Good afternoon, everybody.

Good morning, Professor Holbrook.

Today I’m honored to introduce Professor Andrew Holbrook.

So professor Holbrook earned his bachelor’s from UC Berkeley and a statistics masters and PhD from UC Irvine.

His research touches a number of areas of biomedical interests, including Alzheimer’s and epidemiology.

He’s currently an assistant professor of biostatistics at UCLA, where he teaches their advanced basic computer course.

And he’s the author of several pieces of scientific software.

All of it, I think, is he’s very fond of parallelization, and he also has a package including one on studying Hawkes processes, which he’s going to tell us...

Well, he’s gonna tell us about the biological phenomenon and what’s going on today.

So Professor Holbrook, thank you so much.

Okay, great.

Thank you so much for the kind invitation,

and thanks for having me this morning slash afternoon.

So today I’m actually gonna be kind of trying to present
more of a high level talk that’s gonna just focus on
a couple of different problems that have come up when modeling Hawkes processes
for public health data, and in particular for large scale public health data.
So, today I’m interested in spatiotemporal data in public health, and this can take a number of different forms.
A great example of this is in Washington D.C.
Here, I’ve got about 4,000 gunshots. You’ll see this figure again, and I’ll explain the colors to you and everything like that.
In the year 2018 alone, there were 4,000 gunshots recorded in Washington DC.
And this is just one example of really a gun violence problem in the U S of epidemic proportions.
But spatiotemporal public health data can take on many forms.
So here, for example, I have almost almost 3000 wildfires
in Alaska between the years, 2015 and 2019.
And this is actually just one piece of a larger trend that’s going on in the American west.
And then finally, another example spatiotemporal public health data is, and I believe that we don’t need to spend
too much time on this motivation, but it’s the global spread of viruses.

So for example, here, I’ve got 5,000 influenza cases recorded throughout, through 2000 to 2012. If I want to model this data, what I’m doing is I’m modeling event data. And one of the classic models for doing so is the Poisson process. And I hope that you’ll bear with me if we do just a little bit of review for our probability. We say that accounting process is a homogeneous Poisson process, point process with rate parameter, excuse me, parameter lambda, which is greater than zero. If this process is always equal to zero at zero, if it’s independent increments, excuse me, if it’s increment over non-overlapping intervals are independent random variables. And then finally, if it’s increments are Poisson distributed with mean given by that rate parameter lambda, and then the difference in the times. So we can make this model just a very little bit more complex. We can create an inhomogeneous Poisson point process, simply by saying that that rate parameter is no longer fixed, but itself is a function.
over the positive real line.
And here everything is the exact same,
except now we're saying that it's increments,
it's differences over two different time periods
are Poisson distributed, where now the mean is simply given
by the definite integral over that interval.
So we just integrate that rate function.
Okay.
So then how do we choose our rate function for the problems
that we're interested in?
Well, if we return to say the gun violence example,
then it is plausible that at least sometimes some gun
violence might precipitate more gun violence.
So here we would say that having observed an event,
having observed gunshots at a certain location
at a certain time, we might expect that the probability
of observing gunshots nearby and soon after is elevated,
and the same could plausibly go for wildfires as well.
It's that having observed a wildfire in a certain location,
this could directly contribute to the existence or to the observation of other wildfires.
So for example, this could happen by natural means.
So we could have embers that are blown by the wind,
or there could be a human that is in fact causing these wildfires, which is also quite common.

And then it’s not a stretch at all to believe that viral observation, so a child sick with influenza could precipitate another child that becomes sick with influenza in the same classroom and perhaps on the next day.

So then, the solution to building this kind of dynamic into an inhomogeneous Poisson process is simply to craft the rate function in a way that is asymmetric in time.

