Lecture Notes: Week 1

Matt Thomson

Abstract We are turning our attention to the analysis of single cell transcriptional profiling data (single cell mRNA-seq). This week our goal is to start thinking about this data in the context of statistical models that we can parameterize from observations. In the first lecture, I will discuss the application of maximum likelihood (ML) methods to estimate parameters in models of univariate gene count distributions. The ML method is powerful because it lets us simply set up and solve a wide variety of inference problems from a single framework. Further, we will see how we can define and estimate error associated with inference, and how this can be applied to solve important problems in single cell analysis. My goal this week is to give you the tools that you need to build a single cell 'cancer detector' in your homework set. You will use actual public data to (a) train a simple statistical model of gene expression in healthy and AML (acute myeloid leukemia) human immune systems and then (b) apply this model to calculate likelihood ratios on single cells coming from *new* patient samples. We will analyze the performance of this classifier on new data versus gene and cell number.

Getting Oriented: Maximum Likelihood Estimation

This week we are going to talk in depth about probabilistic models of single cell data. Single cell data allows us to analyze full joint probability distributions, $P(\boldsymbol{g})$, where $\boldsymbol{g} \in \mathbb{R}^{20000}$. We are going to start with univariate (marginal distributions) where we think about a single gene at a time, $P(g_i)$. Studying such distributions can give us insight into how gene expression changes across different conditions. Further, even such marginal distributions can yield a basic (independent), generative model of a cell (the ideal 'gene gas') that is good enough to start doing non-trivial things like cancer cell classification.

Caltech, e-mail: mthomson@caltech.edu

First, we are going to learn how to parameterize models of gene expression from data using ML inference. We will apply the models to perform binary hypothesis testing in the homework. In a future lecture, we will discuss extensions of this approach for comparing entirely different classes of models.

Our first topic is Maximum Likelihood (ML) estimation. In this framework, we construct a 'model' of an underlying process (gene transcription). This model comes in the form of a conditional probability distribution, $P(\boldsymbol{g}|\boldsymbol{\lambda})$ describing the conditional probability of, for example, observing a cell with gene counts, $\boldsymbol{g}$ given an underlying model of the cell's gene expression state. In this formulation, $\boldsymbol{g}$ is, in general, a vector with an integer number of counts for each gene, $g \in \mathbb{N}^{20000}$. The distribution also carries a vector of parameters $\boldsymbol{\lambda} \in \mathbb{R}^k$. Our parameters will be largely real numbers. In this first instance of ML estimation, we think of our models coming from a space of functions indexed by parameters $\boldsymbol{\lambda}$. The underlying parameter space can have constraints (See optional homework problem).

Given a model and data, we aim to parameterize the model from the data. To do so, we define a function called the likelihood function. For completeness and clarity, I am going to give a formal definition of this function.

Let $P(g,\boldsymbol{\lambda})$ denote a parametric set of densities with parameters $\boldsymbol{\lambda} \in \mathbb{R}^k$, let $D = \{\boldsymbol{g_1}, \boldsymbol{g_2} \cdots \boldsymbol{g_n}\}$ be i.i.d measurements of gene expression. The function

$$\mathrm{L}_n(\boldsymbol{\lambda};D) = \log \prod_{i=1}^{n} P(\boldsymbol{g_i},\boldsymbol{\lambda}), \tag{1}$$

is the log likelihood function. Notice that the product is indexed over measurements (as opposed to genes). I have also tried to be explicit that D is a parameter in the function.

Using this function, we can parameterize out model(*estimating* λ as $\widehat{\lambda}$), by maximizing, $L(\lambda;D)$, called the likelihood (often we will use $\log(L(\lambda;D))$, the log-likelihood) over parameters $\boldsymbol{\lambda}$, for an observed set of data, D (called the evidence).

The beautiful thing about the ML framework is that it provides a procedure for going from a statistical model to inference. In this framework, we can simply set-up and solve a large range of modeling problems. We simply need to (i) construct a set of candidate models (ii) estimation parameters of these models (and associated error) from data (iii) apply models to problems of interest including cell-state classification and differential expression.

