So, hi everyone. Since we’re still waiting for people to join, I will first give a brief introduction to Matthew here. First, it’s my honor to introduce Dr. Matthew Stevens as our seminar speaker today. And Matthew is a professor from human genetics and the statistics at University of Chicago. And in the past, his research mainly focused on developing new statistical methods for especially genetic applications. Including, for example, GWAS association studies and fine mapping and populations genetic variants. And today, he will give a talk on some recently developed empirical Baye methods for the estimation for normal mean models that will introduce the (indistinct) properties such as shrinkage, sparsity or smoothness. And he will also discuss how to apply these methods to a range of practical applications. Okay, so let’s wait for another minute and then I will hand it over to Matthew. So I will hand it over to Matthew from here. Let’s welcome him. Thank you very much. It’s a great pleasure to be here and to get the opportunity to present my work to you today. So I guess just a little bit of background.
So few years ago...

Well, I guess, I’ve been teaching sparsity and shrinkage for a while, and it struck me that, in practice, people don’t really use many of these ideas directly... At least not the empirical Bayse versions of these ideas directly in applications. And so, I’m wondering why that is. And partly, it’s the lack of user-friendly, convenient methods for applying these ideas. So I’ve been trying to think about how we can make these powerful ideas and methods more generally applicable or easily applicable in applications. These ideas have been around quite some time. But I think we’ve made some progress on actually just making them a bit simpler, maybe and simpler to apply in practice, so I’m gonna tell you about those today. Oh, sorry.

It’s not advancing, let me see. Okay, kind of related to that, the normal means problem is something we teach quite frequently. It’s not hard to teach but it always struck me whenever I was taught it that it looked like a kind of a toy model that statisticians kind of think up to teach students things.
But never actually use.
And then suddenly, I had an epiphany
and realized that it’s super useful.
And so now, I’m trying to...
I’m not the only one but I’m trying to convince people
that actually, this is a super useful thing
that we should be using in practice.
So here’s the normal means model.
The idea is that you’ve got
a bunch of observations, XJ,
that you can think of as noisy observations
of theta J
and they have some variance.
I’m going to allow each variance to be different.
The simplest version would be to assume
that the variances are all the same
but I’m going to allow them to be different.
But an important point is
that we’re going to assume that the variance is unknown,
which sounds a bit weird
but in applications, we’ll see that there are reasons
why we might think that’s an okay assumption
in some applications.
Okay, so the basic idea is
you’ve got a bunch of measurements
that are noisy measurements
of sum theta J
and they have known variance,
so they have known precision essentially,
and you want to estimate the Theta Js.
And, of course, the MLE is just to estimate see theta J by its corresponding measurement, XJ. And really, it was a big surprise, I think. I wasn’t around at the time but I believe it was a big surprise in 1956 when Stein showed that you can do better than the MLE, at least in terms of average squared error expected square there. And so, there are many different ways to motivate or... To motivate this result. And I think many of them end up not being that intuitive. It is quite a surprising result in generality but I think... So the way I like to think about the intuition for why this might be true, it’s not the only intuition but it’s one intuition for why this might be true, is to have an empirical Bayse thinking to the problem. And so, to illustrate this idea, I use a well-worn device, at this point, which is baseball batting averages. Efron certainly has used this example before to motivate empirical Bayse ideas. This particular example comes from... The data come from this block here, that I referenced at the bottom, which I quite like as an explanation of basic ideas behind empirical Bayse. So this histogram here shows a bunch
of basic baseball batting averages for a particular season in 1900. You don’t need to know very much about baseball to know what’s going on here. Essentially, in baseball, you go and try and hit a ball and your batting average is what proportion of the time you as a bat person end up hitting the ball. And a good baseball batting average is around 0.3 or so.

And in a professional baseball, no one’s really going to have a batting average of zero ’cause they wouldn’t survive. But empirically, there were some individuals in this season who had a batting average of zero, that is they completely failed to hit the ball every time they went up to bat. And there were some people who had a batting average of above 0.4, which is also completely unheard of in baseball. Nobody has a batting average that high, so what’s going on here? Well, it’s a simple explanation is that these individuals at the tails are individuals who just had a few at-bats. They only went and attempted to hit the ball a small number of times. And so, maybe these individuals only had two bats and they missed it both times, they got injured or they weren’t selected.
or, for whatever reason, they didn’t hit the ball many times…

They didn’t go to at bat many times and so, their batting average was empirically zero. Think of that as the maximum likelihood estimate. But if you wanted to predict what they would do, say next season, if you gave them more at bats in the long run, zero would be a bad estimate for obvious reasons.

And the same applies to these individuals up here with very big batting averages. They also had relatively few at-bats and they just happened to hit it above 0.4 of the time. And the individuals who had lots of at-bats are all in the middle here. So these are binomial observations, basically, and the ones who have small N are more likely to be in the tails. And the ones we’re big N are all going to be around in the middle here.