And what we do is we divide this rate function, $\lambda(T)$, into a background portion which is here. I denote $\nu$, and this $\nu$ can be a function itself. And typically $G$ is our triggering function.
G is non increasing. And then the only other thing that we ask is that the different events contribute in an additive manner to the rate. So here, we’ve got the background rate in this picture, We have observation T1. The rate increases. It slowly decreases. We have another observation, the rate increases. And what you see is actually that after T1, we have a nice little bit of self excitation as it’s termed, where we observe more observations. This model itself can be made just a little bit more complex if we add a spatial component. So here now, is the spatiotemporal Hawkes process where I’m simply showing you the background process, which now I’m allowing to be described by a rate function over space. And then, we also have the self excitatory component, which again, although it also involves a spatial component in it, it still has this asymmetry in time. So in this picture, we have these, what are often called immigrant events or parent events in black. And then we have the child events,
the offspring from these events described in blue. So this appears to a pretty good stochastic process model, which is not overly complex, but is simply complex enough to capture contagion dynamics. So for this talk, I’m gonna be talking about some major challenges that are confronting the really data analysis using the Hawkes process. So very applied in nature, and these challenges persist despite the use of a very simple model. So basically, all the models that I’m showing you today are variations on this extremely simple model, as far as the Hawkes process literature goes. So we assume an exponential decay triggering function. So here in this self excitatory component, what this looks like is the triggering function is simply the exponentiation of negative omega, where one over omega is some sort of length scale. And then we’ve got T minus tn. Again, that difference between a T and preceding event times. And then we’re also assuming Gaussian kernel spatial smoothers, very simple. And then finally, another simplifying assumption.
that we’re making is separability. So, in these individual components of the rate function,
we always have separation between the temporal component.
So here on the left, and then the spatial component
and this is a simplifying assumption.
So what are the challenges that I’m gonna present today?
The first challenge is big data because when we are modeling
many events, what we see is the computational complexity
of actually carrying out inference,
whether using maximum likelihood or using say,
Markov chain Monte Carlo,
well, that’s actually gonna explode quickly,
the computational complexity.
Something else is the spatial data precision.
And this is actually related to big data.
As we accrue more data,
it’s harder to guarantee data quality,
but then also the tools that I’m gonna offer up to actually
deal with poor spatial data precision are actually
gonna also suffer under a big data setting.
And then finally, big models.
So, you know, when we’re trying to draw very specific
scientific conclusions from our model, then what happens?
And all these data, excuse me, all these challenges are intertwined, and I’ll try to express that.

Finally today, I am interested in scientifically interpretable inference. So, I’m not gonna talk about prediction, but if you have questions about prediction, then we can talk about that afterward.

Okay. So I’ve shown you this figure before, and it’s not the last time that you’ll see it. But again, this is 4,000 gunshots in 2018. This is part of a larger dataset that’s made available by the Washington DC Police Department. In fact, from 2006 to 2018, we have over 85,000 potential gunshots recorded.

How are they recorded? They’re recorded using the help of an acoustic gunshot locator system that uses the actual acoustics to triangulate the time and the location of the individual gunshots. So in a 2018 paper, Charles Loeffler and Seth Flaxman, they used a subset of this data in a paper entitled "Is Gun Violence Contagious?" And they in fact apply to Hawkes process model to try to determine their question,
But in order to do, though, they had to significantly subset. They took roughly 10% of the data. So the question is whether their conclusions, which in fact work yes to the affirmative, they were able to detect this kind of contagion dynamics. But the question is, do their results hold when we analyze the complete data set? For likelihood based inference, which we're going to need to use in order to learn, in order to apply the Hawkes process to real-world data, for the first thing to see is that the likelihood takes on the form of an integral term on the left. And then we have a simple product of the rate function evaluated at our individual events, observed events. And when we consider the log likelihood, then it in fact will involve this term that I'm showing you on the bottom line, where it's the sum of the log of the, again, the rate function evaluated at the individual events. (background ringing) I'm sorry. You might be hearing a little bit of the sounds of Los Angeles in the background, and there's very little that I can do about Los Angeles.
So moving on.

So this summation in the log likelihood occurs. It actually involves a double summation.

So it is the sum over all of our observations, of the log of the rate function.

And then, again, the rate function because of the very

specific form taken by the self excitatory component

is also gonna involve this summation.

So the upshot is that we actually need to evaluate.

Every time we evaluate the log likelihood, we’re going to need to evaluate N choose two, where N is the number of data points.

N choose two terms, in this summation right here,

and then we’re gonna need to sum them together.

And then the gradient also features this, quadratic computational complexity.

So the solution, the first solution that I’m gonna offer up

is not a statistical solution.

It’s a parallel computing solution.

And the basic idea is, well, all of these terms that we need

to sum over, evaluate and sum over, let’s do it all at once

and thereby speed up our inference.

