) and MSE when

$$\begin{aligned}
\frac{\sigma_0^2}{N} &> (1-\lambda)^2\frac{\sigma_0^2}{N} + \lambda^2(\mu_g-\mu_0)^2 \\
\frac{\sigma_0^2}{N}(1-(1-\lambda)^2) &> \lambda^2(\mu_g-\mu_0)^2 \\
\frac{\sigma_0^2}{N} &> \frac{\lambda^2}{2\lambda-\lambda^2}(\mu_g-\mu_0)^2 \\
\frac{\sigma_0^2}{N} &> \frac{\lambda}{2-\lambda}(\mu_g-\mu_0)^2
\end{aligned}$$

which could easily happen.

The take-home here is that maximum likelihood estimators are but one family of estimator, frequentist inference is one way to use statistical models, and, further, that bias is not a bad word. In fact introducing bias, as we will see, can lower the variance of your estimator and potentially simultaneously the MSE of your estimator as well. Frequentist inference involves computing and using confidence intervals for the sample variance of estimators where it is clear that having a lower variance of your estimator will increase the power of your estimator in terms of being able to discriminate between valid and invalid hypotheses. Bias introduced in the form of regularisation and priors may make for better estimation and inference provided that the introduced bias is helpful in terms of reducing variance.

5 Parameter Estimation

While true statistical inference involves either hypothesis testing under the frequentist framework or characterization of a posterior distribution under the Bayesian framework, a more engineering notion of inference is simply that of parameter estimation. In order to illustrate parameter estimation in meaningful and useful setting we must further ensure an absolutely stable platform of understanding of the multivariate Gaussian distribution, continuing and building upon B14 Estimation towards, again, linear regression then regularization and Bayesian reasoning.

5.1 The Multivariate Gaussian

Let $\boldsymbol{\mu}, \boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{\Sigma} \in PSD^{DxD}$ then let

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \mathrm{e}^{\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}}$$

be the "multivariate normal density function" of random variable $\boldsymbol{x}$ in D dimensions given mean vector $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}]$ and covariance matrix $\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{x})$. NB: we sometimes also work with the precision matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$.

An important thing to know about MVN distributed vectors is that $\boldsymbol{x}$ has the same distribution as $\boldsymbol{Az} + \boldsymbol{\mu}$ where

$$\boldsymbol{z} = [z_1, \dots, z_D], \quad z_d \underset{iid}{\sim} \mathcal{N}(0,1)$$

and $\boldsymbol{A}$ satisfies $\boldsymbol{AA}^T = \boldsymbol{\Sigma}$. (Cholesky)

Certainly we can note that

$$\mathbb{E}[\boldsymbol{Az} + \boldsymbol{\mu}] = \boldsymbol{A}\,\mathbb{E}[\boldsymbol{z}] + \boldsymbol{\mu} = \boldsymbol{A} \times \boldsymbol{0} + \boldsymbol{\mu} = \boldsymbol{\mu}$$

and

$$\mathrm{Cov}[\boldsymbol{Az} + \boldsymbol{\mu}] = \mathrm{Cov}[\boldsymbol{Az}] = \boldsymbol{A}\,\mathrm{Cov}[\boldsymbol{z}]\boldsymbol{A}^T = \boldsymbol{AA}^T = \boldsymbol{\Sigma}$$

But higher-order moments could appear.

We can "derive" the MVN pdf starting from the product of the 1-D normal Gaussian pdfs for the individual components of the vector $\boldsymbol{z}$

$$\begin{aligned} P_{\boldsymbol{Z}}(\boldsymbol{z}) &= \prod_{i=1}^{D} (2\pi)^{1/2}\mathrm{e}^{\{-\frac{1}{2}z_i^2\}} \\ &= (2\pi)^{D/2}\mathrm{e}^{\{-\frac{1}{2}\sum_{i=1}^{D} z_i^2\}} \\ &= (2\pi)^{D/2}\mathrm{e}^{\{-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{z}\}} \end{aligned}$$

Let

$$\boldsymbol{x} = \boldsymbol{Az} + \boldsymbol{\mu} \implies \boldsymbol{z} = \boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$$

where $\boldsymbol{AA}^T = \boldsymbol{\Sigma}$. The multivariate change of variable rule says

$$P_{\boldsymbol{X}}(\boldsymbol{x}) = P_{\boldsymbol{Z}}(\boldsymbol{z}) \begin{vmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_D} \\ \frac{\partial z_2}{\partial x_1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial z_D}{\partial x_1} & \cdots & \cdots & \frac{\partial z_D}{\partial x_D} \end{vmatrix} = P_{\boldsymbol{Z}}(\boldsymbol{z})|\boldsymbol{J}_{\boldsymbol{z}\to\boldsymbol{x}}|$$

where

$$\boldsymbol{J}_{\boldsymbol{z}\to\boldsymbol{x}} \triangleq \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \frac{\partial(z_1, \dots, z_n)}{\partial(x_1, \dots, x_m)} \triangleq \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial x_1} & \cdots & \frac{\partial z_n}{\partial x_m} \end{bmatrix}$$

is a Jacobian and drops out from the multivariate chain rule.

Noting

$$\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}$$

$$\therefore \boldsymbol{A}^{-1} = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}$$

and

$$|\boldsymbol{A}^{-1}| = |\boldsymbol{A}|^{-1}$$

we have

$$\begin{aligned}
P_{\boldsymbol{X}}(\boldsymbol{x}) &= P_{\boldsymbol{Z}}(\boldsymbol{z})|\boldsymbol{A}^{-1}| \\
&= P_{\boldsymbol{Z}}(\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))|\boldsymbol{A}|^{-1} \\
&= (2\pi)^{-D/2}|\boldsymbol{A}|^{-1}\mathrm{e}^{\{-\frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})]^T[\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})]\}} \\
&= (2\pi)^{-D/2}|\boldsymbol{A}\boldsymbol{A}^T|^{-\frac{1}{2}}\mathrm{e}^{\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{A}^{-1})^T\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}} \\
&= (2\pi)^{-D/2}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\mathrm{e}^{\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}} \\
&= (2\pi)^{-D/2}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\mathrm{e}^{\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{\Sigma})^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}}
\end{aligned}$$

