

Embedding Categorical Data for Pricing

Ronald Richman FIA FASSA CERA CPCU

Chief Actuary – Old Mutual Insure

15 August 2024

Background

- Joint work with Mario Wüthrich
- Paper: High-cardinality categorical covariates in network regressions
- Available for free from:
<https://link.springer.com/article/10.1007/s42081-024-00243-4>
- Code: <https://github.com/wueth/High-Cardinality-Covariates-Regularization>

Japanese Journal of Statistics and Data Science
<https://doi.org/10.1007/s42081-024-00243-4>

ORIGINAL PAPER

Risk and Statistics in Actuarial Science



High-cardinality categorical covariates in network regressions

Ronald Richman¹ · Mario V. Wüthrich² 

Received: 23 August 2023 / Revised: 8 December 2023 / Accepted: 12 January 2024
© The Author(s) 2024

Abstract

High-cardinality (nominal) categorical covariates are challenging in regression modeling, because they lead to high-dimensional models. For example, in generalized linear models (GLMs), categorical covariates can be implemented by dummy coding which results in high-dimensional regression parameters for high-cardinality categorical covariates. It is difficult to find the correct structure of interactions in high-cardinality covariates, and such high-dimensional models are prone to over-fitting. Various regularization strategies can be applied to prevent over-fitting. In neural network regressions, a popular way of dealing with categorical covariates is entity embedding, and, typically, over-fitting is taken care of by exploiting early stopping strategies. In case of high-cardinality categorical covariates, this often leads to a very early stopping, resulting in a poor predictive model. Building on Avanzi et al. (ASTIN Bull, 2024), we introduce new versions of random effects entity embedding of categorical covariates. In particular, having a hierarchical structure in the categorical covariates, we propose a recurrent neural network architecture and a Transformer architecture, respectively, for random-effects entity embedding that give us very accurate regression models.

Outlook for the talk

- **Introduce methods for dealing with categorical embeddings...**
- **... with high cardinality = many levels**
- **Regularization to ensure our estimates have credibility**
- **Discuss how we will deal with categories with a natural hierarchy**
- **Tools used = deep learning, so will give brief introduction**
- **Show results on a simulated dataset**

Agenda

- What are embeddings?
- GLMMs
- Let's think Bayesian
- Hierarchical models based on embeddings

Pricing depends on categorical variables

- Most pricing datasets have a large number of categorical variables in addition to continuous variables
- Categorical variables may also arise from preprocessing continuous variables into categorical ones
- Simulated dataset used throughout as example

	VehUse	Town	DrivAge	Vehweight	VehPower	VehAge	VehBrand	VehModel	VehDetail	True
1:	0	1	51	1730	169	3	J	Jb	Jb4	0.13821475
2:	1	1	41	1760	249	2	K	Kd	Kd2	0.11266030
3:	0	1	25	1230	109	2	K	Kd	Kd2	0.15726383
4:	0	0	40	1010	84	9	A	Ad	Ad3	0.06964361
5:	0	0	43	2150	166	5	M	Mc	Mc4	0.11884314

199967:	1	0	46	1460	161	2	E	Ec	Ec3	0.09788983
199968:	0	0	49	1230	92	2	C	Ca	Ca4	0.06995247
199969:	1	0	38	990	107	5	F	Fb	Fb2	0.10659886
199970:	1	1	37	1300	158	3	S	Sd	Sd1	0.12993712
199971:	0	1	41	1100	91	3	A	Ad	Ad3	0.08195614

GLMs with factor levels $\sim$ embeddings

- Actuarial pricing often uses GLMs which has an explicit linear model form (on scale of link function)

$$g^{-1}(\hat{y}) = \beta_1 \cdot x_1 + \cdots + \beta_q \cdot x_q$$

- Suppose x_1 is categorical (or numerical and was binned) then we can represent the GLM as