I do so, using multiple computational tools.

So the first one is I use CP, they’re just multicore CPUs.
These can have anywhere from two to 100 cores. And then I combine this with something called SIMD, single instruction multiple data, which is vectorization. So the idea, the basic idea is that I can apply a function, the same function, the same instruction set to an extended register or vector of input data, and thereby speed up my computing by a factor that is proportional to the size of the vector that I'm evaluating my function over. And then, I actually can do something better than this. I can use a graphics processing unit, which instead of hundreds cores, has thousands of cores. And instead of SIMD, or it can be interpreted as SIMD, but Nvidia likes to call it a single instruction multiple threads or SIMT. And here, what the major difference is the scale at which it’s occurring. And then, the other difference is that actually individual threads or small working groups of threads can work together. So actually the tools that I have available are very complex and a lot of need for care. There’s a lot of need to carefully code this up.
The solution is not statistical, but it’s very much an engineering solution. But the results are really, really impressive from my standpoint, because if I compare. So on the left, I’m comparing relative speed ups against a very fast single core SIMD implementation on the left.

So my baseline right here is the bottom of this blue curve. So I just create a top-line that’s flat. This is the GPU results. If I don’t use SIMD, if I use non vectorized single core computing, of course, this is still pre-compiled C++ implementation. So it’s fast or at least faster than R, and I’ll show you that on the next slide. If I do that, then AVX is twice as fast. As I increased the number of cores, my relative speed up increases, but I also suffer diminishing returns. And then that is actually all these simulations that’s 75,000 randomly generated data points at each iteration of my simulation.
But I can also just look at the seconds per evaluation.

So that's my Y axis on the right-hand side.

So ideally I want this to be as low as possible.

And then I'm increasing the number of data points

on the Y axis, on the X axis, excuse me.

And then as the number of threads that I use,

as I increased the number of threads,

then my implementation is much faster.

But again, you're seeing this quadratic computational complexity at play, right.

All of these lines are looking rather parabolic.

Finally, I go down all the way to the bottom,

where I've got my GPU curve,

again, suffering, computational complexity,

which the quadratic computational complexity,

which we can't get past, but doing a much better job

than the CPU computing.

Now you might ask, well, you might say,

well 100 fold speed up is not that great.

a 100 fold speed up is not that great.

So I'd put this in perspective and say, well,

what does this mean for R, which I use every day?

Well, what it amounts to,

and here, I'll just focus on the relative speed up

over our implementation on the right.

The GPU is reliably over 1000 times faster.
So the way that Charles Loeffler and Seth Flaxman obtained a subset of their data was actually by thinning the data. They needed to do so because of the sheer computational complexity of using the Hawkes model. So, I’m not criticizing this in any way, but I’m simply pointing out why our results using the full data set, differ. So on the left, on the top left, we have the posterior density for the spatial length scale of the self excitatory component. And when we use the full data set, then we believe that we’re operating more at around 70 meters instead of the 126 inferred in the original paper. So one thing that you might notice is our posterior densities are much more concentrated than in blue, than the original analysis in Salmon. And this of course makes sense. We’re using 10 times the amount of the data. Our temporal length scale is also meant, is also, we believe, much smaller, in fact. So now it’s down to one minute instead of 10 minutes. Again, this could be interpreted as the simple result of thinning. And then finally, I just want to focus on this on
the green posterior density. This is the proportion of events that we’re interpreting that arise from self excitation or contagion dynamics. Experts believe that anywhere between 10 and 18% of gun violence events are retaliatory in nature. So actually our inference is kind of agreeing with, it safely within the band suggested by the experts. Actually, another thing that we can do, and that also requires a pretty computationally. So this is also quadratic computational complexity. Again, is post-processing. So if, for example, for individual events, we want to know the probability that the event arose from retaliatory gun violence, then we could look at the self excitatory component of the rate function divided by the total rate function. And then we can just look at the posterior distribution of this statistic. And this will give us our posterior probability that the event arose from contagion dynamics at least. And you can see that we can actually observe a very wide variety of values. So the issue of big data is actually not gonna go away,
as we move on to discussing spatial data precision.

Now, I'll tell you a little bit more about this data.

All the data that we access is freely accessible online, is rounded to the nearest 100 meters by the DC Police Department. And the reason that they do this is for reasons of privacy.