Where we have used

$$|\boldsymbol{A}|^{-1} = |\boldsymbol{A}|^{-\frac{1}{2}}|\boldsymbol{A}|^{-\frac{1}{2}} = |\boldsymbol{A}|^{-\frac{1}{2}}|\boldsymbol{A}^T|^{-\frac{1}{2}} = (|\boldsymbol{A}||\boldsymbol{A}^T|)^{-\frac{1}{2}}$$

and

$$(\boldsymbol{A}^{-1})^T\boldsymbol{A}^{-1} = (\boldsymbol{A}^T)^{-1}\boldsymbol{A}^{-1} = (\boldsymbol{A}\boldsymbol{A}^T)^{-1}$$

with individual steps from "the matrix cookbook" (Petersen et al., 2008), notably

$$\begin{aligned}
(\boldsymbol{A}\boldsymbol{B})^T &= \boldsymbol{B}^T\boldsymbol{A}^T \\
|\boldsymbol{A}| &= |\boldsymbol{A}^T| \\
(\boldsymbol{A}^{-1})^T &= (\boldsymbol{A}^T)^{-1} \\
(\boldsymbol{A}\boldsymbol{B})^{-1} &= \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \\
|\boldsymbol{A}\boldsymbol{B}| &= |\boldsymbol{A}||\boldsymbol{B}|
\end{aligned}$$

So we can represent a multivariate Gaussian as an affine transformation of a vector of individual $\mathcal{N}(0,1)$ distributed random variables. But, given a collection of N vectors $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ which we assume to be $\underset{iid}{\sim}$ $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ how do we get $\boldsymbol{\mu}$ & $\boldsymbol{\Sigma}$?

5.2 Maximum Likelihood Estimation for the Multivariate Gaussian

Let

$$\boldsymbol{X} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$$

Then

$$\mathcal{L}(\boldsymbol{X};\{\boldsymbol{\mu},\boldsymbol{\Sigma}\}) = \prod_{n=1}^{N}(2\pi)^{-D/2}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\mathrm{e}^{\{\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\}}$$

Like before we will estimate $\hat{\boldsymbol{\mu}}_{ML}$ and $\hat{\boldsymbol{\Sigma}}_{ML}$ by taking derivatives of log likelihoods and setting them equal to zero. We will need the following (also from the Matrix Cookbook) valid only for symmetric $\boldsymbol{W}$

$$\frac{\partial \log|\boldsymbol{X}|}{\partial \boldsymbol{X}} = (\boldsymbol{X}^{-1})^T = (\boldsymbol{X}^T)^{-1} \tag{3}$$

$$\frac{\partial \boldsymbol{a}^T\boldsymbol{X}\boldsymbol{b}}{\partial \boldsymbol{X}} = \boldsymbol{b}\boldsymbol{a}^T \tag{4}$$

$$\frac{\partial \boldsymbol{a}^T\boldsymbol{X}^{-1}\boldsymbol{b}}{\partial \boldsymbol{X}} = -\boldsymbol{X}^{-T}\boldsymbol{a}\boldsymbol{b}^T\boldsymbol{X}^{-T} \tag{5}$$

$$\frac{\partial}{\partial \boldsymbol{s}}(\boldsymbol{x}-\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}) \tag{6}$$

First ML estimate $\hat{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\mu}}\log\mathcal{L}(\boldsymbol{X};\{\boldsymbol{\mu},\boldsymbol{\Sigma}\}) &= \sum_{n=1}^{N}\frac{\partial}{\partial \boldsymbol{\mu}}\big(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\big) = 0 \\
\text{using identity } (4) &= \sum_{n-1}^{N}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}) = 0 \\
&= \sum_{n-1}^{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_n - N\boldsymbol{\mu} = 0
\end{aligned}$$

$$\implies \hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$$

Note:

$$\mathbb{E}[\hat{\boldsymbol{\mu}}] = \frac{1}{N}\sum_{n-1}^{N}\mathbb{E}[\boldsymbol{x}_n] = \frac{N}{N}\boldsymbol{\mu} \implies Bias(\hat{\boldsymbol{\mu}}) = 0$$

Now

$$\hat{\boldsymbol{\Sigma}}_{ML} = \arg\max_{\boldsymbol{\Sigma}} \log \mathcal{L}(\boldsymbol{X};\boldsymbol{\mu},\boldsymbol{\Sigma})$$

$$\begin{aligned}\therefore \frac{\partial}{\partial \boldsymbol{\Sigma}} \log \mathcal{L}(\boldsymbol{X};\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \frac{N}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}\frac{\partial}{\partial \boldsymbol{\Sigma}}\left((\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\right)\\ \text{using identity (3)} &= \frac{N}{2}(\boldsymbol{\Sigma}^{-1})^T + \frac{1}{2}\sum_{n=1}^{N}\left(\boldsymbol{\Sigma}^{-T}(\boldsymbol{x}_n-\boldsymbol{\mu})(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-T}\right)\end{aligned}$$

$\boldsymbol{\Sigma}$ is symmetric $\therefore$ $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$

$$\therefore (\boldsymbol{\Sigma}^{-1})^T = (\boldsymbol{\Sigma}^T)^{-1} = \boldsymbol{\Sigma}^{-1}$$

$\therefore$ Setting this expression equal to zero and solving yields

$$N\boldsymbol{\Sigma}^{-1} = \sum_{n=1}^{N}\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}\right)$$

Then pre- and post-multiplying both sides by $\boldsymbol{\Sigma}$ finally gives

$$\hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n-\hat{\boldsymbol{\mu}})(\boldsymbol{x}_n-\hat{\boldsymbol{\mu}})^T$$

So now, given data, we can analytically compute the ML parameters of a multivariate Gaussian.

A note: the ML estimator, particularly for exponential families, has nice properties, particularly in the limit. It can be shown, for instance, that the sampling distribution of most ML estimators tend asymptotically to a normal distribution. Extra reading: Fisher Information Matrix.

5.3 Example: Sensor Fusion as Parameter Estimation

Using these techniques we can consider using maximum likelihood techniques to estimate the location of an object. This example corresponds to an accompanying exercise in the B14 laboratory.

Assume two independent proximity sensors which are unbiased but produce "noisy" measurements of an object's location in 2-D space.