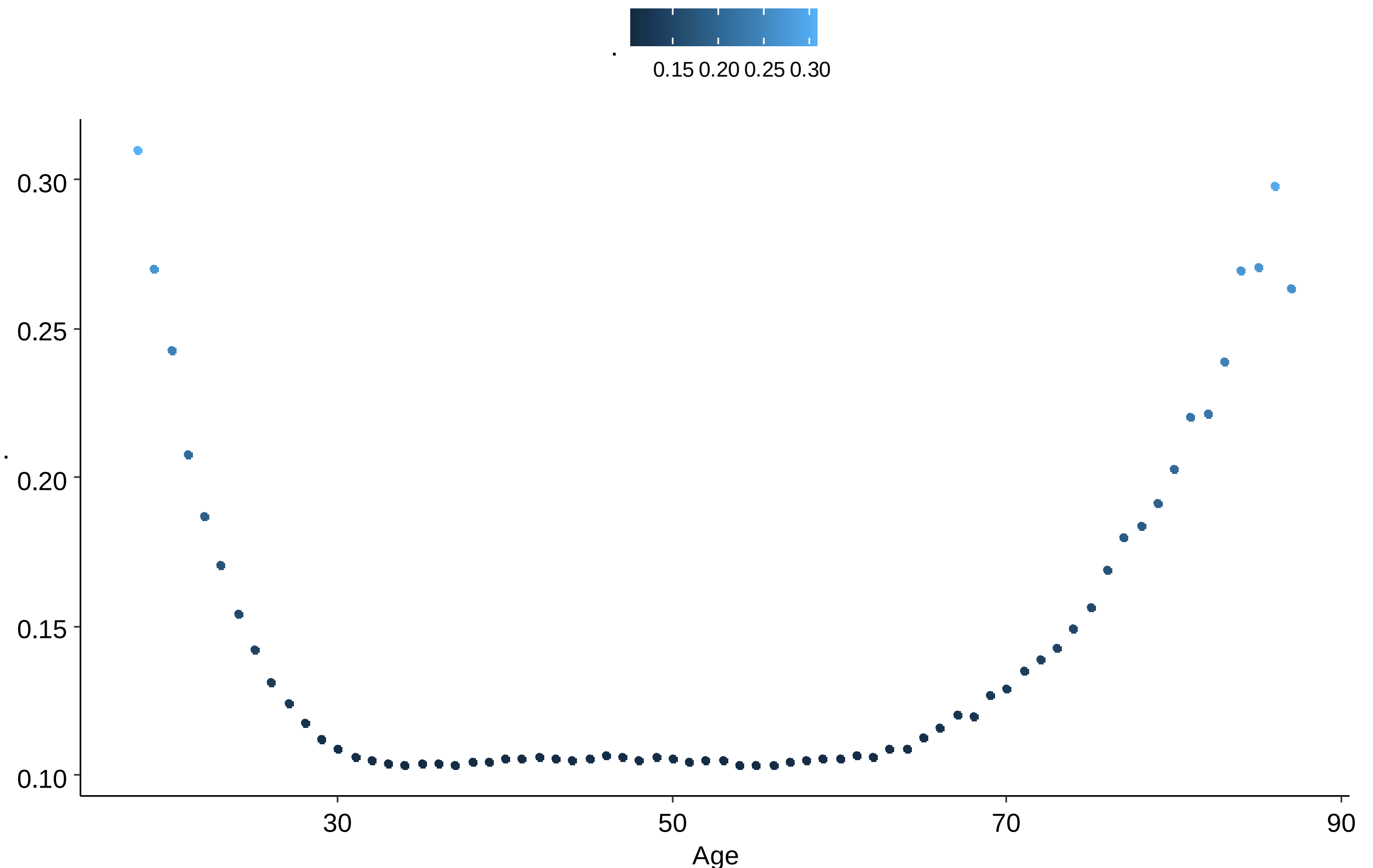
$$g^{-1}(\hat{y}) = \begin{cases} \beta_{1,1} + \cdots + \beta_q \cdot x_q & \text{when } x_1 = x_{1,1} \\ \beta_{1,2} + \cdots + \beta_q \cdot x_q & \text{when } x_1 = x_{1,2} \\ \vdots \\ \beta_{1,q_1} + \cdots + \beta_q \cdot x_q & \text{when } x_1 = x_{1,q_1} \end{cases}$$

- In a GLM, as we vary $x_{1,1} \rightarrow x_{1,2}$ our GLM value changes by $\beta_{1,2} - \beta_{1,1}$
- β coefficients are learned directly from the data...
- ... in deep learning, these are called embeddings
- This is a 1D embedding while we can generalize things to multiple dimensions

GLM example

- Very simple GLM predicting TRUE frequency based on driver age...
- ... produces GLM coefficients which are:
 - Pricing relativities
 - 1D embeddings
- Since output is frequency, coefficients can be interpreted on frequency scale
- In this case, we have treated Age as categorical whereas we could also have treated it as continuous =>
- In that case received a single coefficient
- We will be working with deep learning models and higher dimensional embeddings

```
fit = glm(True ~ as.factor(DrivAge)-1, data = dat)
coefs = fit$coefficients %>% data.table()
```