For example, you are going to construct a model, $P(\boldsymbol{g}|\mathrm{AML})$ by parameterizing univariate gene expression distributions from data. You will statistical models of $P(\boldsymbol{g}|\mathrm{AML})$ and $P(\boldsymbol{g}|\mathrm{Health})$ to build a classifier that runs on new single cell patient

data.

Quick summary of the framework:

ML procedure in brief

- Construct $P(\boldsymbol{g}|\boldsymbol{w})$ ('modeling' or as statistical object eg neural net).
- Parameters $\boldsymbol{w} \in W$, hypothesis space; continuous, constraints
- Maximize $L_n(\boldsymbol{\lambda};D)$ over $\boldsymbol{\lambda}$ given data (easy to very difficult).
- Error estimates using $\frac{d^2L}{d\lambda_i^2}$

References: David MacKay. Information Theory, Inference, and Learning Algorithms. I am including chapter 2 on the web-site. Convex Optimization chapter 7

Getting Oriented: Lets talk about the data

When we perform single cell mRNA-seq, we extract a gene count vector, $\boldsymbol{g}$ for every observed cell in a population. Each cell, we can think of each cell as a vector of natural numbers, $\mathbf{g} = (\mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_n})$, where each $g_i \in \mathbb{N}$ represents the number of mRNA molecules detected from a single gene (across n genes). note: When we start analyzing real data, gene counts, $g_i \in \mathbb{R}$, will become continuous due to normalization.

To give you a sense for numbers, in a human cell, $n \sim 20,000$, and cells contain around $10^5 - 10^6$ mRNAs so that $\sum_{i=1}^{n} g_i \sim 5 \times 10^5$. mRNA is a few percent of total RNA. Single cells express about 5000 different genes, so if we take 5×10^5 total mRNAs as a typical number, a rough guess is that each gene has a copy number of around 100. In your problem set, you will look up transcript numbers from one of Long Cai's papers, and we will also look at this estimate from data.

When we are thinking about single cell data, distributions become important immediately. Gene expression is a stochastic chemical phenomena. Cells experience regulation and so exist in different 'states'. Finally, the measurement itself goes through a noisy measurement process or 'channel' due to more chemical effects that occur during library preparation and sequencing.

Id like to start by having you think about a basic diagram of the measurement we are making and the data that emerges. We are thinking about a population of cells, c_i.

$$P(\boldsymbol{g}) \boxed{\xrightarrow{\text{measurement}}} \widehat{\boldsymbol{g}}$$

So we start with a distribution of gene expression states in the cell population, and we extract a series of vectors which are single cell measurements.

It is often convenient to represent the data as a matrix, $\boldsymbol{D}$:

$$\begin{bmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1n} \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{m1} & g_{m2} & g_{m3} & \cdots & g_{mn} \end{bmatrix}$$

D is a random matrix in the sense that, we could sample a cell population, and each time we would get back a different matrix (even for fixed m and n). Each draw will yield different samples.

Many papers have analyzed stochastic gene expression in single cells with mRNA-seq.

https://www.nature.com/articles/nature13920

Parameter Estimation with Simple Transcription Model

Key Lesson: ML inference on Poisson gene count distributions.

- Mathematical Mechanics of estimation procedure
- Learn simple Poisson transcription model
- Error estimation with curvature estimation

Let's start with a very simple model of transcription and perform parameter inference on gene count data generated from this model. Imagine a population of single cells. We have the ability to detect gene counts and are going to focus on a single gene, g. In the next lecture we will talk about measurement or sampling noise, but lets first consider the problem of perfect sampling. What can we learn about the abundance and expression of this gene?

In general, each cell will have a different number of transcripts of g so that g is a discrete random variable with distribution $P(g)$. In general, many reasons but lets focus on a simple model.