So what would we do? What would we want to do if we wanted to estimate, for example, for this individual, their batting average for next season? If we were gonna predict what they were gonna get. Well, we would definitely want to estimate something closer to the average batting average than 0.4.
That’s the intuition. And one way to frame that problem is that...
So sorry.
So this is the basic idea of shrinkage.
We would want to shrink these estimates towards, in this case, towards the mean.
So how are we gonna do that?
Well, one way to think about it is...
Sorry, let me just...
Yes.
Sorry, just getting my slides.
Okay, so here, the red line represents some underlying distribution of actual batting averages.
So conceptually, some distribution of actual batting averages among players.
And in fact, what we’ve done here is estimate that red line from the data.
That’s the empirical Bayse part of the empirical Bayse.
The empirical part of empirical Bayse is that the red line which we’re going to use as a prior for any given player was
0:09:30.23 –> 0:09:32.59 actually estimated from all the data.
0:09:32.59 –> 0:09:33.89 And the basic idea is
0:09:33.89 –> 0:09:36.26 because we know what the variance
0:09:36.26 –> 0:09:38.22 of a binomial distribution is,
0:09:38.22 –> 0:09:40.08 we can kind of estimate
0:09:40.08 –> 0:09:41.87 what the overall distribution
0:09:41.87 –> 0:09:46.159 of the underlying piece in this binomial look like,
0:09:46.159 –> 0:09:49.12 taking account of the fact that the histogram is
0:09:49.12 –> 0:09:53.37 a noisy observations of that underlying P.
0:09:53.37 –> 0:09:54.203 Every bat...
0:09:54.203 –> 0:09:57.62 Basically, every every estimated batting average is
0:09:57.62 –> 0:09:59.98 a noisy estimate of the true batting average
0:09:59.98 –> 0:10:00.92 with the noise depending on
0:10:00.92 –> 0:10:02.98 how many at-bats they have.
0:10:02.98 –> 0:10:05.133 So once we’ve estimated that red line,
0:10:05.133 –> 0:10:07.58 that prior,
0:10:07.58 –> 0:10:09.96 we can compute the posterior
0:10:09.96 –> 0:10:12.35 for each individual based on that prior.
0:10:12.35 –> 0:10:13.183 And when we do that,
0:10:13.183 –> 0:10:16.11 this is a histogram of the posterior means.
0:10:16.11 –> 0:10:17.71 So these are, if you like,
0:10:17.71 –> 0:10:20.34 shrunken estimates of the batting average
0:10:20.34 –> 0:10:21.173 for each individual.
0:10:21.173 –> 0:10:22.25 And you can see that the individuals
0:10:22.25 –> 0:10:24.9 who had zero at-bats got shrunk
0:10:24.9 –> 0:10:27.12 all the way over somewhere here.
0:10:27.12 –> 0:10:29.34 And that’s because their data really...
0:10:29.34 –> 0:10:31.8 Although, the point estimate was zero,
0:10:31.8 –> 0:10:32.83 they had very few at bats.
0:10:32.83 –> 0:10:37.83 So the information in that data are very slim,
0:10:38.01 –> 0:10:39.79 very little information.
And so, the prior dominates when you're looking at the posterior distribution for these individuals. Whereas individuals in the middle who have more at-bats, will have the estimate that is less shrunken. So that's gonna be a theme we'll come back to later. So how do we form... That's a picture. How do we formulate that? So those were binomial data, I'm gonna talk about normal data. So don't get confused by that. I'm just going to assume normality could do the same thing for a binomial, but I think the normals a more generally useful and convenient way to go. So here's a normal means model again and the idea is that that we're going to assume that thetas come from some prior distribution, G, that was the red line in my example, and we're going to estimate G by maximum likelihood essentially. by maximum likelihood essentially. So we're going to use all the X's, integrating out theta to obtain a maximum likelihood estimate for G. That's stage one, that's estimating that red line. And then stage two is to compute the posterior distribution.
for each batting average,

or whatever theta J we’re interested in,

taking into account that estimated prior and the data on the individual J.

So that’s the formalization of these ideas.

And these posterior distributions are gonna be shrunk towards the prior or the primary.

So I guess I’ve left unspecified here,

what family of priors should we consider for G?

So a commonly used prior distribution is this so-called point-normal,

or sometimes called spike and slab prior distribution.

And these are...

Sorry, I should say,

I’m going to be thinking a lot about problems where we want to induce sparsity.

So in baseball, we were shrinking towards the mean but in many applications,

the natural point towards natural prime mean, if you like, is zero in situations where we expect effects to be sparse, for example.

So I’m gonna be talking mostly about that situation,

although the ideas are more general.

And so, I’m going to be focusing on the sparsity inducing choices of prior family.

And so, one commonly used one is this point normal where there’s some mass pi zero exactly at zero,

and then the rest of the mass is
normally distributed about zero.

So the commonly used one.

In fact, it turns out,

and this is kind of interesting I think,

that it can be easier to do the computations

for more general families.

So for example,

just take the non-parametric family

that’s the zero-centered scale mixture of normal,

so we’ll see that in it,

which includes all these distributions of special cases.

It’s nonparametric.

It includes a point-normal here.

It also includes the T-distribution,

the Laplace distribution,

the horseshoe prior, if you’ve come across that,

this zero-centered scale mixture of normals

and the surprise is that it turns out

to be easier, in some sense,

do the calculations for this family,

more general family,

than this narrow family,

partly because of the convex family.

So you can think of this as a kind of a convex relaxation

of the problem.

So all the computations become...

The optimizations you have to do in the simplest case

become convex when you use this family.

So let me say a bit more about that

for the non-parametric.

11
How do we actually do these non-parametric computations?

Well, we actually approximate the non-parametric computation using a grid idea. So here’s the idea.

We modeled G, our prior, as a mixture of… I like to think of this K as being big. A large number of normal distributions.

All of these normal distributions are centered at zero, that’s this zero here, and they have a different variance. Some of them have very small variances, perhaps even one of them has a zero variance, so that’s the point mass at zero.

And the variance is sigma… Think of Sigma squared K getting gradually bigger until the last Sigma squared K is very big.

So we’re just gonna use a lot of them. Think of K as being, let’s say 100 or 1,000 for the...

In practice, we find 20 is enough but just think of it as being big and spanning a lot of different variances, going from very, very small, to very, very big.

