Talk JSM

S1

Hello everyone, I'm very happy to present at the 2020 Joint Statistical Meeting today the methodological work we did so far with the introduction of the lights model.

The technology is developed in the Machine Learning lab of the start-up Califrais I co-founded, and is part of a research project I'm conducting at the French Institute of Health and Medical Research.

It's a joint work with ….

The idea of this talk is to briefly give the main motivations and technical fondations of the model, with all the intuitions behind the formulas and the explanations of the choices to enable successful learning.

S2

The model deals with..

One of the special feature of the model is its flexibility …

S3

Concerning the use cases, you wanna try this model as soon as you wanna predict the risk…

And what is interesting is that the model provides powerfull interpretations through feature selection in a very flexible way.

To be a little bit more specific, it could be used as a real-time decision support in a medical…

but it could also be used for instance in a customer's…
the time when a client stops using a company's product or service in a churn prediction setting ; in this context the longitudinal data could be the customer's activity recorded during the client lifetime in the company.

S4

Ok so let’s start with the framework introduction.

We are in a survival analysis context, so we denote T the censored time, which is the minimum between t^star the true event time and C the censorring random variable ; and we denote delta the censoring indicator.

X denotes the time-indep features in high-dimension

and we denote Y the multivariate longitudinal process also in high-dimension

Then, without loss of generality, we suppose that the heterogeneity of the population can be represented through latent subgroups G, with K different subgroups.

We denote pi_xi_k(x) the latent class membership probability given x and we consider a softmax link function.

Let’s move forward with the model presentation now

S5

We propose a group-specific generalized linear mixed model for the marker trajectories, with m_kl the linear predictor and the beta’s being… and b^l the random effect

a first interesting point to notice here is that we take the design features u(t) to be a polynomial time vector with a large order, so that we dont have to choose in advance, how flexible the trajectories need to be with time : indeed with an appropriate penalty on beta that we’ll present later, the model will be able to select automatically the proper flexibility.

Then we use the notation capital D for the covariance matrix to account for dependence between the different longitudinal outcomes

Now for the survival submodel, we propose a group-specific Cox risk, where we incorporate all the time-independant features; and then we add a bunch of A known functionals phi_a for each trajectory defining what is called shared associations with this gamma here being the corresponding joint association parameters to infer.

Concerning the choice of the association structure, what's interesting is that one can consider at the same time different parameterizations with no a priori and let the model decide during the learning phase which one is relevant, thanks again to a penalty applied on gamma this time. We consider for instance …

S6

Let us precise a few notations. So we start with a dataset with n examples where every single trajectory can have its own number of measurments.

y_i is the concatenation of all trajectories for subject i

b_i is the concatenation of all the random effects for subject i

we denote the design features matrix U_i and V_i like this

Then beta_k and finally M_ik this vector in R_ni which basically represents linear predictors for all measurements of subject i given that he belongs to subgroup k

S7

We denote theta the collection of all unknown parameters to estimate

Each trajectory is then assumed to be from a one-parameter exponential family.

The density for the survival part writes naturally like this…
And with some mild assumptions, one finally obtains the likelihood of the lights model !

Let's dive now to the inference part to describe the procedure for estimating the parameters of the model

S8

We propose to minimize this penalized objective, which is minus the likelihood + a penalization term composed for each subgroup of an elacticnet penalty for the vector xi_k and then we add sparse group lasso penalties for gamma_k and beta_k to perform feature selection at the trajectory level, which is very important.

and all the zetas are hyper-parameters that need to be carefully tuned through a cross-validation procedure to choose the appropriate penalty strength.

In the end the resulting optimization problem we need to solve is the following.

S9
In order to derive an algorithm for this objective, we introduce a so-called QNMCEM algorithm, being a combination between an EM algorithm with Monte Carlo approximations, and multiple L-BFGS-B algorithms. Let's briefly describe the algorithm.

We first need to compute this joint distribution. We can easily write it but it's a bit of a long formula, so let's skip the details.

For the E-step of the algorithm, and suppose that we are at step w + 1 of the algorithm, we need to compute this quantity which is the expected negative log-likelihood of the complete data conditional on the observed data and the current estimate of the parameters.

The thing is that this expression requires to compute expectations of this form for different functions g, where we denote like this the posterior probability of the latent class membership using current estimate of the parameters.

Again let's skip the technical details here, but we use Monte Carlo approximations to compute those quantities.

S10
For the M-step now, we need to solve this optimization problem to update the parameters.

Of course there are some technicallities that we don't have time to present today. Some parameter updates are given in closed-form, which is the case for D for instance.

Some others require a little more work, for instance for xi_k. Denoting P based on all the quantities involved in Qn that depend on xi_k, then the update for xi_k requires to solve the following convex minimization problem, the point is that it's not differentiable because of the penalty.

We then rewrite the minimization problem as the following differentiable problem with box constraints (using the fact the absolute value of a number is the sum of its positive and negative part, so that l1 norm rewrites like this), and so that one can use the L-BFGS-B routine for the update.
This solver requires the exact value of the gradient, and we can express it with no problem.

We actually use similar tricks for the beta's and gamma's updates.

One last point which is very important in our method concerns the evaluation strategy we propose, which we call the real-time prediction paradigm, also used for the CV. In a few words, the idea is to use this predictive marker rule using the final parameter estimates, with the concordance index metric where ti_max is the time for subject i when one wants to perform the risk prediction – so in practice, the time up to which one has data measurements for Y_i
and this marker turns out to be an estimate of the following posterior probability.

S11
>To conclude, we introduced a prognostic method
>where we consider an appropriate penalized objective in order to perform feature selection and to prevent overfitting.
>We derived an estimation algorithm to perform inference, that, among other things,
>automatically determines significant prognostic longitudinal features
>The python 3 implementation of the model is available at this URL.
>All the details of the maths, as well as the results of the simulation study and the application on real datasets will be available soon on an open-access arxiv paper.

If you are interested in this model, I will be happy to answer your questions. Thank you very much !