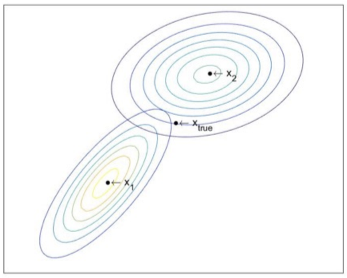


Figure 3: $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^2$ are the sensor measurements, $\boldsymbol{x}_{true} \in \mathbb{R}^2$ is the object's true location

From the sensor manufacturers we know the MVN observation noise variance for each sensor

$$\boldsymbol{x}_i - \boldsymbol{x}_{true} \sim \mathcal{N}(0,\boldsymbol{\Sigma}_i) \iff \boldsymbol{x}_{true} \sim \mathcal{N}(\boldsymbol{x}_i,\boldsymbol{\Sigma}_i) \iff \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{x}_{true},\boldsymbol{\Sigma}_i)$$

Let's say we believe both sensors equally and that they are independent. Then

$$\begin{aligned}P(\boldsymbol{x}_1,\boldsymbol{x}_2|\boldsymbol{x}_{true}) &= P(\boldsymbol{x}_1|\boldsymbol{x}_{true})P(\boldsymbol{x}_2|\boldsymbol{x}_{true})\\ &= \mathcal{N}(\boldsymbol{x}_1;\boldsymbol{x}_{true},\boldsymbol{\Sigma}_1)\,\mathcal{N}(\boldsymbol{x}_2;\boldsymbol{x}_{true},\boldsymbol{\Sigma}_2)\end{aligned}$$

The ML principle says that

$$\hat{\boldsymbol{x}}_{true} = \underset{\boldsymbol{x}_{true}}{\arg\max}\{\log \mathcal{N}(\boldsymbol{x}_1; \boldsymbol{x}_{true}, \boldsymbol{\Sigma}_1) + \log \mathcal{N}(\boldsymbol{x}_2; \boldsymbol{x}_{true}, \boldsymbol{\Sigma}_2)\}$$

Setting up as usual:

$$\frac{\partial}{\partial \boldsymbol{x}_{true}}\big((\boldsymbol{x}_1 - \boldsymbol{x}_{true})^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_{true}) + (\boldsymbol{x}_2 - \boldsymbol{x}_{true})^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}_2 - \boldsymbol{x}_{true})\big) = 0$$

$$\begin{aligned}\text{using identity (4)} &= -2\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_{true}) - 2\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}_2 - \boldsymbol{x}_{true}) = 0 \\ \therefore \boldsymbol{\Sigma}_1^{-1}\boldsymbol{x}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{x}_2 &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})\boldsymbol{x}_{true} \\ \therefore \hat{\boldsymbol{x}}_{true} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{x}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{x}_2)\end{aligned}$$

In multiple dimensions this is a little difficult to parse, but let's assume that the derivation holds in 1-D,then

$$\hat{x}_{true} = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}\Big(\frac{1}{\sigma_1^2}x_1 + \frac{1}{\sigma_2^2}x_2\Big)$$

which is just a weighted average with higher weight on the smallest variance.

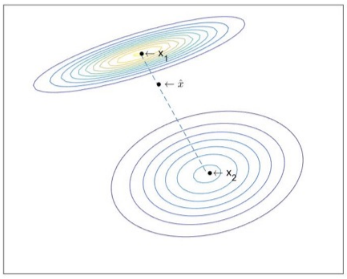


Figure 4: $\hat{\boldsymbol{x}}$ placed between the two sensors measurements but closer to the sensor with the lower covariance, which we "trust" more

6 Linear Regression

If at this point in your education you still think that linear regression is about drawing a line through points, think again! In this section we re-introduce linear regression as, effectively, parameter estimation in a multivariate Gaussian distribution. Unlike the sensor fusion parameter estimation example above, in linear regression we can think about posing a hypothesis in the form of a model and as a result we may think, again, about inference. In regression, inference can be used to say something about the relationship between inputs and outputs but also, by virtue of knowing something about that relationship, so too can it say something about predictions to be made at new inputs.

Arguably frequentist inference is mostly, in practice, about performing analysis of variance (ANOVA) hypothesis tests in multiple-linear-regression models. While still common in practice, our aim with linear regression is to begin to introduce notions of regularization which lead naturally to Bayesian approaches to inference, the latter being a more coherent formalism for expressing uncertainty about estimates.

We'll start with linear regression as you know it

$$\begin{array}{lll} n \in \{1, \ldots, N\} & & \\ y_n \in \mathbb{R} & w \in \mathbb{R} & Y = \{y_1, \ldots, y_N\} \\ x_n \in \mathbb{R} & b \in \mathbb{R} & X = \{x_1, \ldots, x_N\} \end{array}$$

$$y_n | x_n \sim \mathcal{N}(wx_n + b, \sigma^2)$$

$$\implies \mathcal{L}(X; w, b) = \prod_{n=1}^{N} \mathcal{N}(y_n; wx_n + b, \sigma^2)$$

$$\underset{w}{\arg\max} \log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{\{-\frac{1}{2}\frac{(y_n - (wx_n + b))}{\sigma^2}\}}$$

then take derivatives w.r.t. w, b, set $= 0$ etc. (gross).

Alternatively one can formulate this as matrix linear regression with

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} w \\ b \end{bmatrix}$$

with likelihood

$$\mathcal{L}(\boldsymbol{Y}; \boldsymbol{w}) = \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \equiv \boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{I})$$

Review: Least Squares

We can estimate $\boldsymbol{w}$ by directly minimizing the MSE such that

$$\begin{aligned}
\hat{\boldsymbol{w}}_{LeastSquares} &= \arg\min_{\boldsymbol{w}} \mathbb{E}[y_n - \hat{y}_n^2] \\
&= \arg\min_{\boldsymbol{w}} \mathbb{E}[(y_n - (x_n w + b)^2] \\
&= \arg\min_{\boldsymbol{w}} \frac{1}{N} \sum_{n=1}^{N} \big(y_n - (x_n w + b)\big)^2 \\
&= \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} \big(y_n - (\begin{bmatrix} x_n & 1 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix})\big)^2 \\
&= \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} \big(y_n - (\boldsymbol{x}_n \boldsymbol{w})\big)^2 \\
&= \arg\min_{\boldsymbol{w}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}) \\
&= \arg\min_{\boldsymbol{w}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^T \boldsymbol{I} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})
\end{aligned}$$