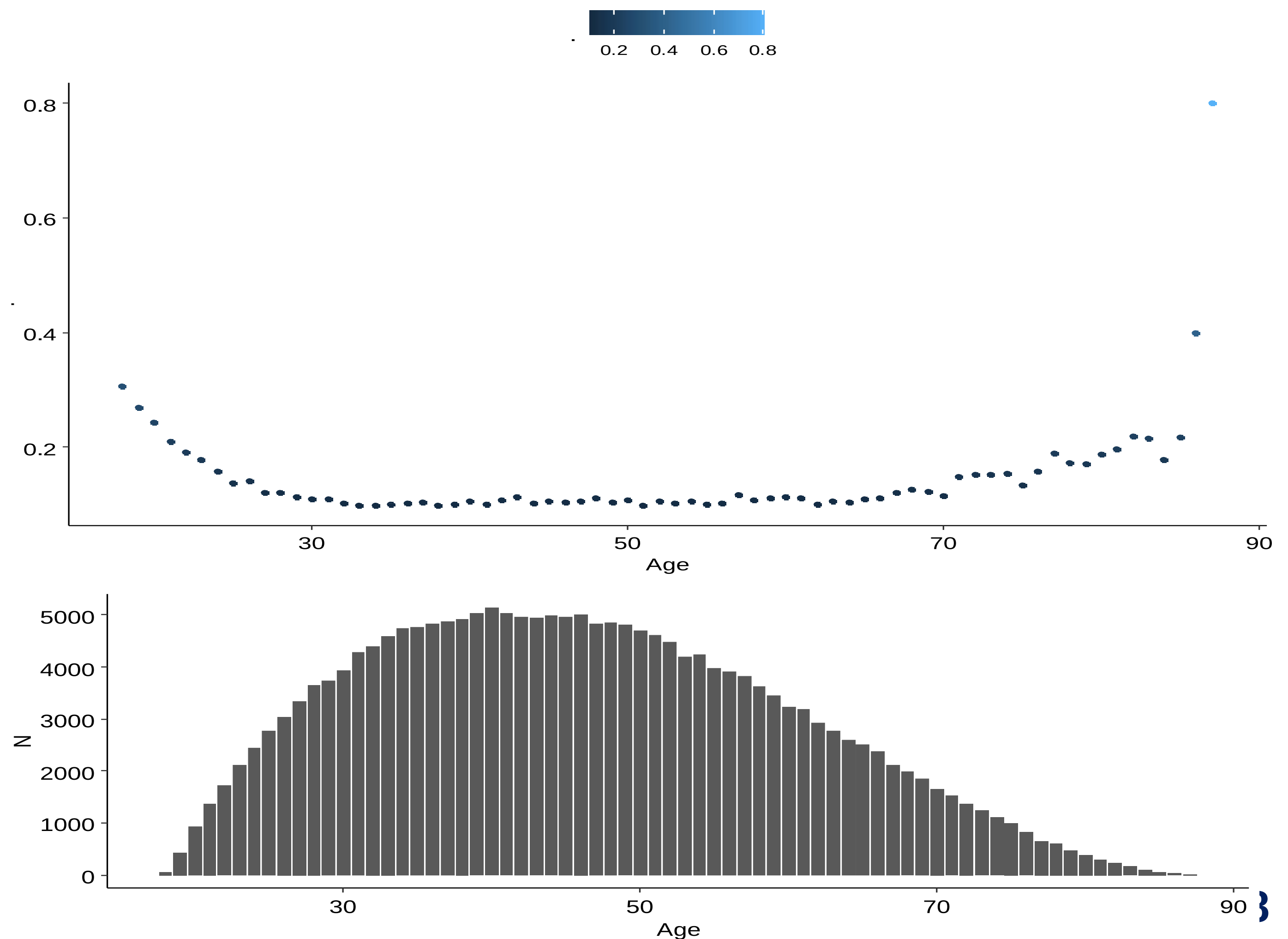


GLM example (2)

- We do not know the true rates, so need to approximate from very noisy experience data
- Example – where we have little exposure, we will struggle to get accurate views of rates
- => may want to smooth rates out OR
- Apply regularization – main theme of the paper
- OR can smooth by hand but:

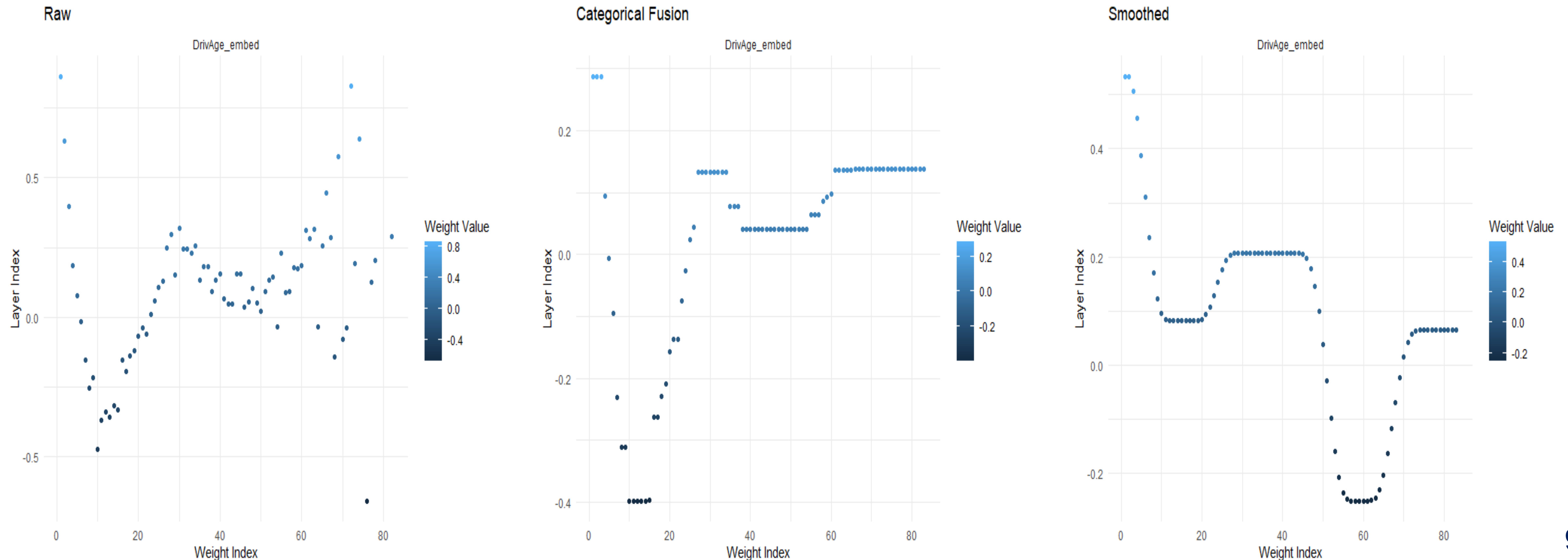
Highly manual and time consuming
Depends on expert judgement

```
fit = glm(ClaimNb ~ as.factor(DrivAge)-1, data = dat, family = poisson(link = "log"))  
coefs = fit$coefficients %>% exp %>% data.table()
```