And then, estimating G just comes down to estimating these pis,

these mixture proportions. And that, then of course, is a finite dimensional optimization problem.
and in the normal means model, it’s a convex problem, well actually, for any mixture, W ell actually, for any mixture, it’s a convex problem, and so there are efficient ways to find the MLE for pi, given the grid of variances. So let’s just illustrate what’s going on here. Here’s a grid of just three normals. The one in the middle has the smallest variance, the one over here has the biggest variance. And we can get a mixture of those, looks like that. So you can see this is kind of a spiky distribution but also with a long tail, even with just a mixture of three distributions. And so, the idea is that you can get quite a flex... It’s a flexible family by using a larger number of variances than three. You can imagine you can get distributions that have all sorts of spikiness and long-tailed behavior. So maybe just to fill in the details here; with that prior as a mixture of normals, the marginal distribution, P of X, integrating out theta is analytic because the sum of normals is normal. So if you have a normally distributed variable and then you have another variable that’s a normal error on top of that, you get a normal.
So the marginal is a mixture of normals that's very simple to work with and estimating pi is a convex optimization problem. You can do it.

You can do an EM algorithm but convex methods, as pointed out by Koenker and Mizera, can be a lot more reliable and faster.

Okay, so let's just illustrate those ideas again.

Here's a potential prior distribution and here's a likelihood. So this is like a likelihood from a normal... This is an estimate... Think of this as a likelihood for theta J in a normal means model. So maybe XJ was one and a half or something and SJ was, I don’t know, something like a half or something or a half squared. So this is meant to represent the likelihood. So what does the posterior look like when we combine this prior, the black line, with this likelihood, the red line? It looks like this green line here. So what you can see is going on here is that you get shrinkage towards the mean, right? But because the black line is long-tailed because of the prior in this case has a long tail, and because the red line... The likelihood lies quite a ways in the tail,
the spiky bit at zero doesn’t have very much impact because it’s completely inconsistent with the data. Zero is basically inconsistent with the data and so the posterior looks approximately normal. It’s actually a mixture of normals but it looks approximately normal ’cause of weight and there, zero is very, very small. Whereas if a... Here’s a different example, the black line is covered by the green line this time because it’s... So I plotted all three lines on the same plot here. The black line is... Think of it as pretty much the green line. It’s still the same spiky prior. but now the likelihood is much flatter. The XJ is the same. Actually, it’s one and a half. but we have an SJ that’s much bigger. So what happens here is that the prior dominates because the likelihood’s relatively flat. and so the posterior looks pretty much like the prior. and you get very strong shrinkage. So think of this as corresponding to those individuals who had very few at-bats, their data are very imprecise, and so their posterior, the green line, looks very like the prior, the black line. Okay, so we’re gonna shrink those observations more. So the key point here, I guess,
is that the observations with larger standard error, larger SJ, get shrunk more. I should say “larger standard deviation” get shrunk more. Here’s another intermediate example where the red line... The likelihood’s kind of not quite enough. It illustrates the idea that the posterior could be bimodal because the prior and the likelihood are indifferent, have weight in different places. So you can get different kinds of shrinkage depending on how spiky the prior is, how long-tailed the prior is, how flat the likelihood is etc. So obviously the shrinkage, the amount of shrinkage you get, depends on the prior, G, which you’re gonna estimate from the data. It also depends on the standard error or the standard deviation, SJ. And one way to summarize this kind of the behavior, the shrinkage behavior, is to focus on how the posterior mean changes with X. So we can define this operator here, as the X posterior mean of theta J, given the prior and its variance or standard deviation and that we observed XJ is equal to X. I’m gonna call this the shrinkage operator
for the prior, G,
and variance, S for standard deviation, S.
Okay, so we could just plot
some of these shrinkage operators.
So the idea here is...
Sorry, this slide has B instead of X.
Sometimes I use B and sometimes I use X.
I've got them mixed up here, sorry.
So think of this as X
and this is S of X.
So these different lines here correspond
to different priors.
So the idea is that by using different priors,
we can get different types of shrinkage behavior.
So this prior here shrinks very strongly to zero.
This green line shrinks very strongly to zero
until B exceeds some value around five,
where it hardly shrinks at all.
So this is kind of a prior that has
kind of a big spike near zero.
But also a long tail,
such that when you get far enough in the tail,
you start to be convinced
that there's a real signal here.
So you can think of that
as this kind of...
This is sometimes called...
This is local shrinkage
and this is global.
So you get very strong local shrinkage towards zero
but very little shrinkage
0:21:30.86 –> 0:21:32.6 if the signal is strong enough.
0:21:32.6 –> 0:21:34.09 That kind of thing.
0:21:34.09 –> 0:21:35.18 But the real point here is that
0:21:35.18 –> 0:21:36.98 by using different priors,
0:21:36.98 –> 0:21:39.44 these different scale mixture of normal priors,
0:21:39.44 –> 0:21:43.46 you can get very different looking shrinkage behaviors.
0:21:43.46 –> 0:21:45.39 Ones that shrink very strongly to zero
0:21:45.39 –> 0:21:46.42 and then stop shrinking
0:21:46.42 –> 0:21:50.313 or ones that shrink a little bit all the way, etc.
0:21:51.91 –> 0:21:54.61 And so, if you’re familiar with other ways
0:21:54.61 –> 0:21:56.27 of doing shrinkage analysis,
0:21:56.27 –> 0:21:57.8 and this is one of them,
0:21:57.8 –> 0:21:59.01 or shrinkage,
0:21:59.01 –> 0:22:00.96 is to use a penalized likelihood.
0:22:00.96 –> 0:22:03.72 Then you can try and draw a parallel
0:22:03.72 –> 0:22:04.76 and that’s what I’m trying to do here.
0:22:04.76 –> 0:22:07.5 Draw a parallel between the Bayesian method
0:22:07.5 –> 0:22:10.8 and the penalized likelihood-based approaches.
0:22:10.8 –> 0:22:12.943 to inducing shrinkage or sparsity.
0:22:14.69 –> 0:22:18.27 Another way to induce shrinkage is to essentially...
0:22:18.27 –> 0:22:20.88 This is the kind of normal log likelihood here
0:22:20.88 –> 0:22:23.47 and this is a penalty here
0:22:23.47 –> 0:22:24.96 that you add for this.
0:22:24.96 –> 0:22:26.12 This could be an L1 penalty
0:22:26.12 –> 0:22:28.37 or an L2 penalty or an L0 penalty,
0:22:28.37 –> 0:22:30.17 or some other kind of penalty.
0:22:30.17 –> 0:22:32.172 So there’s a penalty function here.
0:22:32.172 –> 0:22:34.24 And you define the estimate
0:22:34.24 –> 0:22:35.81 as the value that minimizes
0:22:35.81 –> 0:22:37.851 this penalized log likelihood.
0:22:37.851 –> 0:22:40.52 Sorry, yeah, this is a negative log likelihood.
Penalized least squares, I guess this would be. Okay, so now eight is a penalty function here and Lambda is a tuning parameter that says how strong, in some sense, the penalty is. And these are also widely used to induce shrinkage, especially in regression contexts. And so, here are some commonly used shrinkage operators, corresponding to different penalty functions. So this green line is what’s called the hard thresholding, which corresponds to an L0 penalty. If you don’t know what that means, don’t worry. But if you do, you make that connection. At the red line here is L1 penalty. And these two other ones here are particular instances of some non-convex penalties that are used in regression context, particularly in practice. And I guess that the point here is that, essentially, different prior distributions in the normal means model can lead to shrinkage operators, shrinkage behavior that looks kind of similar to each of these different types of penalty. So you can’t actually mimic the behavior exactly. I've just… Or actually, my student, (indistinct) Kim, chose the priors to visually closely match these
but you can’t get…