So one immediate question that we can ask is, well,

how does this rounding actually affect our inference?

Now we actually observed wildfires of wildly different sizes. And the question is, well, how does...

If we want to model the spread of wildfires, then it would be useful to know where the actual ignition site, the site of ignition was.

Where did the fire occur originally?

And many of these fires are actually discovered out in the wild, far away from humans.

And there’s a lot of uncertainty.

There’s actually a large swaths of land that are involved.

Finally, this, this global influenza data is very nice for certain reasons.

For example, it features all of the observations, actually provide a viral genome data.

So we can perform other more complex analyses on the data.
And in fact, I'll do that in the third section for related data. But the actual spatial precision for this data is very poor. So, for some of these viral cases, we know the city in which it occurred. For some of them, we know the region or the state in which it occurred. And for some of them, we know the country in which it occurred. So I'm gonna start with the easy problem, which is analyzing the DC gun violence, the DC gunshot data. We take that at face value and we simply use, place a uniform prior over the 10,000 meters square that is centered at each one of our observations. So here I'm denoting our actual data, our observed data with this kind of Gothic X, and then I'm placing a prior over the location at which the gunshot actually occurred. And this is a uniform prior over a box centered at my data. And using this prior actually has another interpretation similar to some other concepts from the missing data literature.
And use of this prior actually corresponds to using something called the group data likelihood. And it’s akin to the expected, complete data likelihood if you’re familiar with the missing data literature. So what we do, and I’m not gonna get too much into the inference at this point, but we actually use MCMC to simultaneously infer the locations, and the Hawkes model parameters, the rate function parameters at the same time. So here, I’m just showing you a couple of examples of what this looks like. For each one of our observations colored yellow, we then have 100 posterior samples. So these dynamics can take on different forms and they take on different forms in very complex ways, simply because what we’re essentially doing when we’re... I’m going to loosely use the word impute. When we’re imputing this data, when we’re actually inferring these locations, we’re basically simulating from a very complex n-body problem. So on the left, how can we interpret this? Well, we’ve got these four points and the model believes that actually they are farther away
from each other than observed.
Why is that?
Well, right in the middle here, we have a shopping center,
where there’s actually many less gunshots.
And then we’ve got residential areas where there are many more gunshots on the outside.
And the bottom right, we actually have all of these,
we believe that the actual locations of these gunshots collect closer together, kind of toward a very high
intensity region in Washington, DC.
And then we can just think about the general posterior displacement.
So the mean posterior displacement.
So in general, are there certain points that, where the model believes that the gunshots occurred
further away from the observed events?
And in general, there’s not really.
It’s hard to come up with any steadfast rules.
For example, in the bottom, right, we have some shots,
some gunshots that show a very large posterior displacement,
and they’re in a very high density region.
Whereas on the top, we also get large displacement
and we’re not surrounded by very many gunshots at all.
So it is a very complex n-body problem
that we’re solving. And the good news is, for this problem, it doesn’t matter much anyway. The results that we get are pretty much the same.

I mean, so from the standpoint of statistical significance, we do get some statistically significant results. So in this figure, on the top, I’m showing you 95% credible intervals, and this is the self excitatory spatial length scale. We believe that it’s smaller, but from a practical standpoint, it’s not much smaller.

It’s a difference between 60 meters and maybe it’s at 73 meters, 72 meters. But we shouldn’t take too much comfort because actually as we increase the spatial precision, we find that the model that does not take account of the rounding, performs much worse.