We will solve this in the next section

Back to Linear Regression

We have

$$\arg\max_{\boldsymbol{w}} \log \mathcal{L}(\boldsymbol{Y}; \boldsymbol{w}) = \boldsymbol{w}$$
$$\text{such that} \frac{\partial}{\partial \boldsymbol{w}} \log \mathcal{N}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}; \boldsymbol{I}) = \boldsymbol{0}$$

Plugging in to the MVN pdf and taking logs we can see

$$\begin{aligned}
0 &= \frac{\partial}{\partial \boldsymbol{w}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^T \boldsymbol{I} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}) \\
\text{(Noting the similarity to least squares)} \quad &= \frac{\partial}{\partial \boldsymbol{w}} \big\{ -2(\boldsymbol{X}\boldsymbol{w})^T \boldsymbol{Y} + (\boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{X}\boldsymbol{w}) \big\} \\
&= \frac{\partial}{\partial \boldsymbol{w}} \big\{ -2\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{Y} + \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} \big\} \\
\text{From matrix cookbook} \quad &= -2\boldsymbol{X}^T \boldsymbol{Y} + (\boldsymbol{X}^T \boldsymbol{X} + (\boldsymbol{X}^T \boldsymbol{X})^T) \boldsymbol{w} \\
&= -\boldsymbol{X}^T \boldsymbol{Y} + \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} \\
\implies \hat{\boldsymbol{w}}_{ML} = \hat{\boldsymbol{w}}_{LeastSquares} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}
\end{aligned}$$

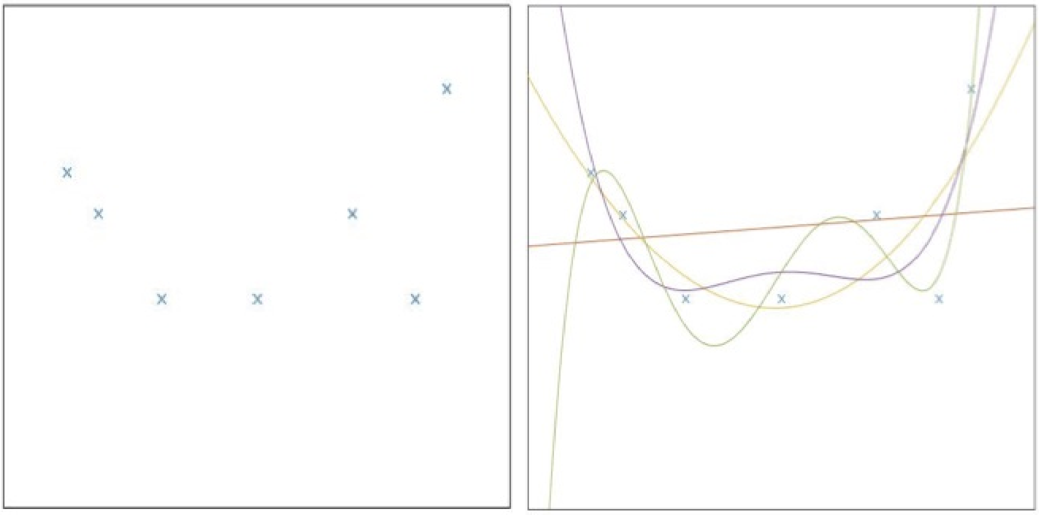
OK, so we have that the linear regression estimator is the same as the least squares estimator. Not that linear refers to linear (not affine) in the parameters.

We might ask, when can this estimator be computed? Let's look at $\boldsymbol{X}$.

$\boldsymbol{X} \in \mathbb{R}^{N \times P}$ i.e. N rows and P columns.

$$\boldsymbol{X} = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}$$

$\boldsymbol{X}^T\boldsymbol{X} \in \mathbb{R}^{P \times P}$ and $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ requires $\boldsymbol{X}^T\boldsymbol{X}$ non-singular, i.e. $\operatorname{rank}(\boldsymbol{X}^T\boldsymbol{X}) \geq P$. You might ask "when could the design matrix $(\boldsymbol{X})$ be rank deficient?" Let's consider the following linear regression.



(a) Noisy data points from some generative system

(b) Regression options: fits from polynomials of different orders

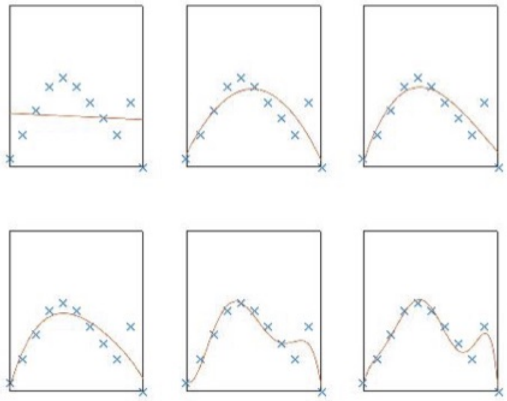
There are several possible interpretations of said data. Let's say I prefer $\boldsymbol{y} = a\boldsymbol{x}^3 + b\boldsymbol{x}^2 + c\boldsymbol{x} + d$ for some unknown $\boldsymbol{w} = \begin{bmatrix} a & b & c & d \end{bmatrix}^T$. Note that nothing from the above need change except the definitions

$$\boldsymbol{X} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N^3 & x_N^2 & x_N & 1 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

It is entirely feasible that one might prefer or even need a polynomial of order $P > N$ and $\boldsymbol{X}^T\boldsymbol{X}$ becomes rank deficient.

6.1 Regularization

As model complexity increases the ability for a model to fit the data becomes higher. Polynomial regression serves as an excellent pedagogical tool for explaining what happens when a model begins to overfit.



As the polynomial order increases the ability of the model to exactly reproduce the target values increases; however out-of-sample performance typically degrades. To combat this we typically *regularise* or *bias* models towards reasonable solutions. In this case we might like the polynomial weights to be small. One way to do this is to directly penalise large weights, i.e. modify our ML objective

$$\begin{aligned}E_D(\boldsymbol{w}) &= \frac{1}{2}\sum_{n=1}^{N}\left\{y_n - \boldsymbol{w}^T\boldsymbol{\Phi}_n(x_n)\right\}^2 \\ &= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})\end{aligned}$$

where E is an "energy", i.e. log probability, and $\boldsymbol{\Phi}(\boldsymbol{X}) \in \mathbb{R}^{N\times P}$ is the “feature” or “design” matrix given by

$$\boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{\Phi}_1(x_1) & \dots & \boldsymbol{\Phi}_P(x_1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Phi}_1(x_N) & \dots & \boldsymbol{\Phi}_P(x_N)\end{bmatrix}$$

to include a term like

$$E_w(\boldsymbol{w}) = \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

where λ is a smoothing tuning parameter.