Some of these have kinks and stuff that you can’t actually, formally, exactly mimic but you can get qualitatively similar shrinkage behavior from different priors as different penalty functions. So you should think about the different priors as being analogous to different penalty functions. And so, the key… How does EB, empirical Bayse shrinkage, differ from, say, these kinds of penalty-based approaches, which I should say are maybe more widely used in practice? Well, so shrinkage is determined by the prior, G, which we estimate in an empirical Bayse context by maximum likelihood. Whereas in typical shrinkage… Sorry, typical penalty-based analyses, people use cross validation to estimate parameters. And the result is that cross-validation is fine for estimating one parameter but it becomes quite cumbersome to estimate two parameters, and really tricky to estimate three or four parameters ’cause you have to go and do a grid of different values and do a lot of cross-validations and start estimating all these different parameters. So the point here is really because we estimate G by maximum likelihood,
we can actually have a much more flexible family in practice that we can optimize over more easily. It's very flexible,
you can mimic a range of penalty functions so you don't have to choose whether to use L1 or L2 or L0.
You can essentially estimate over these non-parametric prior families.
Think of that as kind of deciding automatically whether to use L0, L1, L2 or some kind of non-convex penalty, or something in between.
And the posterior distribution, of course then, another nice thing is that it gives not only the point estimates but, if you like, it also gives shrunken interval estimates which are not yielded by a penalty-based approach.
So I guess I'm trying to say that there are potential advantages of the empirical Bayes approach over the penalty-based approach.
And yeah, although I think, people have tried, particularly Efron has highlighted the potential for empirical Bayes
the potential for empirical Bayes to be used in practical applications, largely in the practical application.
So I've seen empirical Bayes shrinkage hasn't really been used very, very much.
So that's the goal,
is to change that.
So before I talk about examples, I guess I will pause for a moment to see if there are any questions. And I can’t see the chat for some reason so if anyone… So please unmute yourself if you have a question.

Okay, thank you. It’s all clear.

I’m happy to go on but I just wanna...

Okay, so we’ve been trying to think about how to use these ideas, make these ideas useful in practice for a range of practical applications. We’ve done work on multiple testing, on high dimensional linear aggression, and also some on matrix factorization. From previous experience, I’ll probably get time to talk about the first two and maybe not the last one, but there’s a pre-print on the archive. You can see if you’re interested in matrix factorization. Maybe I’ll get to get to talk about that briefly.

But let me talk about multiple testing first. So the typical multiple testing setup,
where you might typically, say, apply a Benjamini-Hochberg type procedure is you’ve got a large number of tests, So J equals one to N, and test J yields a P value, PJ, and then you reject all tests with some PJ less than a threshold gamma, where that threshold is chosen to control the FDR in a frequented sense. So that’s the typical setup. So how are we going to apply the normal means model to this problem? Okay, well, in many applications, not all but in many, the P values are derived from some kind of effect size estimate, which I’m going to call “Beta hat J,” which have standard errors, SJ, that satisfy approximately, at least, that Beta J hat is normally distributed about the true value Beta J with some variance given it by SJ. So in a lot... I work a lot in genetic applications. So in genetic applications, we’re looking at different genes here. So Beta J hat might be the estimate of the difference in expression, let’s say, of a gene, J, between, say, males and females. And Beta J would be the true difference at that gene.
And you’re interested in identifying which genes are truly different… Have a different mean expression between males and females here. And the reason that SJ is approximately known is because you’ve got multiple males and multiple females that you’re using to estimate this difference. And so, you get an estimated standard error of that Beta hat as well. And so, once you’ve set the problem up like this, it looks suddenly like a normal means problem and we can kind of apply the empirical Bayes normal means idea. We’re gonna put a prior on the Beta Js that is sparsity inducing. That is, it’s kind of centered at zero, maybe it’s got a point mass at zero. But we’re gonna estimate that prior from the data. Okay. And so, not only can you get posterior means for Beta, as I said, you can get posterior interval estimates. So you can kind of do things like compute the posterior in 90% credible interval, given that prior and the likelihood for each Beta J. and we could reject, for example, if the interval does not contain zero. And I’m not going to talk about this in detail because the details are in...
a biostatistics paper from 2017.

