DeepPumas: Combining Machine Learning and Traditional Statistical Modelling

Mohamed Tarek, Niklas Korsbo {mohamed,niklas}@pumas.ai PumasAI

Understanding NLME

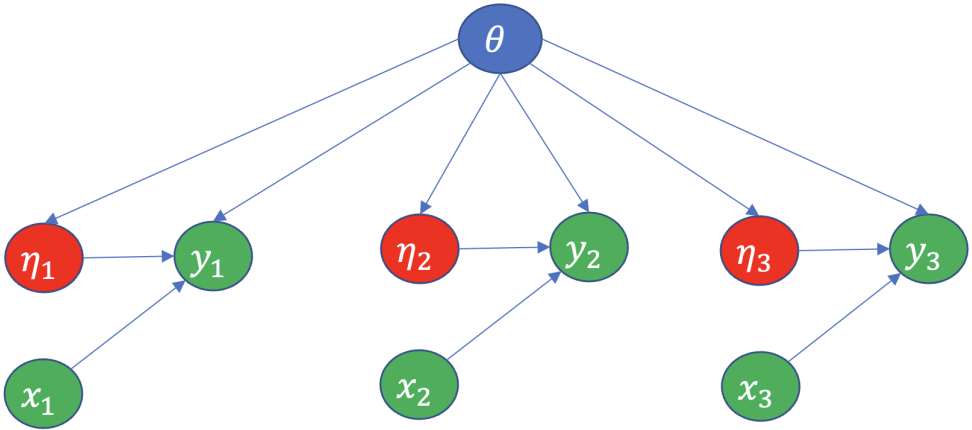
Definition

Nonlinear mixed effects (NLME) models are popular in pharmacometrics. Parameters of NLME models are either:

- Fixed effects θ: model parameters modelled as deterministic quantities, or
- Random effects η: model parameters modelled as random variables

In pharmacometrics, NLME models also tend to be **hierarchical** in nature where fixed effects are **population-level parameters** and random effects are **subject-specific parameters**.

Definition



Definition

θ

$\eta_i \mid \theta$

x_i

$y_i \mid \theta, \eta_i, x_i$

```
@model begin
  @param begin
    θ ∈ VectorDomain(4, lower = zeros(4))
    Ω ∈ PSDDomain(2)
    Σ ∈ RealDomain(lower = 0.0)
    a ∈ RealDomain(lower = 0.0, upper = 1.0)
  end
  @random begin
    η ~ MvNormal(Ω)
  end
  @covariates sex wt etn
  @pre begin
    θ1 := θ[1]
    Ka = θ1
    CL = θ[2] * ((wt / 70)^0.75) * (θ[4]^sex) *
      exp(η[1])
    Vc = θ[3] * exp(η[2])
  end
  @dynamics begin
    Depot' = -Ka * Depot
    Central' = Ka * Depot - (CL / Vc) * Central
    Res' = Depot - Central
  end
  @derived begin
    conc = @. Central / Vc
    dv ~ @. Normal(conc, conc * Σ)
    T_max = maximum(t)
  end
  @observed begin
    obs_cmax = maximum(dv)
  end
end
```

Random Effects in Other Fields

Random effects are not unique to pharmacometrics or to hierarchical models. In other fields, random effects are often called:

- Latent variables, or
- Hidden variables

These are generally just **unobserved quantities** that we are **uncertain** about their value and we model them as **random variables**.

Understanding Random Effects

All variables in your model can be grouped into 2 categories:

- Observable variables, e.g. the drug concentrations
- Unobservable variables, e.g. the typical value of clearance or a subject-specific deviation from a typical value

Some unobservable parameters are modelled as **deterministic quantities**, aka **fixed effects**. Others are given a **prior distribution** for what we believe their values might be prior to observing any of the observable variables. These are known as the **random effects**.