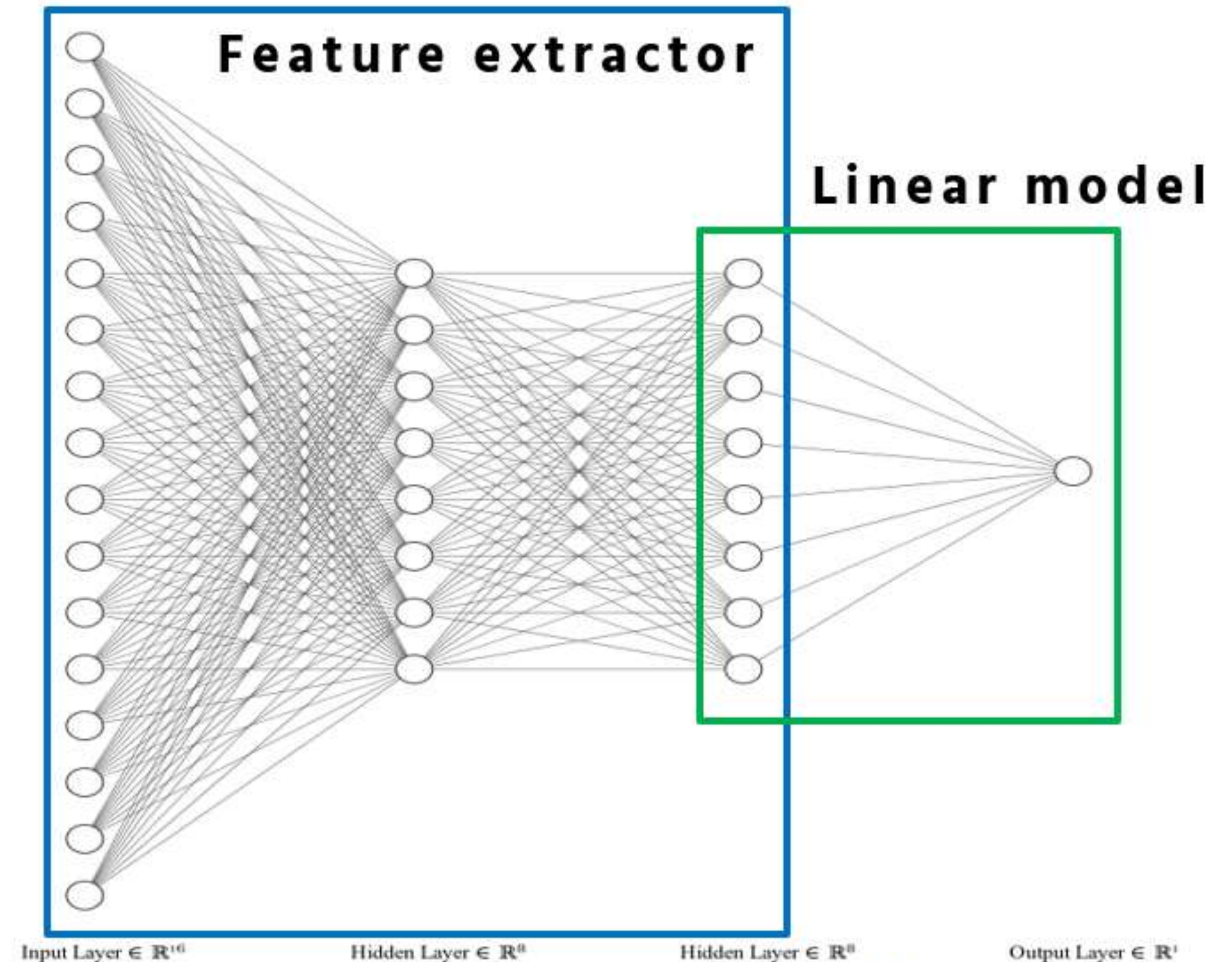
Two types of embeddings

- Numerical data that have been binned have a natural order – so can smooth/apply the Fused LASSO. Examples below.
- Categorical data does not have a natural order – so need to think about other approaches, usually similar to credibility theory in some way.



Deep Learning - FCN generalizes GLM

- Intermediate layers = representation learning, guided by supervised objective.
- Last layer = (generalized) linear model, where input variables = new representation of data
- No need to use GLM – strip off last layer and use learned features in, for example, XGBoost
- Or mix with traditional method of fitting GLM



Embedding Layer – Categorical Data

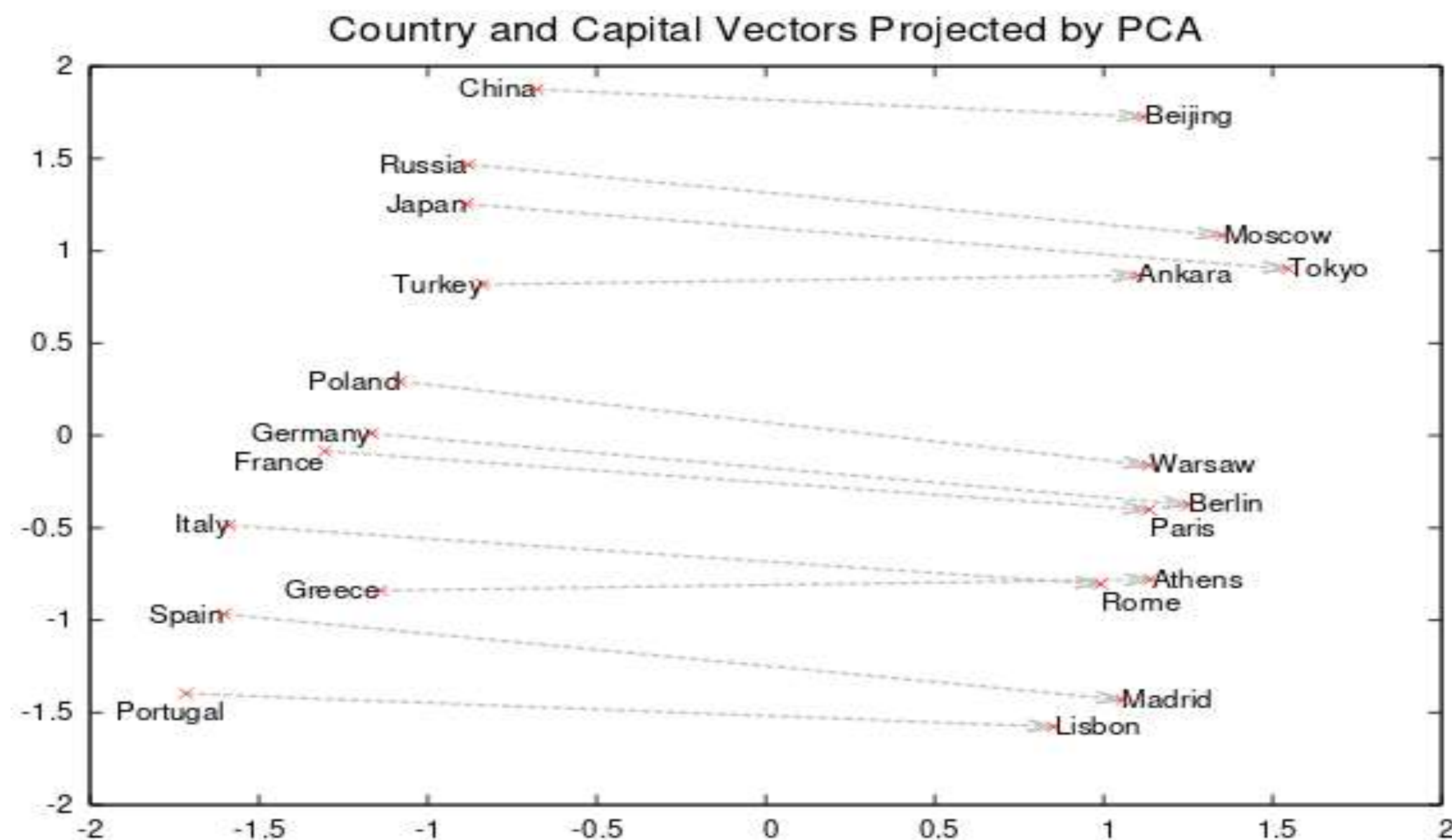
- **One hot encoding expresses the prior that categories are orthogonal => similar observations not categorized into groups**
- **Traditional actuarial solution – credibility**
- **Embedding layer prior – related categories should cluster together:**
 - Learns dense vector transformation of sparse input vectors and clusters similar categories together
 - Can pre-calibrate to MLE of GLM models, leading to CANN proposal of Wüthrich and Merz (2019)