I should say that the idea of using empirical Bayes for FDR actually dates back to before Benjamini and Hochberg.

Duncan Thomas has a really nice paper that was pointed out to me by John Witty that actually contains these basic ideas but not nice software implementation, which maybe explains why it hasn’t caught on in practice yet.

Duncan Thomas has a really nice paper that was pointed out to me by John Witty that actually contains these basic ideas but not nice software implementation, which maybe explains why it hasn’t caught on in practice yet.

Efron’s also been a pioneer in this area.

So... Okay, so I don’t want to dwell on that because, actually, I think I’ll just summarize what I think is true compared with Benjamini-Hochberg.

You get a bit of an increase in power by using an empirical Bayes approach.

The Benjamini-Hochberg approach is more robust to correlated tests though, so the empirical Bayes normal means model does assume that the tests are independent.

And, in practice, we have seen that correlations can cause problems.

If you’re interested in that, I have a pre-print with Lei Sun on my website. But the empirical Bayes normal means also provides these interval estimates, which is kind of nice.

Benjamini-Hochberg does not.
So there are some advantages of the empirical Bayes approach and maybe some disadvantages compared with Benjamini-Hochberg. But I think that the real benefit actually comes when we look at multi-variate extensions of this idea. So I just wanted to briefly highlight those and spend some time on those.

And now, my Beta Js are a vector of observation. So I think of this as measuring, say, gene J in multiple different tissues. In fact, we’ve got 50 different tissues in the example. We’re gonna assume that the estimates are...
normally distributed with mean, the true values and some variance, covariance matrix now, which we’re going to assume, for now, is known. That’s actually a little trickier but I’m gonna gloss over that for… If you want to see details, take a look at the paper. I just wanna get the essence of the idea across. We’re still going to assume that Beta J comes from some prior, G, and we’re still gonna use a mixture of normals, but now we’re using a mixture of multi-variate normals. And unlike the univariate case, we can’t use a grid of… We can’t use a grid of values that span all possible covariance matrices here. It’s just too much. So we have to do something to estimate these covariance matrices, as well as estimate the pis. And again, if you want to see the details, take a look at a Urbut et al. But let me just illustrate the idea of what’s going on here, or what happens when you apply this method to some data. So this is… I said 50, we have 44 tissues in this particular example.
So each row here is a tissue. These yellow ones here are brain tissues, different brain tissues, and I think we’ll see one later that’s blood. I think this one might be blood. Anyway, each one is a tissue; lung, blood, etc. You don’t need to know which ones are which, for now.

And so, what we’ve done here is plot the Beta hat and plus or minus two standard deviations for each tissue at a particular... In this case, a particular snip, actually (indistinct). So this is an eQTL analysis, for those of you who know what that means. If you don’t, don’t worry about it. Just think of it as having an estimated effect plus or minus two standard deviations in 44 different tissues, and we want to know which ones are quote, “significantly different from zero.” And so what happens...

Didn’t expect that to happen. Sorry, okay. Yeah, these are just... These are just two examples. So this is one example, here’s another example where we’ve done the same thing.
Estimated effects, plus or minus two standard deviations.

So what you can see in this first one is that it looks like, at least, the brain tissues have some kind of effect. That’s what you’re supposed to see here. And maybe there are some effects in other tissues. There’s a tendency for effects to be positive, which might suggest that maybe everything has a small effect to everywhere, but particularly strong in the brain. And whereas in this example, this one appears to have an effect in just one tissue. This is the blood actually.

So this is an effect in blood but mostly, it doesn’t look like there’s an effect in other tissues. But these, just to emphasize, these are the raw data, the Beta hats and the standard errors. There’s no shrinkage occurred yet. But the idea is that the empirical Bayse approach takes all these examples, examples like this and examples like this, to learn about what kinds of patterns are present in the data. That is, “What does G look like?” So it learns from these examples
that there are some effects that look like they’re shared among the brain tissues, and there are some effects that are... These are actually somewhat rare but rarely, there’s an effect that’s specific to one tissue like, in this case, blood. And it also learns, in this case actually, that there’s a lot of null things, because there are a lot of null things as well. So it puts lots of mass on the null as well and that causes the shrinkage. And then, having estimated those patterns from the data, it computes posteriors. And so, here’s the data and then the posterior intervals for the same... For that first example. And what you can see is that because of the combining information across tissues, you get standard errors that are getting smaller, the brain estimates all get shrunk towards one another, and all these... There’s some borrowing strength of information, borrowing information across these tissues, to make these these look like some of them are kind of borderline significant. Now, it looks like there’s probably an effect in every tissue but a much stronger effect in brain. Whereas this example here,
it recognizes that this looks like an effect that’s specific to blood. And so, it shrinks everything else strongly towards zero because it knows that most things are null, it’s learned that from the data, but the blood estimate gets hardly shrunk at all. We saw that kind of behavior where things that are near zero can get shrunk towards zero, whereas other things that are far away don’t get shrunk as much. And it’s really hard to do that kind of thing without doing some kind of model-based analysis, doing Benjamini-Hochberg type art non-model based without making any assumptions or making minimal assumptions, very hard to capture this kind of thing, I think. So I think the empirical Bayse approach has big advantages in this setting. I’ll pause before I talk about regression. Any questions there? So Matthew, I have some basic questions. So in your means multivariate multiple testing case, I guess for each of the plot, you are looking at maybe a particular genes influence on some... - Good, yeah. Sorry, I did skip over it a bit. So these are eQTLs. So actually, what I’m plotting here is each... This is a single snip associated with a single gene. And this is it’s,
"How associated is this snip with this genes expression level in the different brain tissues, in the blood tissue in lung and spleen, etc?"