So for example, if you look in the table, then we have the fixed locations model, where I’m not actually inferring the locations. And I just want to see, what’s the empirical coverage of the 95% credible intervals? And let’s just focus on the 95% credible intervals, specifically, simply because actually the other intervals,
the 50% credible interval, the 80% credible interval, they showed the similar dynamic, which is that as we, so if we start on the right-hand side, we have precision down to down to 0.1. This is a unit list example. So we have higher precision, actually. Then we see that we have very good coverage, even if we don’t take this locational coarsening into account. But as we increase the size of our error box, then we actually lose coverage, and we deviate from that 95% coverage. And then finally, if we increase too much, then we’re never actually going to be capturing the true spatial length scale, whereas if we actually do sample the locations, we perform surprisingly well, even when we have a very high amount of spatial coarsening. Well, how else can we break the model? Another way that we can break this model, and by break the model, I mean, my naive model where I’m not inferring the locations. Another way that we can break this model is simply by considering data. where we have variable spatial coarsening. That is where different data points are coarsened different amounts, so we have a variable precision.
So considering the wildfire data, we actually see something with the naive approach where we’re not inferring the locations. We actually see something that is actually recorded elsewhere in the Hawkes process literature. And that is that when we try to use a flexible background function, as we are trying to do, then we get this multimodal posterior distribution. And that’s fine. We can also talk about it in a frequentist, from the frequency standpoint, because it’s observed there as well in the maximum likelihood context, which is, we still see this multimodality. What specific form does this multimodality take? So what we see is that we get modes around the places where the background rate parameters, the background length scale parameters are equal to the temporal, excuse me, the self excitatory length scale parameters. So for the naive model, it’s mode A, it believes that the spatial length scale is about 24 kilometers, and that the spatial length scale of the self excitatory dynamics are also roughly 24 kilometers. And then for the other mode,
we get equal temporal length scales. So here, it believes 10 days, and 10 days for the self excitatory in the background component. And this can be very bad indeed. So for example, for mode A, it completely, the Hawkes model completely fails to capture seasonal dynamics, which is the first thing that you would want it to understand. The first thing that you would want it to understand is that wildfires... Okay, I need to be careful here because I’m not an expert on wildfires. I’ll go out on a limb and say, wildfires don’t happen in Alaska during the winter. On the other hand, when we use the full model and we’re actually simultaneously inferring the locations, then we get this kind of Goldilocks effect, where here, the spatial length scale is somewhere around 35 kilometers, which is between the 23 kilometers and 63 kilometers for mode modes A and B, and we see that reliably. I can stop for some questions because I’m making good time. Does anybody have any questions, if you want to ask? What’s the interpretation?
of the spatial length scale and the temporal length scale?

What do those numbers actually mean?

Yeah, thank you.

So, the interpretation of the...

I think that the most useful interpretation,

So just to give you an idea of how they can be interpreted.

So for example, for the self excitatory component, right,

that's describing the contagion dynamics.

What this is saying is that if we see a wildfire,

then we expect to observe another wildfire

with mean distribution of one day.

So the temporal length scale is in units days.

So in the full model, after observing the wildfire,

we expect to see another wildfire with mean, you know,

on average, the next day.

And this of course, you know, we have this model

that's taking space and time into account.

So the idea though, is that because of the separability

in our model, we're basically simply expecting to see it somewhere.

Thank you.

Any other questions?

Hi, can I have one question?

Go head.
Student: Okay.
I'm curious.

What is a main difference between the naive model A and the naive model B?

Okay.

So, sorry.

This is... I think I could have presented this aspect better within the table itself.

So this is the same exact model.

But all that I'm doing is I'm applying the model multiple times.

So in this case, I'm using Markov chain Monte Carlo.

So one question that you might ask is, well, what happens when I run MCMC multiple times?

Sometimes I get trapped in one mode.

Sometimes I get trapped in another mode.

You can just for, you know, a mental cartoon, we can think of like a (indistinct) mixture of Gaussian distribution, right.

Sometimes I can get trapped in this Gaussian component.

Sometimes I could get trapped in this Gaussian component.

So there's nothing intrinsically wrong with multimodality.

We prefer to avoid it as best we can simply because it makes interpretation much more difficult.

In this case, if I only perform inference
and only see mode A, then I’m never actually gonna be picking up on seasonal dynamics. Does that (indistinct)?

Yeah, it’s clear.

Okay.

Okay, and I also (indistinct).

So for the full model, you can capture the spatial dynamic property.

So how do you estimate a baseline part?

Oh, okay, great.

In the exact same way.

Okay, I see.

So I’m jointly, simultaneously performing inference.

over all of the model parameters.

And I can go all the way back.

Right.

’Cause it’s actually a very similar model.

Yes.

So this is my baseline.

And so, for example, when we’re talking about that temporal

smooth that you saw on that last figure, where I’m supposed to be capturing seasonal dynamics.