	Actuary	Accountant	Quant	Statistician	Economist	Underwriter
Actuary	1	0	0	0	0	0
Accountant	0	1	0	0	0	0
Quant	0	0	1	0	0	0
Statistician	0	0	0	1	0	0
Economist	0	0	0	0	1	0
Underwriter	0	0	0	0	0	1

	Finance	Math	Statistics	Liabilities
Actuary	0.5	0.25	0.5	0.5
Accountant	0.5	0	0	0
Quant	0.75	0.25	0.25	0
Statistician	0	0.5	0.85	0
Economist	0.5	0.25	0.5	0
Underwriter	0	0.1	0.05	0.75

Modern ML relies on embeddings!

- New approaches for deep learning rely on learning embeddings for NLP and applying Transformers to these
- Approach proposed in 2017 relies on attention mechanisms
- Extended to other NLP tasks, computer vision and more recently, tabular data



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

Research Products Safety Company

ChatGPT Overview Team Enterprise Education Pricing

ChatGPT

Get answers. Find inspiration.
Be more productive.

Free to use. Easy to try. Just ask and ChatGPT can help with writing, learning, brainstorming, and more.

Start now ↗ Download the app >

How do Transformers work - attention

- Most actuarial models rely on fixed relationship between covariates (features) and outcomes; more flexible models (varying coefficient models) in statistical literature allow coefficients of models to vary with time (or other covariates)
- Generalized approach to allow for varying relationships between covariates and outcomes is attention (example from Xu et al., 2015)



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Intuitions for self-attention

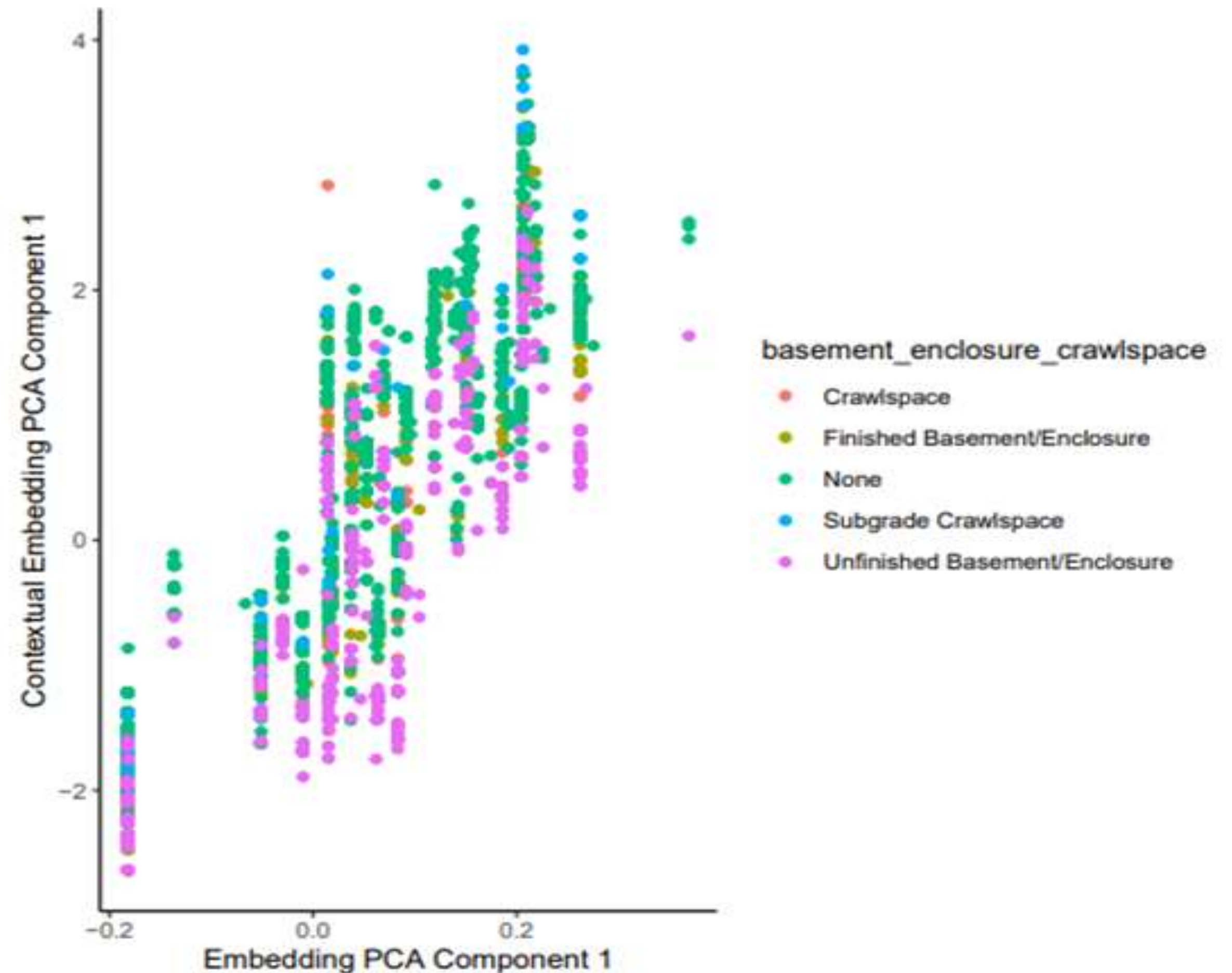
- When building models, relationships between covariates and outcomes may depend on context.
- Famous example in non-life insurance – young drivers and male drivers often have increased frequency, but young male drivers may experience even higher frequency => context allow for via interaction effect
- Automated method for building context into models: self-attention i.e. apply attention over inputs; sequence example (Cheng, Dong, & Lapata, 2016)