The idea is that the scientific goal is to understand which genetic variants are impacting gene expression in different tissues, which might tell us something about the biology of the tissues and the regulation going on in the different tissues.

So in this case, I don’t think I fully understand why it’s multivariate multiple tests, not univariate because you are looking at each gene versus each snip. Right, so sorry.

Think of J indexing eQTL. So we’ve got 2 million potential eQTLs, so that’s the multiple part of it. For 2 million potential eQTLs, that’s... And then each eQTL has data on 44 tissues, so that’s the multi-variate part of it. Speaking over each other. If you thought about it in terms of say P values or maybe Z scores, you have a matrix of Z scores. There are two million rows and there are 44 columns. You have a Z score or a P value.
0:40:04.736 -> 0:40:08.51 for each element in that matrix,
0:40:08.51 -> 0:40:11.242 and what we’re assuming is that
0:40:11.242 -> 0:40:12.84 the rows are independent,
0:40:12.84 -> 0:40:14.97 which is not quite true but still,
0:40:14.97 -> 0:40:16.515 we’re assuming that the rows are independent
0:40:16.515 -> 0:40:17.792 and the columns,
0:40:17.792 -> 0:40:20.11 we’re assuming that they can be correlated.
0:40:20.11 -> 0:40:20.943 And in particular,
0:40:20.943 -> 0:40:22.259 we’re assuming that the...
0:40:22.259 -> 0:40:24.33 Well, we’re assuming that both
0:40:24.33 -> 0:40:25.91 the measurements can be correlated,
0:40:25.91 -> 0:40:27.3 so it’s V,
0:40:27.3 -> 0:40:29.84 but also that the effects can be correlated.
0:40:29.84 -> 0:40:31.36 So that’s to capture the idea
0:40:31.36 -> 0:40:32.55 that there might be some effects
0:40:32.55 -> 0:40:35.77 that are shared between say brain tissues–
0:40:35.77 -> 0:40:36.603 - I see.
0:40:36.603 -> 0:40:37.93 I see.
0:40:37.93 -> 0:40:39.74 So this multi-variate is different
0:40:39.74 -> 0:40:42.99 from our usual notion where the multivariate
0:40:42.99 -> 0:40:43.95 and multivariate snip.
0:40:43.95 -> 0:40:46.03 So there’s multivariate tissue.
0:40:46.03 -> 0:40:49.464 I guess, are the samples from the same cohort?
0:40:49.464 -> 0:40:51.86 - Yeah, so in this particular case,
0:40:51.86 -> 0:40:54.83 the samples are from the same individuals.
0:40:54.83 -> 0:40:56.18 So these different brain tissues...
0:40:56.18 -> 0:40:57.361 There’s overlap anyway, let’s say.
0:40:57.361 -> 0:40:59.92 And so, that’s what causes this...
0:40:59.92 -> 0:41:02.08 That causes headaches, actually, for this–
0:41:02.08 -> 0:41:04.063 - Okay, got it, thanks. - Yeah.
0:41:04.063 -> 0:41:05.3 Just to emphasize,
it doesn’t have to be different tissues.
The whole method works on any matrix of Z scores, basically.
As long as you think that the rows correspond to different tests and the columns correspond to different, say, conditions for the same test. Examples might be looking at the same snip across lots of different phenotypes, so looking at schizophrenia, looking at bipolar, looking at different diseases or different traits, and you can have a Beta hat for that snip and a standard error for that snip in every trait. And you could try to learn, "Oh look, there are some traits that tend to share effects and other traits that don’t," or, often in experiments, people treat their samples with different treatments. They challenge them with different viruses. They look to see which things are being changed when you challenge a cell with different viruses or different heat shock treatments or any kind of different treatment. So yeah, basically, the idea is very generic. The idea is if you’ve got a matrix of Z scores
where the effect, say, look likely to be shared among column sometimes and the rows are gonna be approximately independent, or at least you’re willing to assume that, then you can apply the method. Okay, got it, thanks. So, actually, that’s an important kind of… Also, something that I’ve been thinking about a lot is the benefits of modular or generic methods. So if you think about what methods are applied in statistics a lot, you think T-test, linear regression. These are all kind of very generic ideas. They don’t… And Benjamini-Hochberg. The nice thing about Benjamini-Hochberg is you just need a set of P values and you can apply Benjamini-Hochberg. You don’t have to worry too much about where those P values came from. So I think, for applications, it’s really useful to try to think about what’s the simplest type of data you could imagine inputting into the procedure in order to output something useful? And sometimes, that involves making compromises because to make a procedure generic enough, you have to compromise on what… On maybe what the details of what are going in. So here, what we’ve compromised on is that we take a matrix of Z scores,
or potentially Beta hats and their standard errors, we can do either, and that’s the input. So that makes it relatively generic. You don’t have to worry too much about whether those Beta hats and the standard errors, or the Z scores, are coming from logistic regression or linear regression, or whether that controlling for some covariance or all sorts of… From a mixed model, etc. As long as they have the basic property that the Beta hat is normally distributed about the true Beta with some variance that you are willing to estimate, then you can go. Sorry, (indistinct). A short question. So in practice, how you choose… How many mix… How many distribution you want to mixture like the (indistinct)? - (indistinct) Yeah, so great question. And my answer, generally, is just use as many as you want. So as many as you can stomach. The more you use, the slower it is. And so, you might worry about over-fitting, but it turns out that these procedures are very robust to over-fitting
because of this fact that the mean is fixed at zero.