Understanding Random Effects

- Inference is the task of learning about the values of unobserved quantities in your model given the observed ones.
- In this world view, **missing data** is just observable variables that were never observed, and they can be modelled as any other **unobserved variable**.
- We generally call observable quantities in a study **data** and unobservable quantites **parameters**.

Understanding Random Effects

In hierarchical NLME models, random effects can be thought of as **unobserved covariates**, which may be:

- Measurable but unknown,
- Immeasurable but has a physical meaning,
- Immeasurable and abstract or has no interpretable meaning!

These are things we don't know about our subjects but that we care to quantify because they are believed to have predictive power.

Random Effects in Pscychology

- **Latent variables**: extraversion, wisdom, intelligence, personality traits, etc.

- **Observed variables**: personality test results and IQ test results

Random Effects in Economics

- **Latent variables**: quality of life, political stability, happiness
- **Observed variables**: wealth, GDP, employment rate, working hours

Random Effects in Machine Learning

All the following models use random effects or latent variables:

- Probabilistic principal component analysis
- Latent Dirichlet allocation
- Variational autoencoders (https://www.youtube.com/watch?v=Q1XuXwPVFko)
- Generative adversarial networks (https://www.youtube.com/watch?v=kSLJriaOumA)
- Many more "generative models"

Many of these models use standard Gaussian random effects $N(0, I)$ as input to a neural network.

Random Effects in Machine Learning

In machine learning models, latent variables **may have no interpretable meaning**. They are just **unobserved degrees of freedom** in a model.

Random Effects as Placeholders

In DeepPumas, we often use random effects as placeholders anywhere in the model we suspect there is an unobserved quantity that affects our observations.

Conclusions

- Random effects describe some sort of unobserved covariates
- Random effects don’t have to be directly interpretable
- Neural networks with simple random effects as inputs have a very high data fitting and data generation power

Hands-on

Backup Slides

NLME Model Specification

Let

- θ be the population-level, fixed effects
- η_i be the random effects associated with subject i
- x_i be the observed covariates of subject i
- y_i be the observed response of subject i

Conditional Likelihood

Assume there are 3 subjects. The conditional likelihood for the population is given by:

$$p(y \mid \theta, \eta_1, \eta_2, \eta_3, x) = \prod_{i=1}^{3} p(y_i \mid \theta, \eta_i, x_i)$$

This is the total probability of observing all the pairs (x_i, y_i) for all $i \in 1 \ldots 3$.

Joint Maximum Likelihood Estimation

How to fit the data to our model? Should we maximize the conditional probability wrt all of θ and the η_is simultaneously?

Joint Maximum Likelihood Estimation

How to fit the data to our model? Should we maximize the conditional probability wrt all of θ and the η_is simultaneously? **NO!** But why?

Joint Maximum Likelihood Estimation

- This maximization would be the maximum likelihood estimate (MLE) of the joint probability.

- This will ignore any connection between the various η_is. So you would be ignoring the hierarchical structure in your data.

Joint Maximum-a-Posteriori Estimation

What if we multiply the objective by the prior over the η_is as regularization to establish a connection between the η_is and constrain their values?

$$p(y \mid \theta, \eta_1, \eta_2, \eta_3, x) \cdot \prod_{i=1}^{3} p(\eta_i \mid \theta) = \prod_{i=1}^{3} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta)$$

Joint Maximum-a-Posteriori Estimation

Better, but not good enough!

- This maximization would be the maximum-a-posteriori (MAP) solution of the joint probability.
- If the covariance of the prior of η_i is a parameter in θ (a typical case), the posterior may not have a well defined mode!
- When there are a few data points per subject, a degenerate MAP solution involves weakening the prior over the η_is to the point where you would be effectively fitting a separate model for each subject.
- So it doesn't sufficiently guard against degenerate MAP solutions and over-fitting a few data points per subject.