Basic models of transcription

Simple transcription results in Poisson distributed gene count distributions at steady state

A lot is known about the 'forward' models or expectations. Let's get some intuition for $P(g)$ by developing simple models of transcription. Lets think about what needs to happen for transcription to occur. We have a gene, a section of DNA in the genome. There are molecules called RNA polymerase molecules that can bind to the genome, unwind DNA locally, and catalyze the synthesis of an mRNA molecule based upon the DNA template in an ATP dependent fashion.

Many aspects of this process are not deterministic but are subject to what we might call 'stochastic' or probabilistic effects. A cell might have k polymerase molecules. At any given time, these molecules diffuse in the cell, and bind DNA, they might, then, start a processive transcription event, or fall off the DNA.

We can think about this basic model as being impacted by at least two stochastic process, basic transcription, and basic degradation of the mRNA product (typical half-life hours).Instead of thinking about g directly, we are going to think about the distribution of g across a population of cells and derive an expression for this distribution $P(g,t)$. Specifically, we will show that g is Poisson distributed.

We think of g as the number of molecules of the gene G in the system. Therefore, we can model a single cell as being in a state g and a Markov process takes the cell from state to state through transcription and degradation of the transcript.

$$g \xrightarrow{p_t} g+1$$
$$g \xrightarrow{p_d} g-1$$

where p_t is the probability of a transcription event in a time dt and p_d is the probability of a degradation event in the same time window.

We write down a differential equation for $P(g,t)$ and model the probability flux of cells through different states in the population.

$$\mathrm{g} \neq 0 : \frac{dP(g)}{dt} = p_t\, P(g-1) - p_t\, P(g) + p_d\, (g+1)\, P(g+1) - p_d\, g\, P(g)$$

$$\mathrm{g} = 0 : \frac{dP(0)}{dt} = -p_t P(0) + p_d\, P(1).$$

Note, that the probability flux depends on a probability density $P(g)$ multiplied by a probability, p_d. Also note that, in the case of degradation, we multiply the probability flux by the number of molecules in a given state. This accounts for the fact that any single molecule can be subject to degradation. Equations of this form are known as chemical master equations. They are equations over probability density in a discrete space of chemical states and are related to Fokker-Planck equations in physics.

We can solve for $P(g)$ when the process is stationary, $\frac{dP}{dt} = 0$. We have $P(1) = \frac{p_t}{p_d} P(1)$. Now, the rest of the system can be solved through a kind of induction procedure.

$$\begin{aligned}
0 &= p_d\, P(1) + 2p_d\, P(2) - P(1)\, (p_t + p_d) \\
P(2) &= P(1)\frac{p_t}{2\, p_d} = P(0)(\frac{p_t}{p_d})^2(\tfrac{1}{2}) \\
P(g) &= P(0)(\frac{p_t}{p_d})^g(\tfrac{1}{g!}) \\
&\sum_{g=1}^{\infty} P(g) = 1 \\
&\sum_g P(0)\, (\frac{p_t}{p_d})^g\, \frac{1}{g!} = 1 \\
P(0) &= \exp(-\tfrac{p_d}{p_t})
\end{aligned}$$

Therefore,

$$P(g) = \frac{\lambda^g\, \exp(-\lambda)}{g!} \tag{2}$$

This is a very nice solution, the distribution is Poisson. I like this derivation because it gives you a sense of this interplay between the chemical modeling and the distribution form. Further, see how a chemical process can give rise to a distribution of states in a cell population. The chemical model is generative in this interesting way. What distributions could you generate with chemistry?