Note that this notion of a design or feature matrix is an extremely powerful notion as it allows for arbitrary covariants and features (functions of the inputs) to be introduced as predictors.

Maximum probability corresponds to minimum energy, so we might wish to find

$$\begin{aligned}\boldsymbol{w}^* &= \arg\min_{\boldsymbol{w}}\big(E_D(\boldsymbol{w}) + E_w(\boldsymbol{w})\big) \\ &= \arg\min_{\boldsymbol{w}}\Big(\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}\Big)\end{aligned}$$

which arises at $\frac{\partial}{\partial \boldsymbol{w}}\big(E_D(\boldsymbol{w}) + E_w(\boldsymbol{w})\big) = 0$

$$\begin{aligned}\therefore 0 &= \frac{\partial}{\partial \boldsymbol{w}}\Big(\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}\Big) \\ &= \boldsymbol{\Phi}^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) + \lambda\boldsymbol{w} \\ &= -\boldsymbol{\Phi}^T\boldsymbol{y} + (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})\boldsymbol{w} \\ \implies \boldsymbol{w} &= (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}\end{aligned}$$

What does this mean and do?

1. Inverting $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ becomes unstable or impossible if $\boldsymbol{\Phi}$ is rank deficient. Adding positive elements to the diagonal ensures that $(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})$ is full rank and therefore invertible.
2. $\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$ looks a lot like a MVN, i.e. $\frac{1}{2}(\boldsymbol{w} - \boldsymbol{0})^T\lambda\boldsymbol{I}(\boldsymbol{w} - \boldsymbol{0})$ which in energy terms is minimised when $\boldsymbol{w}$ is close to $\boldsymbol{0}$ (i.e. E_w is the energy term for a zero-mean Gaussian prior).
3. This is a form of *bias* that is helpful!

This form of regularized estimator, when the regularization is interpretable as a prior, is known as a maximum aposteriori (MAP) estimator for reasons that will soon become clear.

7 Bayesian Linear Regression

To interpret this kind of regularization further we appeal to Bayesian reasoning, e.g. Bayesian linear regression.

Let’s consider Bayes rule in the context of linear regression

$$P(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\Phi}, \beta, \lambda) = \frac{P(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{w}, \beta)P(\boldsymbol{w}|\lambda)}{P(\boldsymbol{y}|\boldsymbol{\Phi}, \beta, \lambda)}$$

this is a little confusing, so let’s drop the constants

$$\begin{aligned}P(\boldsymbol{w}|\boldsymbol{y}) &= \frac{P(\boldsymbol{y}, \boldsymbol{w})P(\boldsymbol{w})}{P(\boldsymbol{y})} \\ &\propto P(\boldsymbol{y}|\boldsymbol{w})P(\boldsymbol{w}) \\ &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{w}, \beta\boldsymbol{I})\mathcal{N}(\boldsymbol{w}; \boldsymbol{0}, \lambda\boldsymbol{I})\end{aligned}$$

This is quite interesting. It says that a Bayesian approach to linear regression introduces bias and, simultaneously, that solving for the MAP $\boldsymbol{w}$ is clearly equivalent to solving a regularized least squares problem.

What is more, we can analytically derive the full posterior distribution, if we know (and understand) the following about the MVN

$$\begin{aligned}
P(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) && (7)\\
P(\boldsymbol{y}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}) && (8)\\
\implies P(\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^T) && (9)\\
\implies P(\boldsymbol{x}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{\Sigma}\{\boldsymbol{A}^T\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) && (10)
\end{aligned}$$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{L}\boldsymbol{A})^{-1}$. These facts about multivariate Gaussian distributions can be derived by completing the matrix square or simply looked up in, for instance, Bishop (2006) 2.3.3 from where these were copied.

7.1 Posterior Inference

Equations 7-10 constitute Bayes rule for Gaussians and using them we can immediately derive

$$\begin{aligned}
P(\boldsymbol{w}|\dots) &= \mathcal{N}(\boldsymbol{w}; \boldsymbol{\Sigma}\{\boldsymbol{\Phi}^T\frac{1}{\beta}\boldsymbol{I}(\boldsymbol{y})\}, \boldsymbol{\Sigma})\\
&= \mathcal{N}(\boldsymbol{w}; \frac{1}{\beta}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y}, \boldsymbol{\Sigma})
\end{aligned}$$

and

$$\boldsymbol{\Sigma} = (\frac{1}{\lambda}\boldsymbol{I} + \frac{1}{\beta}\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}$$

which is the posterior distribution of the weight vector. We already know its mode and mean: having the full distribution allows ease of model combination and propagation of uncertainty throughout inference and computation.

Let us pause for a moment here and reflect on what this means. Instead of constructing a hypothesis test in the form of H_0 : "the regression coefficient corresponding to this particular column of the design matrix is 0" and using a heuristic procedure to falsify such hypotheses one at a time in order to build up a robust, low-dimensional linear model, instead we simply define *inference* to be the characterization of the complete posterior distribution of the model parameters of interest. Such a posterior distribution assigns a weight, so to speak, to every possible setting of the parameters, here a vector of weights, that we can use to ask inferential questions like, what's the posterior probability that $w_1 > 0.8$, i.e.

$$p(w_1 > 0.8|\cdots) = \int \mathbb{I}(w_1 > 0.8)p(\boldsymbol{w}|\dots)d\boldsymbol{w}$$

which is, in the case of multivariate Gaussians is a trivial integral to evaluate if you know the marginalization and conditioning properties of Gaussian distributions here. (from Sam Roweis' excellent cheat sheets[1])

Let $\boldsymbol{z} = [\boldsymbol{x}^T\boldsymbol{y}^T]^T$ be normally distributed according to

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{B} \end{bmatrix}\right). \tag{11}$$

where all the vector and matrix partition sizes make sense then the marginal distributions of $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\begin{aligned}
\boldsymbol{x} &\sim \mathcal{N}(\boldsymbol{a}, \boldsymbol{A}) && (12)\\
\boldsymbol{y} &\sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B}) && (13)
\end{aligned}$$

which, along with the cumulative distribution function of the standard normal is sufficient to answer our posterior inference question.