The diagram shows the sentence "The FBI is chasing a criminal on the run ." with ten rows of self-attention weights. Each row highlights a different word or phrase in blue, indicating its attention to other words in the sentence. The words "The", "FBI", "is", "chasing", "a", "criminal", "on", "the", and "run" are all highlighted in blue in every row, while the period "." is not. The words "The", "FBI", "is", "chasing", "a", "criminal", "on", "the", and "run" are also highlighted in red in every row, while the period "." is not. This illustrates how the model's attention shifts across the sentence to understand the context of each word.

The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

An actuarial example

- Transformer model recently applied in Kuo & Richman (2021) to model flood loss severity
- Model tries to predict damage ratio (proportion of exposure damaged) using several continuous and categorical covariates
- Best results achieved using Transformer based model
- Embeddings for categorical covariates + Self-attention for context = contextual embeddings
- Example shown for flood zone embeddings (left) colored according to house design variable (crawl space)

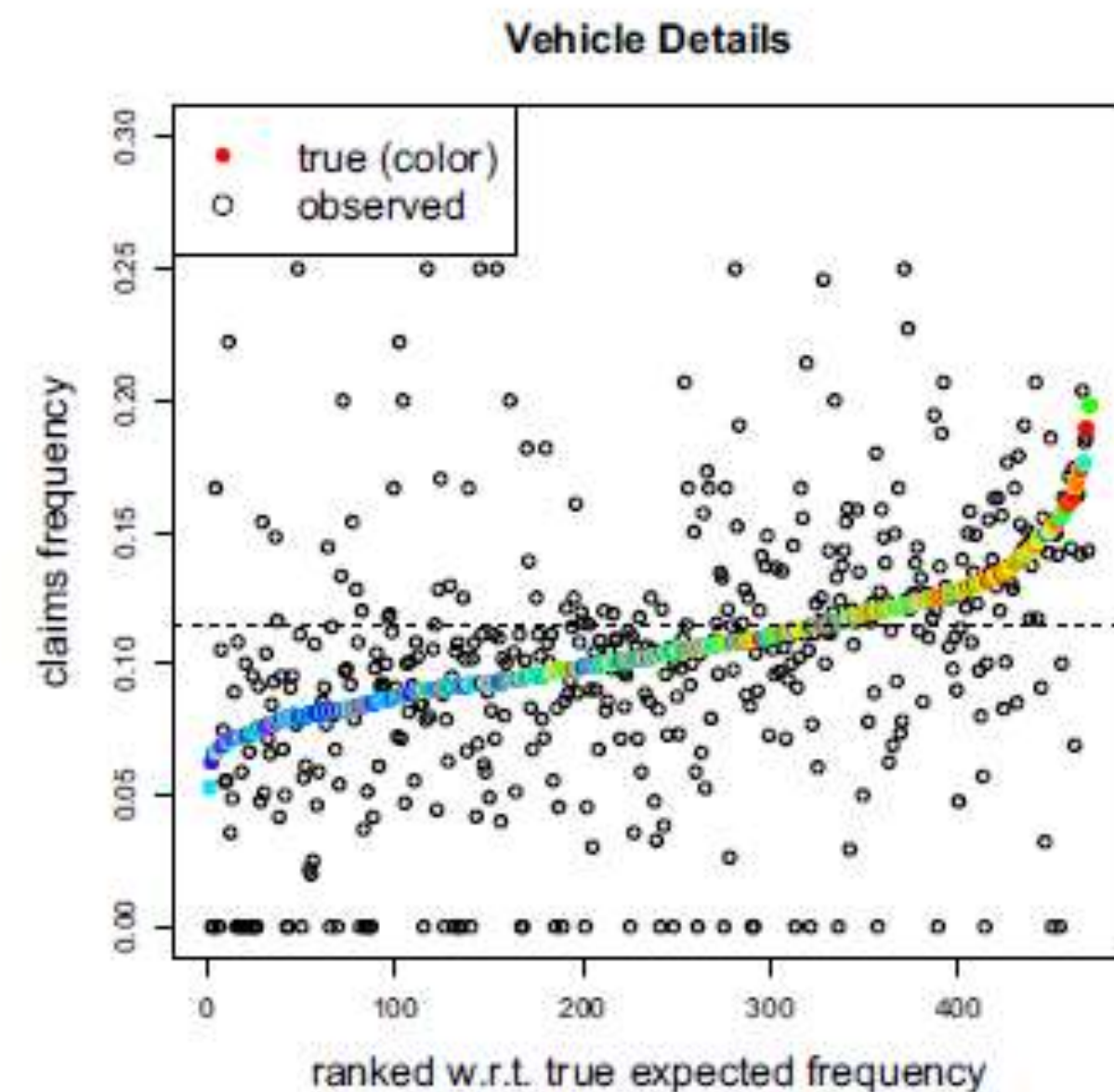
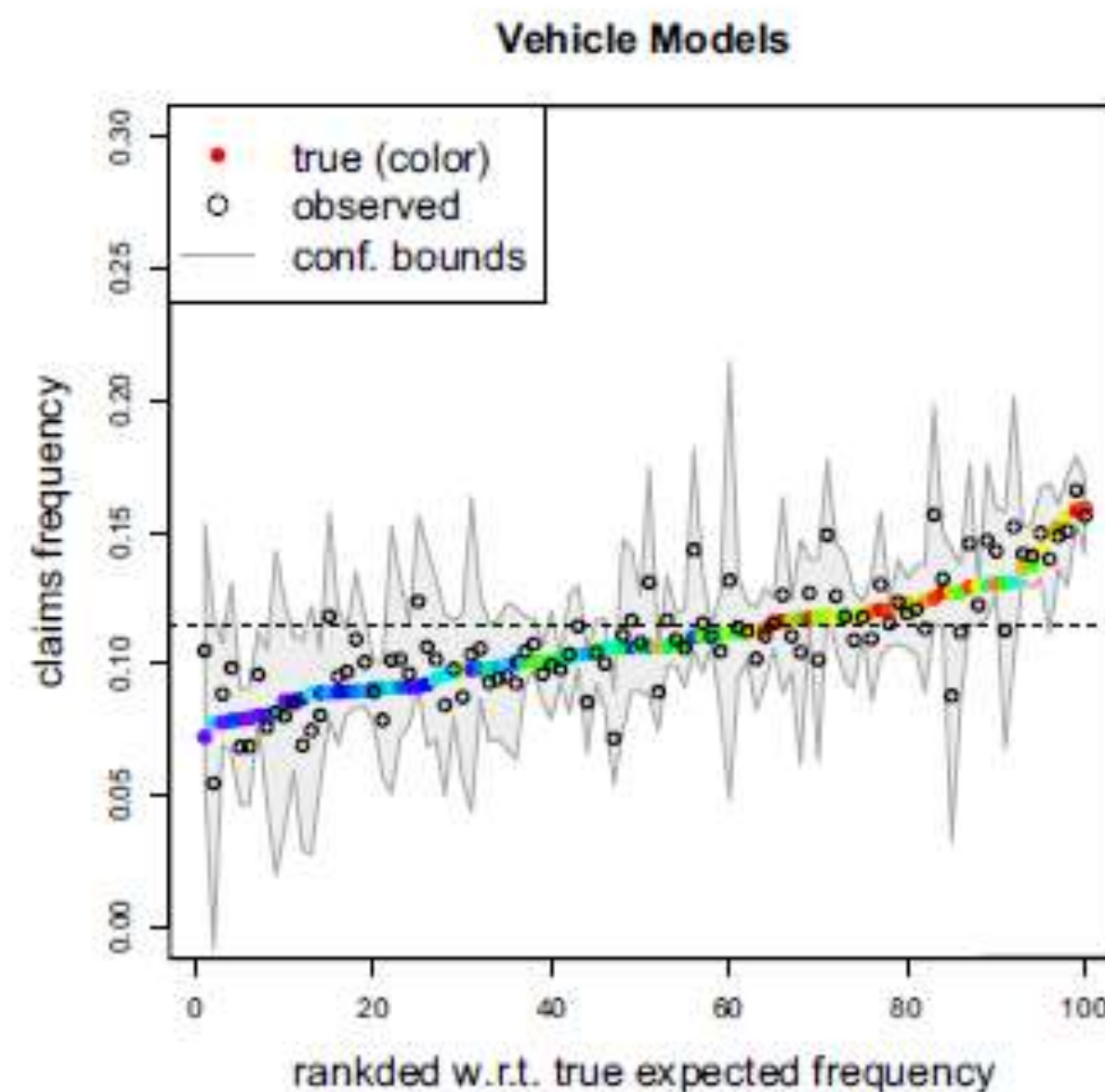
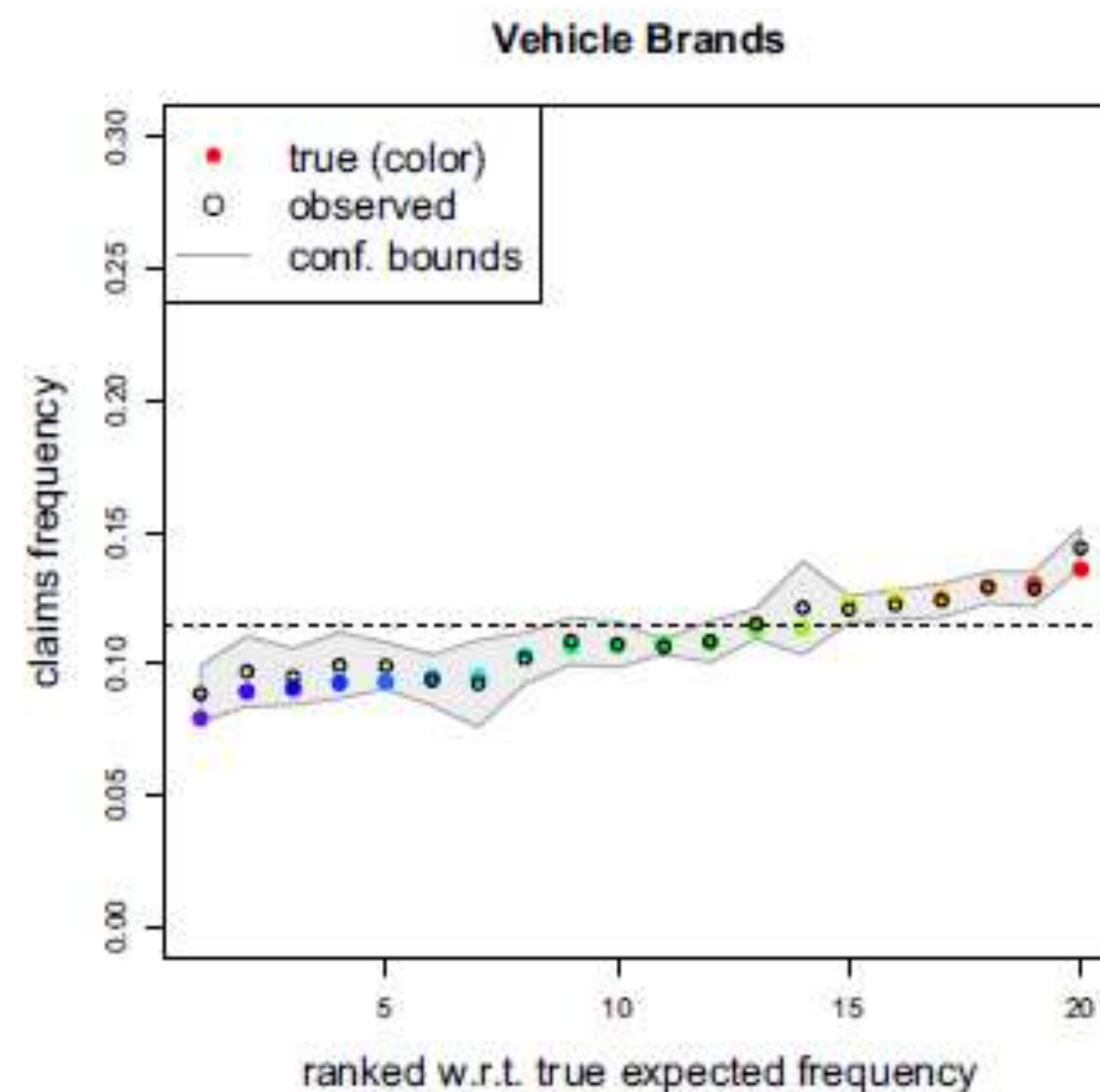