The mean and variance of the distribution are both λ. Notice that $\lambda = tfracp_t p_d$ which is interesting. At steady state, we will estimate lambda. Think about how you could get information on the underlying parameters individually.
The mean and variance are equal:

$$E[g] = \lambda$$

$$(E[g^2] - E[g]^2) = \text{var}(g) = \lambda$$

This simple analysis has been a work-horse in the field of single cell gene expression. People measure something they call the coefficient of variation:

$$\text{C}_v = \frac{\sqrt{\text{var}(g)}}{E[g]}$$

For a Poisson, $C_v^2 = \frac{1}{\lambda}$ and the Fano factor ($\frac{\text{var}(g)}{E[g]} = 1$)

Estimating λ using ML estimation

With this model in hand, we can do ML estimation. We, specifically, have the required conditional probability density.

$$P(g|\lambda) = \frac{\lambda^g \ \exp(-\lambda)}{g!} \tag{3}$$

Now, imagine that we perform a series of measurements collecting data g_i from this population where i indexes measurement, $i \in 1 \cdots n$, and n is the cell number. This is our data or evidence.

We calculate the log-likelihood as:

$$L(\lambda; g) = \sum_{i=1}^{n} \log(P(g_i|\lambda)) \tag{4}$$

Notice that, for a sequence of (i.i.d) measurements, $\boldsymbol{g} = g_1 \cdots g_n$ taken over n cells, the log-likelihood has this nice property of summing over the evidence. This result emerges from the independence of the joint measurement distribution

$P(g_1, g_2) = P(g_1)P(g_2)$.

For a sequence of measurements g_i, we find λ by through direct evaluation:

$$\text{argmax}_\lambda L(\lambda; g).$$

Note, that log is concave and by setting $\frac{d\log(L)}{d\lambda} = 0$, we maximize the function over the continuous set of parameters $\lambda \in R$.

Note, this maximization is an important step in the procedure. In some cases (multinomial distribution), we will want to perform this operation with constraints on the domain of the parameters. For example, when estimating probabilities we want $\sum_i p_i = 1$ as can be imposed with constraints. This is a generally interesting topic and intersects with methods from linear programming.

We take:

$$L(\lambda) = \sum_i g_i \log(\lambda) - \lambda + \log(g_i!)$$
$$\frac{dL}{d\lambda} = \sum_i (\frac{g_i}{\lambda} - 1)$$

We see directly that $\frac{dL}{d\lambda} = 0$, when

$$\widehat{\lambda} = \frac{\sum_i g_i}{n}, \tag{5}$$

where we are calling our ML estimate of λ by the name $\widehat{\lambda}$. We know, therefore, that $\widehat{\lambda}$ approaches λ_0 the 'true' value of $\lambda = \frac{p_t}{p_d}$ as n, and thus, our number of observations increases. So we can effectively estimate the steady state level of the gene from measurements.

An interesting feature of the ML method is that we can study the shape of $L(\lambda)$. We can calculate the curvature of the function as this point, and find that,

$$\frac{d^2\log(L)}{d\lambda^2}|_{\lambda_0} = -\sum_i \frac{g_i}{\lambda^2}$$

so that the curvature of $L(g, \lambda) < 0$ and the function is concave for positive λ. Our solution is a global maximum. In this way, we found a simple and natural estimate of λ from our data.

Also notice that $\frac{d^2 \log(L)}{d\lambda^2} = \frac{\widehat{\lambda}\, n}{\lambda^2}$.
The result has some nice additional properties. Using a method that comes from Laplace, we can immediately estimate the error in $\widehat{\lambda} = \lambda_0$ (our estimate of λ). We perform a Taylor expansion of the log-likelihood around λ_0, this is a saddle point approximation. I immediately we have that $\frac{dL}{d\lambda} = 0$ at $\hat{\lambda} = \lambda_0$.

$$L(\lambda) = L(\lambda_0) + \frac{d^2L}{d\lambda^2}|_{\lambda_0} \frac{(\lambda - \lambda_0)^2}{2} \cdots$$

$\frac{d^2L}{d\lambda^2}$ is going to be the critical quantity in our analysis. Lets call $c = \frac{d^2L}{d\lambda^2}$, and remember that $L(\lambda) = \log(P(\boldsymbol{g}|\lambda))$. In the vicinity of λ_0

$$P(\boldsymbol{g}|\lambda) \sim \exp(c\frac{(\lambda - \lambda_0)^2}{2})$$

We can use this local Gaussian approximation to calculate the variance in our estimate near the maximum of the likelihood function as $\sigma^2 = -\frac{1}{\frac{d^2Q}{d\lambda^2}} = \frac{1}{c}$.
For the Poisson estimation task, $\frac{d^2L}{d\lambda^2} = -\sum_i \frac{g_i}{\lambda^2} = \frac{n}{\lambda_0}$. SO that the $\mathrm{Var}[\widehat{\lambda}] = \frac{\lambda}{n}$.