Eight Schools Model in Pumas

- Population level fixed effects: μ, τ
- Random effect of subject i: η_i
- Covariate of subject i: σ_i
- Observation of subject i: y_i

$$
\begin{aligned}
\mu &\in \mathcal{R} \\
\tau &\in \mathcal{R}^+ \\
\eta_i &\sim N(\mu, \tau) \\
y_i &\sim N(\eta_i, \sigma_i)
\end{aligned}
$$

Marginal Likelihood

How to guard against degenerate MAP solutions and over-fitting a few data points? **Use the marginal likelihood.**

Marginal Likelihood

The marginal likelihood is:

$$
\begin{aligned}
p(y \mid \theta, x) &= \int_{\eta_3} \int_{\eta_2} \int_{\eta_1} \left(\prod_{i=1}^{3} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) \right) d\eta_1 d\eta_3 d\eta_3 \\
&= \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i \\
&= \prod_{i=1}^{3} p(y_i | \theta, x_i)
\end{aligned}
$$

Marginal Likelihood

$$
\begin{aligned}
p(y \mid \theta, x) &= \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i \\
&\approx \prod_{i=1}^{3} \frac{1}{M} \sum_{j=1}^{M} p(y_i \mid \theta, \eta_{i,j}, x_i)
\end{aligned}
$$

where $\eta_{i,j}$ is the jth sample from the prior $p(\eta_i \mid \theta)$.

Marginal Likelihood

$$
\begin{aligned}
p(y \mid \theta, x) &= \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i \\
&\approx \prod_{i=1}^{3} \frac{1}{M} \sum_{j=1}^{M} p(y_i \mid \theta, \eta_{i,j}, x_i)
\end{aligned}
$$

So we are taking the **average conditional likelihood** $p(y_i \mid \theta, \eta_i, x_i)$ for each subject i over all the possible values of η_i **sampled from the prior** $p(\eta_i \mid \theta)$. **Degenerate weak priors are heavily penalized!**

Marginal Likelihood

Maximizing the marginal probability turns out to be implicitly learning and using the conditional posteriors $\eta_i \mid \theta, x_i, y_i$ in the process. This is non-obvious!

Expectation Maximization

$$p(y \mid \theta, x) = \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i$$

One way to **maximize the marginal likelihood** above is using the stochastic approximation expectation-maximization (**SAEM**) algorithm.

Expectation Maximization

$$p(y \mid \theta, x) = \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i$$

In SAEM, maximizing the marginal likelihood is done by maximizing:

$$\text{obj} = \sum_{i=1}^{3} E_{\eta_i|\theta', x_i, y_i} \Big(\log p(y_i \mid \theta, \eta_i, x_i) + \log p(\eta_i|\theta) \Big)$$

where θ' is the value of θ from the previous M step.

Expectation Maximization

$$\text{obj} = \sum_{i=1}^{3} E_{\eta_i|\theta',x_i,y_i}\Big(\log p(y_i \mid \theta, \eta_i, x_i) + \log p(\eta_i|\theta)\Big)$$

Notice how the term $E_{\eta_i|\theta',x_i,y_i} \log p(\eta_i|\theta)$ is only a constant away from:

$$\begin{aligned}
-\text{KL}\Big(p(\eta_i \mid \theta', x_i, y_i) \,||\, p(\eta_i|\theta)\Big) &= -E_{\eta_i|\theta',x_i,y_i} \log \frac{p(\eta_i \mid \theta', x_i, y_i)}{p(\eta_i|\theta)} \\
&= E_{\eta_i|\theta',x_i,y_i} \log \frac{p(\eta_i|\theta)}{p(\eta_i \mid \theta', x_i, y_i)} \\
&= E_{\eta_i|\theta',x_i,y_i} \log p(\eta_i|\theta) - E_{\eta_i|\theta',x_i,y_i} \log p(\eta_i \mid \theta', x_i, y_i) \\
&= E_{\eta_i|\theta',x_i,y_i} \log p(\eta_i|\theta) - \text{const}
\end{aligned}$$