Also the conditional distributions (not needed specifically here but still supremely useful) are

$$\begin{aligned}
\boldsymbol{x}|\boldsymbol{y} &\sim \mathcal{N}(\boldsymbol{a} + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{b}), \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^T) && (14)\\
\boldsymbol{y}|\boldsymbol{x} &\sim \mathcal{N}(\boldsymbol{b} + \boldsymbol{C}^T\boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{a}), \boldsymbol{B} - \boldsymbol{C}^T\boldsymbol{A}^{-1}\boldsymbol{C}) && (15)
\end{aligned}$$

[1] `http://www.cs.nyu.edu/~roweis/notes.html`

7.2 Prediction as Inference

Often we are interested in making predictions about an output y for new data x. This can be written as

$$P(y|x, \boldsymbol{w}, \boldsymbol{\Phi}, \lambda, \beta) = \int P(y|\boldsymbol{w}, x, \beta)P(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\Phi}, \lambda, \beta)d\boldsymbol{w}$$

where the first term in the integral $P(y|\boldsymbol{w}, x, \beta)$ is the model of the noise $\mathcal{N}(y; \boldsymbol{x}^T\boldsymbol{w}, \beta)$ and the second term $P(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\Phi}, \lambda, \beta)$ is the posterior of $\boldsymbol{w}$, $\mathcal{N}(\boldsymbol{w}; \frac{1}{\beta}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{-1}\boldsymbol{y}, \boldsymbol{\Sigma})$. Makes use of Eqn. 9 above we can see that this integral has an analytic form as well, namely

$$P(y|x, \boldsymbol{w}, \boldsymbol{\Phi}, \lambda, \beta) = \mathcal{N}(y; \boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{\Phi}^{-1}\boldsymbol{y}, \frac{1}{\beta} + \boldsymbol{x}^T(\frac{1}{\lambda}\boldsymbol{I} + \frac{1}{\beta}\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{x})$$

Pictorially this looks like

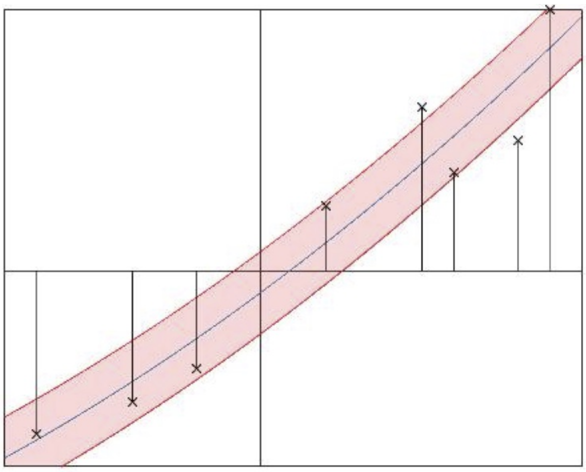


Figure 6: The integration over $\boldsymbol{w}$ averages over "possible lines" yielding a distribution over predictions.

As we can see here even the predictive distribution is Gaussian in the Bayesian linear regression setting. What does this mean though? By propagating uncertainty about the value of the weights throughout the computation we get error bars on our predictions. Analyses using those are constrained by the model family but otherwise are safer, in general, than using point estimates.

We may also note at this point a few important points that apply not only to this example to any inference task treated in this way. First, MAP estimators can be interpreted as being a lot like estimator **C** in that the regularization imposed by a prior can be used to inject knowledge about your guess as to what the parameter value should be, injecting a form of bias that can be useful. Second, and perhaps more important pedagogically, is that it should be acknowledged that the analytic kind of Bayesian posterior inference results derived for the linear regression model do not exist for the predominant portion of definable models. This does not mean that posterior inference is impossible, it simply means that the characterization of the posterior distribution might need to be computational and approximate. C19 expands significantly on these kinds of techniques.

Perhaps the most important take-home from this requires one to step back from this specific regression example to note that inference, i.e. asking a question about the nature of the world or making a prediction about something yet to be seen, can be framed in terms of manipulation of conditional probabilities in a way that is in keeping with Bayes rule plus integration of a test query against the resulting posterior.

Lastly, a particularly powerful generalization of these Gaussian results can be derived in the case of an infinite dimensional parameter space, i.e. an infinite-dimensional Gaussian distribution known as the Gaussian process. Additional non-examinable reading on these fascinating and surprisingly useful mathematical constructs can be found in (Rasmussen, 2006).

8 Classification

Classification and regression are close cousins. In probabilistic regression we model the conditional distribution of output given input (features).

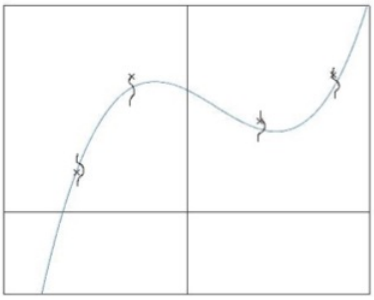


Figure 7: Output $\boldsymbol{y}$ as a function of input features $\boldsymbol{x}$

In 2-class classification we do the same except the output is discrete rather than continuous, e.g. $y_i = 1 \implies x_i \in C_1$.[2] The goal is to learn a conditional distribution of the class label given the input.
Imagine, in 1-D input for now, that we have data like

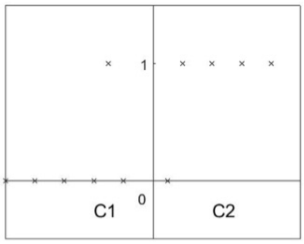


Figure 8

It's clear that a linear regression fit to such data will do *something* with a decision rule like $x_i \in C_1$ if $y_i < 0.5$ but such a rule a) doesn't allow a sharp decision boundary and b) doesn't have a coherent probabilistic interpretation: i.e. what does output = 7 mean?

What we might like is 1) a "noise" distribution that penalizes misclassification appropriately and 2) a controllable function that stretches input space. The Bernoulli distribution is just the ticket for 1). For any given y_i we can write

$$P(y_i|\dots) = p_i^{y_i}(1-p_i)^{1-y_i}$$

which if we know $P(y_i = 1) = p_i$ then "getting it right" has likelihood p_i and wrong $1 - p_i$. If $p_i = 0.9$ and we observe $y_i = 1$ then we're better than if we observe $y_i = 0$. The only question is, what's p_i? We only know that p_i must be between 0 and 1; the choice is arbitrary past that.