Using these ideas, we can estimate confidence intervals for our data. We calculate c directly and we can quote error bars for our estimate directly from the data using c. See example in mathematica notebook.

The Fisher information is the expected value of c given the underlying physical system. The Fisher information, $I(\lambda)$, tells us on average home much observations will carry about an unknown

is the expected value of the curvature:

$$I(\lambda) = \langle -(\frac{d^2L(\lambda)}{d\lambda^2})|_{\lambda_0} \rangle$$

Where the expectation value is taken over the distribution from which the data is coming.

For example, for the Poisson distribution, the Fisher information is:

A result known as the Cramer-Rao bound says that (similar logic to the above):

$$E[(\lambda - \widehat{\lambda})^2] \geq / \frac{1}{n\, I(\lambda)}$$

For $\mathrm{Poi}(\lambda)$, $I(\lambda) = \frac{1}{\lambda}$. See Mathematica note book.
This is a very useful result because we can define fundamental limits on sensing. For example in the Poisson case, for mean gene aboundance λ you can show that we required $n > \frac{1}{\lambda * \varepsilon^2}$ cell measurements to achieve error ε.

State-Switching and Gamma Distribution

In general, our model of transcription is too simplified and ignores phenomena like promoter state switching. Results in the literature show that promoter toggling leads to a Gamma distribution, so that:

$$P(g) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\, g^{\alpha-1}\, \exp(-\beta m)$$

where $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} \exp(-x)\, dx$.

Try plotting this function, it has a shape defined by the rise of a polynomial and the fall of an exponential.
https://www.sciencedirect.com/science/article/pii/S1046202313000959
https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html

Mixture of Poisson Sources

The nice thing about this estimate is that there is a very simple route to generalization. As long as we can construct a conditional probability model of our system, we can perform ML estimation of parameters and error analysis. Two examples that we will pursue–you could imagine complicating our model or making it more realistic in two different ways.

Adding additional states to the cell population. Imagine that instead of a single transcriptional process in the cell population we have two classes of cells. In one class of cells, are gene is expressed at λ_1 and in the second at λ_2.
Lets model the population as a whole as a mixture of Poisson distributions with weighting w_1 and $w_2 = 1 - w_1$.

$$P(g|\lambda_1, \lambda_2) = w_1 \exp(-\lambda_1)\frac{\lambda_1^g}{g!} + (1 - w_1)\, \exp(-\lambda_2)\frac{\lambda_2^g}{g!}$$

We now have a likelihood function with three variables and a more complex structure.

However, we can proceed as before. Notice that we can still calculate $\frac{dL(g))}{d\lambda_1} = 0$ but that the resulting equation is now non-linear and has more zeros.

Binary Hypothesis testing

In the first problem set, we are going to construct statistical models of two single cell data sets on a gene by gene basis. With these statistical models, in hand we are going to build a cancer detector using binary hypothesis testing.

One of the reasons that ML estimation is so useful is that it enables direct comparison of models to do. In the problem set, you will consider the case where you have two models of cellular gene expression. One is parameterized from cancer cells and the other from healthy cells, and you will apply the two models to new cell populations to estimate a likelihood ratio for cancer.
Consider two models or hypothesis:

- $H_0 = P(g|\lambda_0)$
- $H_1 = (g|\lambda_0)$

Given a sequence of i.i.d measurements $g_1 \cdots g_n$, we would like a test that allows us to discriminate between the two hypothesis.