Expectation Maximization

- **In other words, maximizing** $E_{\eta_i|\theta',x_i,y_i} \log p(\eta_i|\theta)$ **wrt** θ **indirectly minimizes KL** $\left(p(\eta_i \mid \theta', x_i, y_i) \mid\mid p(\eta_i|\theta)\right)$ **in each SAEM step.** This is non-obvious!
- This is why shrinkage is generally not an issue when using the entire conditonal posterior $p(\eta_i \mid \theta, x_i, y_i)$ to make predictions after a fit.
- This is also why the combination of a weak prior and a strong posterior (due to over-fitting a few data points) is heavily penalized when maximizing the marginal likelihood.

Expectation Maximization

obj is typically approximated using:

$$\text{obj} \approx \frac{1}{M} \sum_{i=1}^{3} \sum_{j=1}^{M} \left(\log p(y_i \mid \theta, \eta_{i,j}, x_i) + \log p(\eta_{i,j} | \theta) \right)$$

where $\eta_{i,j}$ is the jth sample from the conditional posterior $\eta_i \mid \theta', x_i, y_i$. The samples are sampled using Markov Chain Monte Carlo (MCMC). So SAEM is learning and using the conditional posterior!

Laplace Method

$$
\begin{aligned}
p(y \mid \theta, x) &= \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i \\
&= \prod_{i=1}^{3} \int_{\eta_i} p(y_i, \eta_i \mid \theta, x_i) d\eta_i
\end{aligned}
$$

When we have conditional idenfiability wrt η_i given θ, we can also use the Laplace method to approximate the above integrals around the conditional modes η_i^*.

Laplace Method

$$
\begin{aligned}
\text{obj} = p(y \mid \theta, x) &= \prod_{i=1}^{3} \int_{\eta_i} p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) d\eta_i \\
&= \prod_{i=1}^{3} \int_{\eta_i} p(y_i, \eta_i \mid \theta, x_i) d\eta_i \\
&\approx \prod_{i=1}^{3} \left(p(y_i \mid \theta, \eta_i^*, x_i) \cdot p(\eta_i^* \mid \theta) \cdot \sqrt{(2\pi)^K / |-H_i|} \right)
\end{aligned}
$$

where K is the dimension of η_i, and $|-H_i| = \left| -\frac{d^2 \log p(y_i, \eta_i | \theta, x_i)}{d\eta_i d\eta_i'} \right|_{\eta_i = \eta_i^*}$.

Laplace Method

A Gaussian approximation of the conditional posterior of η_i around the mode η_i^* can be derived from the Laplace integration method:

$$\eta_i \mid \theta, x_i, y_i \sim N(\eta_i^*, -H_i^{-1})$$

$$p(\eta_i \mid \theta, x_i, y_i) \approx \sqrt{\frac{|-H_i|}{(2\pi)^K}} e^{\frac{1}{2}(\eta_i - \eta_i^*)^T H_i (\eta_i - \eta_i^*)}$$

where $-H_i$ is the precision matrix and $-H_i^{-1}$ is the covariance matrix of the multivariate Gaussian.

Laplace Method

$$\text{obj} \approx \prod_{i=1}^{3} \left(p(y_i \mid \theta, \eta_i^*, x_i) \cdot p(\eta_i^* \mid \theta) \cdot \sqrt{(2\pi)^K / |-H_i|} \right)$$

Conclusions

Recall: we are trying to figure out how to fit NLME models to data in a way that makes sense by taking advantage of the random effects or latent variables. **Summary**:

- Joint MLE was ruled out because it ignored the hierarchical structure
- Joint MAP was ruled out because it can lead to degenerate solutions in some cases
- The marginal likelihood penalizes degenerate weak priors
- Marginal likelihood maximization algorithms learn and use (an approx. of) the conditional posteriors $\eta_i \mid \theta, x_i, y_i$