For now, assume $x_i \in \mathbb{R}$. Let's consider

$$p_i = \frac{1}{1 + \mathrm{e}^{-x_i\beta}}$$

the logistic sigmoid function. More generally

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$

For large values of x, $\lim\limits_{x\to\infty} \sigma(x) = 1$, and for small $\lim\limits_{x\to-\infty} \sigma(x) = 0$. The function looks like

[2] Here C_1 means 'Class 1'.

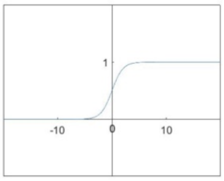


Figure 9: $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$

This is a non-linear function that stretches x-space. One may shift this function by adding an offset

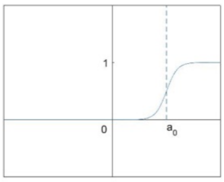


Figure 10: $\sigma(x - a_0) = \frac{1}{1+\mathrm{e}^{-(x-a_0)}}$

or make the slope more or less steep by adding a weight in front of x

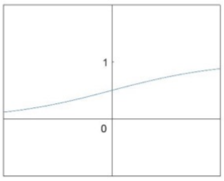


Figure 11: $\sigma(a_1 x) = \frac{1}{1+\mathrm{e}^{-(a_1 x)}}$

The logistic sigmoid function has some nice properties which can be verified algebraically, for instance

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

and

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

8.1 Logistic Regression

With these choices, likelihood and non-linear transform, we get logistic regression classification which imposes the following classification likelihood

$$p(\boldsymbol{Y} = \{y_1, \ldots, y_N\} | \boldsymbol{X} = \{\boldsymbol{x}_1, \ldots \boldsymbol{x}_N\}; \boldsymbol{\beta}) = \prod_{n=1}^{N} \left(\sigma(\boldsymbol{x}_n^T \boldsymbol{\beta})\right)^{y_n} \left(1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\beta})\right)^{1-y_n} \tag{16}$$

where $\boldsymbol{x}_n \in \mathbb{R}^{D+1}, \boldsymbol{\beta} \in \mathbb{R}^{D+1}, y_n \in \{0, 1\}$, and $\boldsymbol{x}_n = \begin{bmatrix} 1 \\ \boldsymbol{x}_n \end{bmatrix}$.

Given a dataset $\boldsymbol{X}, \boldsymbol{Y}$ logistic regression learns a projection vector $\boldsymbol{\beta}$ that finds a linear decision boundary that gracefully tolerates misclassifications as shown in the following figure.

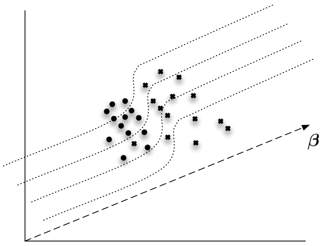


Figure 12: Linear decision surface formed in $\boldsymbol{x}$ space by logistic sigmoid function.

The standard question is, of course, how to train logistic regression. Maximum likelihood methods can be employed here as per usual. It can be verified easily that

$$\frac{\partial}{\partial\boldsymbol\beta}\log\mathcal{L}(\boldsymbol{Y},\boldsymbol{X};\boldsymbol\beta) = \frac{\partial}{\partial\boldsymbol\beta}\log\left(\prod_n \left(\sigma(\boldsymbol{x}_n^T\boldsymbol\beta)\right)^{y_n}\left(1-\sigma(\boldsymbol{x}_n^T\boldsymbol\beta)\right)^{1-y_n}\right) \tag{17}$$

$$= \sum_n y_n \left(\sigma(\boldsymbol{x}_n^T\boldsymbol\beta)x_n^T\right) - \sum_n (1-y_n)\left(1-\sigma(\boldsymbol{x}_n^T\boldsymbol\beta)\right)x_n^T \tag{18}$$

which effectively pulls and pushes on $\boldsymbol\beta$ depending on the sign induced by the value of y_n.

Unfortunately you may also verify that if you set this equal to 0 and attempt to solve for $\boldsymbol\beta$ you will not arrive at an analytic solution.

While there are fancier techniques for quickly optimizing functions (see B1 and C25), including likelihoods, it is worth driving home the utility of gradient descent/ascent as a general optimization technique, here used for estimation of maximum likelihood parameters of a parametric classification model.

Revision: Gradient Ascent

Gradient ascent is the simplest and in some ways the most important tool in the statistics / machine learning arsenal.

In 1D let's say we have a function

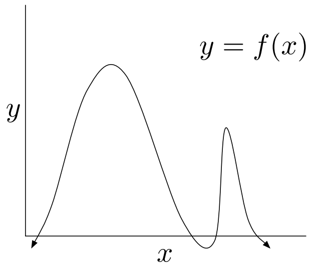


Figure 13: Some function $y = f(x)$.

$y = f(x)$, a starting point x_0, and we would like to find the value x_{max} that maximizes $f(x)$. If we are able to compute the derivative of f with respect to x, $\frac{df}{dx}$ then we can ascend f by starting at x_0 and stepping in the direction opposite that of $\left.\frac{df(x)}{dx}\right|_{x_0}$.

This suggestions an iterative procedure for maximum likelihood estimation, a special case of function maximization, that is powerfully general if not always optimally efficient:

Algorithm 1 Gradient Ascent of f w.r.t. x given $\frac{df}{dx}$

```
$\eta \leftarrow$ learning rate
$x = x_0$ starting point
while still ascending do
    $x = x - \eta \left.\frac{df(x)}{dx}\right|_x$
return $x$
```

Some important things to note.

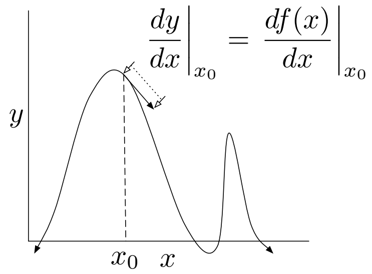

$\left.\frac{dy}{dx}\right|_{x_0} = \left.\frac{df(x)}{dx}\right|_{x_0}$
y
x_0
x

Figure 14: Slope of some function $y = f(x)$ evaluated at x_0.