So all the components have a mean zero and have some covariance and because of that, they have limited flexibility to overfit. They’re just not that flexible. And in the univariate case, that’s even more obvious, I think. That in the univariate case, every one of those distributions, any mixture of normals that are all centered at zero is unimodal at zero and has limited... Can’t have wiggly distributions that are very spiky and overfitting. So these methods are relatively immune to overfitting in practice. If you’re worried about that, you can do a test-train type thing where you use half your tests to train, and then you look at the log likelihood out of sample on others, and then tweak the number to avoid overfitting. And we did do that early on in the methods but we don’t do it very often now, or we only do it now when we’re worried ’cause generally it seems like overfitting doesn’t seem to be a problem, but if we see results are a little bit weird or a bit concerning, we try it to make sure we’re not overfitting.
Okay, thank you. I should say that, in the paper, we kind of outlined some procedures we use for estimating these variance, co-variance matrices but they’re not like... They’re kind of like... The whole philosophy is that we could probably do better and we’re continuing to try and work on better methods for estimating this as we go forward. So we’re continually improving the ways we can estimate this. Okay, so briefly, I’ll talk about linear regression. So here’s your standard linear regression where, so we’ve N observations, X is the matrix of covariates here, B are the regression coefficients. I’m kind of thinking of P as being big, potentially here. And the errors normal. And so, the empirical Bayes idea would be to assume that the Bs come from some prior distribution, G, which comes from some family, curly G. And what we’d like to do is estimate G and then shrink the estimates of B, using empirical Bayse type ideas and posterior count computations. But it’s not a simple normal means model here, so how do we end up applying...
the empirical Bayse methods to this problem?

Well, let’s just... I’m gonna explain our algorithm by analogy with penalized regression algorithms because the algorithm ends up looking very similar, and then I'll tell you what the algorithm is actually kind of doing.

So a penalized regression would solve this problem. So if you've seen the Lasso before... I hope many of you might have. If you’ve seen the Lasso before, this would be solving this problem with $H$ being the L1 penalty, absolute value of $B$, right?

There are many, many algorithms to solve this problem but a very simple one is coordinate ascent. So essentially, for each coordinate, it just iterates the following. For each coordinate,
you have some kind of current estimate for $B$s.

So what do here is you form the residuals by taking away the effects of all the $B$s except the one you’re trying to update, the one you’re trying to estimate.

So $X$ minus $J$ here is all the covariates except covariate $J$. $B$ minus $J$ is all the corresponding coefficients. So this is the residual.
RJ is the residual, after removing all the current estimated effects apart from the Jth one. Then you basically compute an estimate of the Jth effect by regressing those residuals on XJ. And then you shrink that using a shrinkage operator that we saw earlier. Just to remind you that a shrinkage operator is the one that minimizes this penalized least squares problem. And it turns out, it’s not hard to show that this is coordinate ascent for minimizing this penalized objective function. And so every iteration of this increases that objective function or decreases it.

Okay, so it turns… So our algorithm looks very similar. You still compute the residuals, you compute a Beta hat you compute a standard error, which is familiar form. And then you, instead of shrinking using that penalized regression operator, you use a… Sorry, I should say, this is assuming G is known. I’m starting with G.
G is known.

So you can shrink...

Instead of using the penalty-based method,

you use the posterior mean shrinkage operator here

that I introduced earlier.

So it’s basically exactly the same algorithm,

except replacing this penalty-based shrinkage operator

with an empirical Bayse

or a Bayesean shrinkage operator.

And so, you could ask what that’s doing

and it turns out that what it’s doing is

it’s minimizing the Kullback-Leibler Divergence

between some approximate posterior, Q,

and the true posterior, P, here

under the constraint that this Q is factorized.

So this is what’s called

a variational approximation

or a mean-field,

or fully factorized variational approximation.

If you’ve seen that before,

you’ll know what’s going on here.

If you haven’t seen it before,

it’s trying to find an approximation

to the posterior.

This is the true posterior,

it’s trying to find an approximation

to that posterior that minimizes

the Kullbert-Leibler Divergence

between the approximation

and the true value under in a simplifying assumption

that the posterior factorizes,
which, of course, it doesn’t, so that’s why it’s an approximation. So that algorithm I just said is a coordinate ascent algorithm for maximizing F or minimizing the KL divergence. So every iteration of that algorithm gets a better estimate estimate of the posterior, essentially. Just to outline and just to give you the intuition for how you could maybe estimate G, this isn’t actually quite what we do so the details get a bit more complicated, but just to give you an intuition for how you might think that you can estimate G; Every iteration of this algorithm computes a B hat and a corresponding standard error, so you could imagine storing these steps here, you could imagine storing these through the iterations and, at the end, you could apply the empirical Bayes normal means procedure to estimate G from these B hats and standard errors, and something close to that kind of works. The details are a bit more complicated than that. So let me give you some kind of intuition for what we’re trying to achieve here based on simulation results. So these are some simulations we’ve done.
The covariates are all independent here.

The true prior is a point normal, that means that most of the effects are zero.

Well, actually maybe here, one of the effects is nonzero, five of the effects is nonzero, 50 of the effects are nonzero, and 500 of the effects of nonzero. And actually, there are 500 effects.