Well, if tau T, which I’m just calling my temporal length scale, if that is too large,
then I'm never going to be capturing those seasonal dynamics, which I would be hoping to capture precisely using this background smoother. So it looks like they assume the formula for the baseline, and then you estimate some parameters in these formulas.

In my understanding, in the current Hawkes literature, somebody uses (indistinct) function to approximate baseline also. This is also interesting. Thank you.

As just a quick follow up on when you were showing the naive model, this maybe a naive question on my part. Did you choose naive model A to be the one that does the type seasonality or is that approach just not (indistinct) seasonality?
So I think that the point is that sometimes based on, you know, I'm doing MCMC. It's random in nature, right. So just sometimes when I do that, I get trapped in that mode A, and sometimes I get trapped in that mode B. The label that I apply to it is just arbitrary, but maybe I'm not getting your question. No, I think you did. So, it's possible that we detect it. It's possible that we don't. Exactly. And that's, you know, That's what it is. That actually inferring the locations can somehow, at least in this case, right, this is kind of nice though, that this can actually give you, that can help resolve that multimodality. So this is a case study, really, that this can help resolve that multimodality. So back to the comparison between CPU and GPU. Let's say, if we increase the thread of CPU, say like to infinity, will it be possible that the speed of CPU match the speed up of GPU? So.
You’re saying if we increase.
So, can I ask you one more time?
Can I just ask for clarification?
You’re saying if we increase what to infinity?
'I think in the graph you’re increasing the threads'
of CPU from like one to 80.
And the speed up increase as the number
of threats increasing.
So just say like, let’s say the threads of CPU
increase to infinity, will the speed up match,
because GPU with like (indistinct).
Very high, right. Yeah, yeah.
Let me show you another figure,
and then we can return to that.
I think it’s a good segue into the next section.
So, let me answer that in a couple slides.
Okay, sounds good.
Okay.
So, questions about.
I’ve gotten some good questions about how
do we interpret
the length scales and then this makes me think about,
well, if all that we’re doing is interpreting
the length scales, how much is that telling us about?
the phenomenon that we’re interested in?
And can we actually craft more complex hierarchical models
so that we can actually learn something perhaps
even biologically interpretable?

So here, I’m looking at 2014, 2016 Ebola virus outbreak data. This is over almost 22,000 cases.

And of these cases, we have about 1600 that are providing us genome data.

And then of those 1600, we have a smaller subset that provide us genome data, as well as spatiotemporal data.

So often people use genome data, say RNA sequences in order to try to infer the way that different viral cases are related to each other.

And the question is, can we pull together sequenced and unsequenced data at the same time?

So what I’m doing here is, again, I’m not inventing this.

This is something that already exists.

So all that I’m doing is modifying my triggering function G,

and giving it this little N, this little subscript right there,

which is denoting the fact that I’m allowing different viral observations to contribute to the rate function in different manners.

And the exact form that that’s gonna take on for my specific simple model that I’m using, is I’m going to give this data N.

And I’m gonna include this data N parameter
in my self excitatory component.

And this data N is restricted to be greater than zero.

So if it is greater than one,

I’m gonna assume that actually, this self excite,

excuse me, that this particular observation,

little N is somehow more contagious.

And if data is less than one,

then I’m going to assume that it’s less contagious.

And this is an entirely unsatisfactory part of my talk,

where I’m gonna gloss over a massive part of my model.

And all that I’m gonna say is that

this Phylogenetic Hawkes process, which I’m gonna be telling

you about in the context of big modeling,

and that challenge is that we start

with the phylogenetic tree, which is simply the family tree

that is uniting my 1600 sequenced cases.

And then based on that, actually conditioned on that tree,

we’re gonna allow that tree to inform the larger

covariants of my model parameters, which are then going to

contribute to the overall Hawkes rate function

in a differential manner, although it’s still additive.

Now, let’s see.

Do I get to go till 10 or 9:50?
So you can go till 10.

Okay, great.

So then, I'll quickly say that if I'm inferring all of these rates, then I'm inferring over 1300 rates.

So that is actually the dimensionality of my posterior distribution.

So a tool that I can use, a classic tool over 50 years old at this point, is I can use the random walk metropolis algorithm, which is actually going to sample from the posterior distribution of these rates. And it's gonna do so in a manner that is effective in low dimensions, but not effective in high dimensions.