- If f is multimodal this procedure may ascend to a single non-optimal mode.
- η, the learning rate, is very important – too large and this procedure can diverge (think of resonating back and forth across a quadratic hump by stepping too far every time) or too small and this procedure might take forever to converge.
- These exact arguments and in fact the procedure works in high-dimensions with gradients; more serious considerations apply though – see conjugate gradients (C25), adagrad, ADAM, etc.

As a practical consideration you should always check your gradient calculations and computation algorithms by comparing your computed gradient to finite differences computed for each dimension via $\left.\frac{df(x)}{dx}\right|_x \approx \frac{f(x+\Delta x)-f(x)}{\Delta x}$ for some small choice of $\Delta x \approx 10e^-5$.

Back to Classification

What we have uncovered here is relatively profound, namely, that classification can be framed as probabilistic regression with a different likelihood. Taking this view allows us to leverage all the techniques for parameter estimation and inference for regression and apply them directly to classification. For instance regularizing logistic regression can be seen as placing a prior on the classifier parameters $\boldsymbol{\beta}$ which might allow, for instance, a larger number of features to be included in the classifier, potentially improving its performance. Maximum likelihood parameter estimation can be used to train classifiers, so too can MAP inference. Fully Bayesian treatments of classification require techniques beyond the scope of these lectures, however this too is possible.

It should be noted that the energy perspective of linear regression can be taken for classification as well, in fact this is the primary perspective that has been taken by the field of machine learning for the last few decades leading up until now. This perspective allows different objectives/losses to be considered (rather than likelihood) (for example one could simply count the number of data-points correctly classified and attempt to optimize that quantity directly). Different losses and different forms of regularization can be framed in the energy minimization framework and these give rise to support vector machines and neural nets (also topics of more advanced courses, though, in theory not terribly more difficult than anything covered here).

Having showed that regression and probabilistic approaches to classification are closely related, it is worth noting that the regression approach is not the only approach to classification. In fact there are widely divergent views on how to perform classification; they are, roughly

- Model $p(x|y)$ and $p(y)$ where y is the class of datum x then assign class via Bayes rule

  $$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

  which requires learning the data distribution $p(x)$ which is typically harder than the next approach (generative modeling, for instance Gaussian mixture models; covered in C19)
- Model $p(y|x)$ directly (logistic and linear regression are examples)
- Learn $f(x) : \mathcal{X} \to \mathcal{Y}$ directly (function approximation without probabilistic interpretation)

9 Conclusion

To conclude these four lecture let's consider inference, estimation, and classification.

We have considered ML estimation in which a point estimate of a parameter within a particular model family is found by maximizing a likelihood function. We have seen that in big N little p settings, meaning lots of data and a few parameters, we can estimate model parameters, particularly in the model setting.

Further we can perform *inference* in two different ways; we can examine the sampling distribution of estimators *or* we can compute posterior distributions of parameters. The former is called frequentist inference, the latter Bayesian.

Perhaps as important as what has been taught in this course is what has been not. Mathematical statistics for engineering and, for instance, inference as frequentist hypothesis testing for linear models is worth an entire course as it may well be one of the most important and frequently used things you end up doing in your career.

For expediency we have quickly moved to Bayesian methods of inference which have the convenience of being easier to formalize coherently but the disadvantage of being less well recognized and used in the real world, still, even today, though this is thankfully changing.

We have not dwelled for a long time on how to formulate a model, how to choose features, or how to choose which inference questions to pose as posterior integrals. These are largely application specific and, given sufficient exposure to a domain, easy-enough to come up with.

Finally, we have not covered Bayesian inference in any detail at all; notably not even given a treatment of Bayesian inference in logistic regression. The conceptual framework should be clear now though, even if how to compute or represent such a posterior is not.

References

Peter J Bickel and Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics*. CRC Press, 2015.

Richard A Johnson. *Probability and statistics for engineers*. Prentice Hall, 2000.

Christopher M Bishop. Pattern recognition and machine learning (information science and statistics). 2006.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

Carl Edward Rasmussen. Gaussian processes for machine learning. MIT Press, 2006.

A Probability distributions

Summarised in Table 1.

Distribution	Parameters	Support	PDF/PMF	Mean	Variance/Covariance
Bernoulli	$\theta \in [0,1]$	$x \in \{0,1\}$	$\begin{cases} \theta & \text{if } x = 1 \\ 1-\theta & \text{if } x = 0 \end{cases}$	θ	$\theta(1-\theta)$
Beta	$\alpha, \beta > 0$	$x \in [0,1]$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Binomial	$N \in \mathbb{N}, \theta \in [0,1]$	$x \in \{0,\dots,N\}$	$\binom{N}{x}\theta^x(1-\theta)^{N-x}$	$N\theta$	$N\theta(1-\theta)$
Gamma	$\alpha, \beta > 0$	$x > 0$	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x)$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$
Categorical	$\boldsymbol{\theta} \in [0,1]^K, \sum_k \theta_k = 1$	$x \in \{1,\dots,K\}$	θ_x	N/A	N/A
Dirichlet	$\boldsymbol{\alpha} \in (0,\infty)^K$	$\mathbf{x} \in [0,1]^K, \sum_k x_k = 1$	$\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k x_k^{\alpha_k - 1}$	$\frac{\boldsymbol{\alpha}}{\sum_k \alpha_k}$	$\text{Var}[x_k] = \frac{\alpha_k(\alpha_0-\alpha_k)}{\alpha_0^2(\alpha_0+1)}$
Multivariate Normal	$\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma}$ positive semi-definite $d \times d$	$\mathbf{x} \in \mathbb{R}^d$	$\frac{1}{\sqrt{\lvert 2\pi\boldsymbol{\Sigma}\rvert}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$
Normal	$\mu \in \mathbb{R}$, $\sigma^2 > 0$	$x \in \mathbb{R}$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	μ	σ^2
Poisson	$\lambda > 0$	$x \in \{0,1,2,\dots\}$	$\frac{\lambda^x}{x!}\exp(-\lambda)$	λ	λ

Table 1: Summary of common probability distributions

[a]

[a] Where $\Gamma(z) := \int_0^\infty x^{z-1}e^{-x}\,\mathrm{d}x$ is the *Gamma function*; more on `https://en.wikipedia.org/wiki/Gamma_function`.