We aim to construct a decision function that we can apply to decide whether H_0 or H_1 is accepted. We can think of the decision function as specifying a set in $\boldsymbol{g}$ the measurement space in which H_0 or H_1 is going to be accepted. Call this set A_n.

We can think of two types of error. False positive error in which we call a measurement to be in H_1 when it was actually came from λ_0. The probability that our decision function fires true on a null example. Second, we define a false negative probability where the decision function fires false when the sample was in fact in λ_1.

In general you would like to minimize both probabilities, but there is a trade-off between them. We can give a bound on the best achievable error in classification by the properties of P_1 and P_2 in terms of the KL-divergence.
For now, we are going to talk about optimal tests. The Neyman-Pearson lemma says that the optimal test for this decision problem is the likelihood ratio test.

We construct a decision region:

$$A(T) = \{g : \frac{P(g|\lambda_1)}{P(g|\lambda_0)} > T\}$$

In practice it is often useful to use the log of this ratio. So that we consider:

$$\mathrm{LR}(\lambda) = \log \frac{P(g|\lambda_1)}{P(g|\lambda_0)}$$

This quantity LR is called the evidence for hypothesis 1.

A few comments on this ratio. First, evidence is additive over samples, so we can calculate a cumulative evidence for i.i.d measurements due to the behavior of log.
Second, optimal values of T can be determined from models of the system. See mathematica notebook.

Inference and information

Generative Models

Once we have a model $P(g)$, an interesting feature of this model is that we can (i) sample from it to generate synthetic data (ii) use it to calculate the likelihood of new observations with respect to that model. Next time we will talk about hypothesis tests and other aspects of this framework. Now, I will simply ask you to think about the following extension.

We consider a gene expression distribution, $P(\boldsymbol{g})$ where $g \in \mathcal{N}^m$, a large gene expression space. Let us approximate $P(g)$ by assuming that g_i are independent, so that:

$$P(\boldsymbol{g}) \approx \prod_{i=1}^{m} P(g_i)$$

In general, this will not be an accurate approximation. However, it can be used to provide a useful first order model of a process. See for example Bialek:

Generalizations

In ML estimation, we are considering a set of distributions, $q \in Q$ where Q is a family of distributions that maximize the likelihood:

$$\text{argmax q}(Data)\ \text{q} \in \mathbb{Q}$$

lets instead consider that we want to minimize $-\log(q)$. We can, then, write an expression for the expectation value of this quantity over the distribution p as:

$$E[\log(1/q(x))] = \sum p(x) \log \frac{1}{q(x)} = \sum p(x) \log \frac{p(x)}{q(x)} - \sum p(x) \log p(x)$$

These quantities have a special meaning in information theory.

In addition to parameter estimation, I can do hypothesis testing on parameters and on models.

0.1 Statistical questions:hypothesis testing and model selection

0.2 Statistical questions:Decoding

0.3 Model Generalizations

0.4 Analysis Challenge

Exploratory data analysis–
- moments, mean, variance - comparisons: parametric models

0.5 Applications and Practical questions

(1) Estimate λ (2) Same different (3) Differential expression - confidence of answer vs sample number - parametric - non-parametric - parameterizing a distribution

(4) doing things with measurement noise–how does noise change the problem? (5) optimality

Here is data from a series of cancer patients. Lets do the following calculations in homework:

(1) differential expression on gene count distributions (2) Clustering samples using independent model

0.6 Inference and Sampling from distributions

Some notes: Things to learn. Sampling from distributions and methods.

Generation of data from your model.

Very important topic. Lets look back at what we did, we established a physical system where it's stationary distribution is $P(g)$. We can think of this as a machine for generating samples from $P(g)$. Think about this.

Generalization of this exact idea.

0.7 Public Data

- A couple papers on this: Raj, Regev,Islam
- Three data sets.
- data in the folder from different cell populations and pooled cells.

Getting the data

1 Linear algebra of gene expression data.

-kmeans