And the way that it works is say, we start at negative three, negative three. What we want to do is we want to explore this high density region of this bi-variate Gaussian, and we slowly amble forward, and eventually we get there.

But this algorithm breaks down in moderate dimensions.

An algorithm that I think many of us are aware of at this point, that is kind of a workhorse in high dimensional Bayesian inference is Hamiltonian Monte Carlo.
our log posterior in order to intelligently guide
the MCMC proposals that we’re making. So, again, let’s just pretend that we start
at negative three, negative three, but within a small number of steps,
we’re actually effectively exploring
that high density region, and we’re doing so because we’re using that gradient information
of the log posterior.
I’m not going to go too deep right now into
the formulation of Hamiltonian Monte Carlo, for the sake of
time.
But what I would like to point out,
is that after constructing this kind of physical system
that is based on our target distribution
on the posterior distribution, in some manner,
we actually obtain our proposals within the
MCMC.
We obtain the proposals by simulating, by
forward simulating
the physical system, according to Hamilton’s equations.
Now, this simulation involves is a massive number
of repeated gradient evaluations.
Moreover, if the posterior distribution is an ugly one,
that is if it is still conditioned, which we interpret as,
the log posterior Hessian has eigenvalues
that are all over the place.
Then we can also use a mass matrix, $M$, which is gonna allow us to condition our dynamics, and make sure that we are exploring all the dimensions of our model in an even manner.

So the benefit of Hamiltonian Monte-Carlo is that it scales to tens of thousands of parameters. But the challenge is that that HMC necessitates repeated computation at the log likelihood, it’s gradient and then preconditioning.

And the best way that I know to precondition actually involves evaluating the log likelihood Hessian as well.

And I told you that the challenges that I’m talking about today are intertwined.

Well, we’ve already managed to speed up the log likelihood computations that are quadratic in computational complexity.

Well, it turns out that the log likelihood gradient and the log likelihood Hessian are all quadratic and computational complexity.

So this means that as the size of our data set grows, we’re going to... HMC, which is good at scaling to high dimensional models.
is going to break down because it’s just gonna take too long
to evaluate the quantities that we need to evaluate.
To show you exactly how these parallel gradient calculations can work.
So, what am I gonna do?
I’m gonna parallelize again on a GPU or a multi-core CPU implementation,
and I’m interested in evaluating or obtaining the quantities in the red box.
These are simply the gradient of the log likelihood with respect to the individual rate parameters.
And because of the summation that it involves,
we actually obtain in the left, top left,
we have the contribution of the first observation.
we have the contribution of the second observation.
all the way up to the big int observation,
that contribution to the gradient term.
And these all need to be evaluated and summed over.
So what do we do?
We just do a running total, very simple.
We start by getting the first contribution.
We keep that stored in place.
We evaluate the second contribution,
all at the same time in parallel,
and we simply increment our total observat-
903 00:54:01.360 --> 00:54:04.820 excuse me, our total gradient by that value.
904 00:54:04.820 --> 00:54:05.810 Very simple.
905 00:54:05.810 --> 00:54:07.373 We do this again and again.
906 00:54:08.340 --> 00:54:10.810 Kind of complicated to program, to be honest.
907 00:54:10.810 --> 00:54:11.763 But it’s simple.
908 00:54:15.812 --> 00:54:16.645 It’s simple when you think about it from the high level.
909 00:54:19.210 --> 00:54:21.370 So I showed you this figure before.
910 00:54:21.370 --> 00:54:24.060 And well, a similar figure before,
911 00:54:24.060 --> 00:54:25.630 and the interpretations are the same,
912 00:54:25.630 --> 00:54:29.910 but here I’ll just focus on the question that I received.
913 00:54:29.910 --> 00:54:32.060 In the top left, we have the gradient.
914 00:54:32.060 --> 00:54:33.870 In the bottom left, excuse me,
915 00:54:33.870 --> 00:54:35.160 top row, we have the gradient.
916 00:54:35.160 --> 00:54:36.810 Bottom row, we have the Hessian,
917 00:54:36.810 --> 00:54:41.810 and here I’m increasing to 104 cores.
918 00:54:41.810 --> 00:54:45.970 So this is not infinite cores, right.
919 00:54:45.970 --> 00:54:47.320 It’s 104.
920 00:54:47.320 --> 00:54:50.233 But I do want you to see that there’s diminishing returns.
921 00:54:54.260 --> 00:54:57.480 And to give a little bit more technical
922 00:54:57.480 --> 00:54:59.093 response to that question,
923 00:55:01.530 --> 00:55:03.940 the thing to bear in mind is that
924 00:55:03.940 --> 00:55:07.700 it’s not just about the number of threads that we use.
925 00:55:07.700 --> 00:55:12.170 It’s having a lot of RAM very close
926 00:55:12.170 --> 00:55:15.110 to where the computing is being done.
927 00:55:15.110 --> 00:55:18.230 And that is something that GPUs,
929 00:55:25.510 --> 00:55:28.470 So why is it important to do all this parallelization?
Well, this is really, I want to kind of communicate this fact because it is so important.