So the X-axis here just shows the number of non-zero coordinates and the results I’ve shown here are the prediction error, so we’re focusing on prediction error, the out of sample prediction error, using three different penalty-based approaches. The Lasso, which is this line, the L0Learn, which is this line, which is L0 zero penalty, and Ridge, which is this penalty, the L2 penalty.

So the important thing to know is that the L0 penalty is really designed, if you like, to do well under very sparse models. So that’s why it’s got the lowest prediction error when the model is very sparse, but when the model is completely densed, it does very poorly. Whereas Ridge is designed much more to... It’s actually based on a prior
that the effects are normally distributed. So it’s much better at dense models than sparse models.

And you can see that at least relative to L0Learn, Ridge is much better for the dense case but also much worse for the sparse case.

And then Lasso has some kind of ability to deal with both scenarios, but it’s not quite as good as the L0 penalty when things are very sparse, and it’s not quite as good as the Ridge penalty when things are very dense.

So our goal is that by learning the prior G from the data, we can adapt to each of these scenarios and get performance close to L0Learn when the truth is sparse and get performance close to Ridge regression when the truth is dense.

And so, the red line here actually shows the performance of our method. And you can see, indeed, we do even slightly better than L0Learn in this part here and slightly better than cross-validated Ridge regression in this here.

The difference between these two is just that the Ridge regression is doing cross-validation to estimate the tuning parameter and we’re using empirical Bayes maximum likelihood.
0:54:27.65 –> 0:54:28.483 to estimate it.
0:54:28.483 –> 0:54:29.96 So that’s just that difference there.
0:54:29.96 –> 0:54:31.927 And the Oracle here is using the true...
0:54:31.927 –> 0:54:34.7 You can do the Oracle computation
0:54:34.7 –> 0:54:35.61 for the Ridge regression
0:54:35.61 –> 0:54:37.603 with the true tuning parameter here.
0:54:37.603 –> 0:54:41.598 I should say that may be that this...
0:54:41.598 –> 0:54:45.04 Maybe I should just show you the results.
0:54:45.04 –> 0:54:47.01 So here is a bunch of other penalties,
0:54:47.01 –> 0:54:49.109 including elastic net, for example, you might wonder,
0:54:49.109 –> 0:54:50.98 which is kind of a compromise
0:54:50.98 –> 0:54:52.29 between L1 and L2.
0:54:52.29 –> 0:54:55.79 And you can see, it does do the compromising
0:54:55.79 –> 0:54:56.9 but it doesn’t do as well
0:54:56.9 –> 0:54:59.17 as the empirical Bayse approach.
0:54:59.17 –> 0:55:01.38 And here are some other non-convex methods
0:55:01.38 –> 0:55:03.7 that are more, again...
0:55:03.7 –> 0:55:06.11 They’re kind of more tuned to the sparse case
0:55:06.11 –> 0:55:07.313 than to the dense case.
0:55:07.313 –> 0:55:11.98 As promised, I’m gonna skip over
0:55:11.98 –> 0:55:14.03 the matrix factorization
0:55:14.03 –> 0:55:18.32 and just summarize to give time for questions.
0:55:18.32 –> 0:55:20.9 So the summary is that
0:55:20.9 –> 0:55:23.13 the empirical Bayse normal means model provides
0:55:23.13 –> 0:55:24.74 a flexible and convenient way
0:55:27.978 –> 0:55:31.78 And we’ve been spending a lot of time trying
0:55:31.78 –> 0:55:32.875 to apply these methods
0:55:32.875 –> 0:55:34.61 and provide software to do
0:55:34.61 –> 0:55:36.21 some of these different things.
And there's a bunch of things on my publications page and if you're interested in, if you can’t find what you’re looking for, just let me know, I’d be happy to point you to it. Thanks very much.

Thanks Matthew, that’s a great talk. I wonder whether the audience have any questions for Matthew.

So I do have some questions for you. So I think I really like the idea of applying empirical Bayes to a lot of applications and it’s really seems empirical Bayes has great success. But I do have a question or some doubt about the inference part, especially in that linear regression model.

So currently, for the current work you have been doing, you essentially shrink each of the co-efficient that based on, essentially, their estimated value, but in some applications, such as a GWAS study or fine mapping, different snips can have very different LD score structure. So in this case, how much we can trust the inference, the P value, from this (indistinct)? So, great question. So let me just first... First emphasize that the shrink... The estimate here is being done removing the effects,
The estimated effects of the other variables. So each iteration of this, when you’re estimating the effect of snip J, in your example, you’re taking the estimated effects of the other variables into account. So the LD structure, as you mentioned, that’s the correlation structure between the Xs is formerly taken into account. However, there is a problem with this approach for very highly correlated variables. So let’s just suppose there are two variables that are completely correlated, what does this algorithm end up doing? It ends up basically choosing one of them and ignoring the other. The Lasso does the same in fact. So it ends up choosing one of them and ignoring the other. And if you look to the posterior distribution on its effect, it would be far too confident in the size of the effect. because it would assume that the other one had zero effect. And so it would have a small credible
And so, we’ve worked recently on a method for doing that. A different method, different work than what I’ve just described here for doing fine mapping using variational approximations that don’t have this problem, and it’s on my webpage. It’s Wang et al in JRSS-B, just recently, this year. 2021, I guess. That’s awesome. Thanks, so any more question for Matthew from the audience?

Okay, I think we’re of running out of time also. So if you have any question about the stuff to (indistinct) you want to use, I think you can contact either the authors of the paper or Matthew off the line. And thank you again for agreeing to present your work here. It’s looks really useful and interesting. Thank you for having me.