This slide underlines almost the entire challenge of big modeling using the spatiotemporal Hawkes process.

The computing to apply this model to the 20,000 plus data points took about a month using a very large Nvidia GV100 GPU.

Why? Because we had to generate 100 million Markov chain states at a rate of roughly three and a half million each day.

After 100 million Markov chain states, this is the empirical distribution on the left of the effective sample sizes across all of the individual rates that we’re inferring.

The minimum is 222, and that’s right above my typical threshold of 200, because in general, we want the effective sample size to be as large as possible.

Well, why was it so difficult? Well, a lot of the posterior, a lot of the marginal posteriors for our different parameters were very complex.
So for example, here, I just have one individual rate, and this is the posterior that we learned from it. It’s bi-modal. And not only is it bi-modal, but the modes exist on very different scales. Well, why else is it a difficult posterior to sample from? Well, because actually, as you might imagine, these rates have a very complex correlation in structure. This is kind of repeating something that I said earlier. when we were actually inferring locations, which is that what this amounts to is really simulating a very large n-body problem. But what’s the upshot? Well, we can actually capture these individual rates, which could give us hints at where to look for certain mutations that are allowing, say in this example, the Ebola virus to spread more effectively. And here, red is generally the highest, whereas blue is the lowest. We can get credible intervals, which can give us another way of thinking about, you know, where should I be looking in this collection of viral samples, for the next big one?
And then I can also ask, well, how do these rates actually distribute along the phylogenetic tree? I can look for clades or groups of branches that are in general, more red in this case than others.

So, something that I... Okay, so it’s 10 o'clock, and I will finish in one slide.

The challenges that I’m talking about today, they’re complex and they’re intertwined, but they’re not the only challenges. There are many challenges in the application of spatiotemporal Hawkes models, and there’s actually a very large literature.

So some other challenges that we might consider, and that will also be extremely challenging to overcome in a big data setting.

So, kind of the first challenge is flexible modeling. So here, we want to use as flexible a Hawkes model as possible.

And this challenge kind of encapsulates one of the great ironies of model-based nonparametrics, which is that, the precise time that we actually want to use a flexible model, is the big data setting.

I mean, I don’t know if you recall my earlier slide where I was showing the posterior distribution of some of the length scales associated with
the Washington DC data, and they’re extremely tight.

But this is actually exactly where we’d want to be able to use a flexible model, because no matter what, if I apply my model to 85,000 data points, I’m going to be very certain in my conclusion, conditioned on the specific model that I’m using.

There’s also boundary issues, right. This is a huge, a huge thing. So for those of you that are aware of the survival literature, which I’m sure many of you are, you know, they’re censoring.

So what about gunshots that occurred right outside the border of Washington DC, and it occurred as a result of gunshots that occurred within the border? And then we can flip that on its head. What about parent events outside of Washington DC that precipitated gun violence within Washington DC.

And then finally, sticking with the same example, differential sampling.

You can be certain that those acoustic gunshot locators, location system sensors are not planted all over Washington DC. And how does their distribution affect things?
Okay.

This is joint work with Mark Suchard at UCLA, also at UCLA.

And then my very good friend, Xiang Ji at Tulane.

It’s funded by the K-Award Big Data Predictive Phylogenetics with Bayesian learning, funded by the NIH.

And that’s it.

Thank you.

All right.

Thank you so much, Professor Holbrook.

Does anybody have any other questions?

Yeah.

Any other questions from the room here, or from Zoom?

Yeah.