Agenda

- What are embeddings?
- GLMMs
- Let's think Bayesian
- Hierarchical models based on embeddings

What can go wrong with embeddings?

- We may have categorical variables where there are very few examples of each level
- E.g.: Car vehicle brand/model/variant – lots of Toyotas but very few Abarth's!
- Noisy data generated by claims process =>
- May not have enough credibility to set values of factor levels/embeddings appropriately
- Example from simulated data (colour = Vehicle Brand)

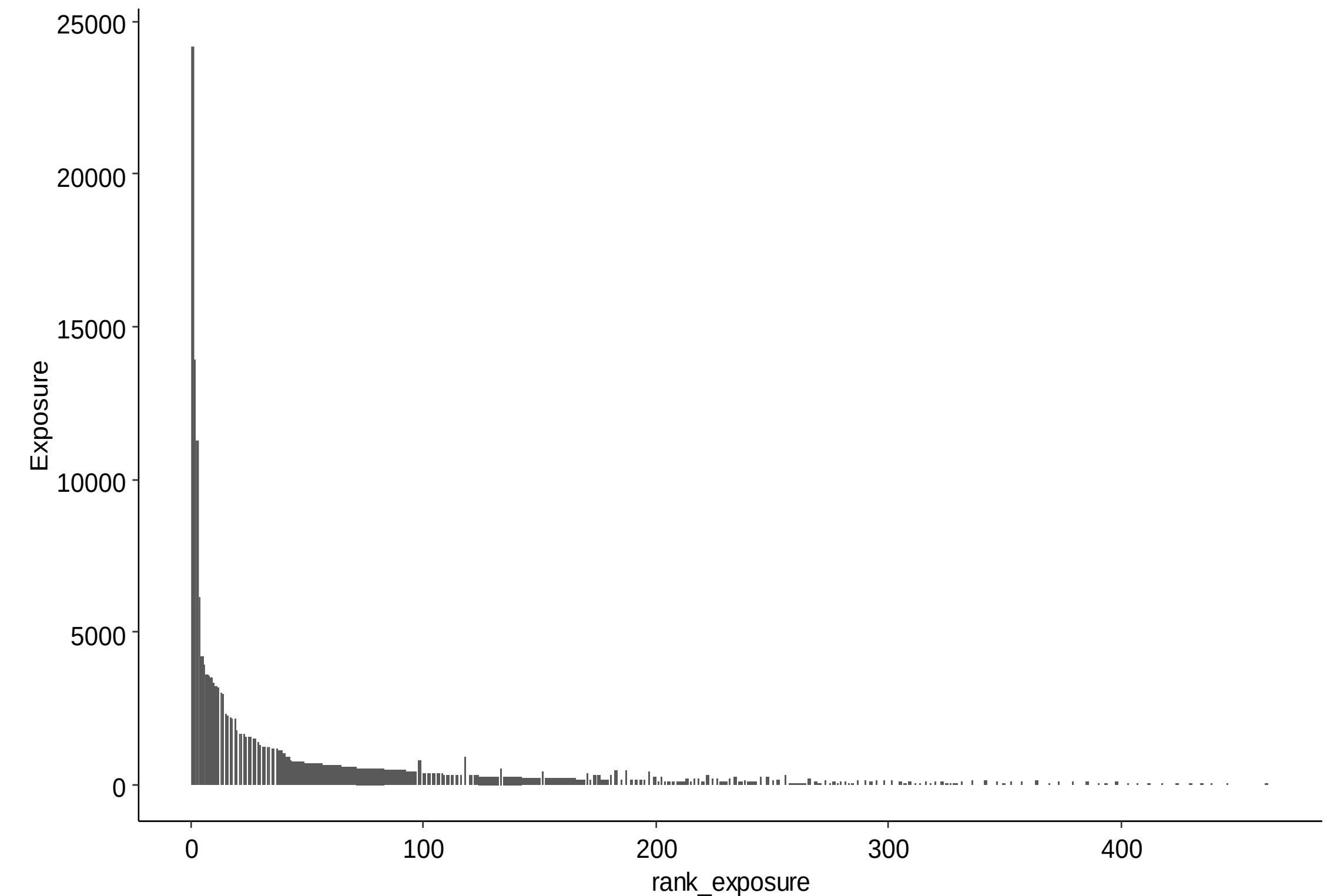
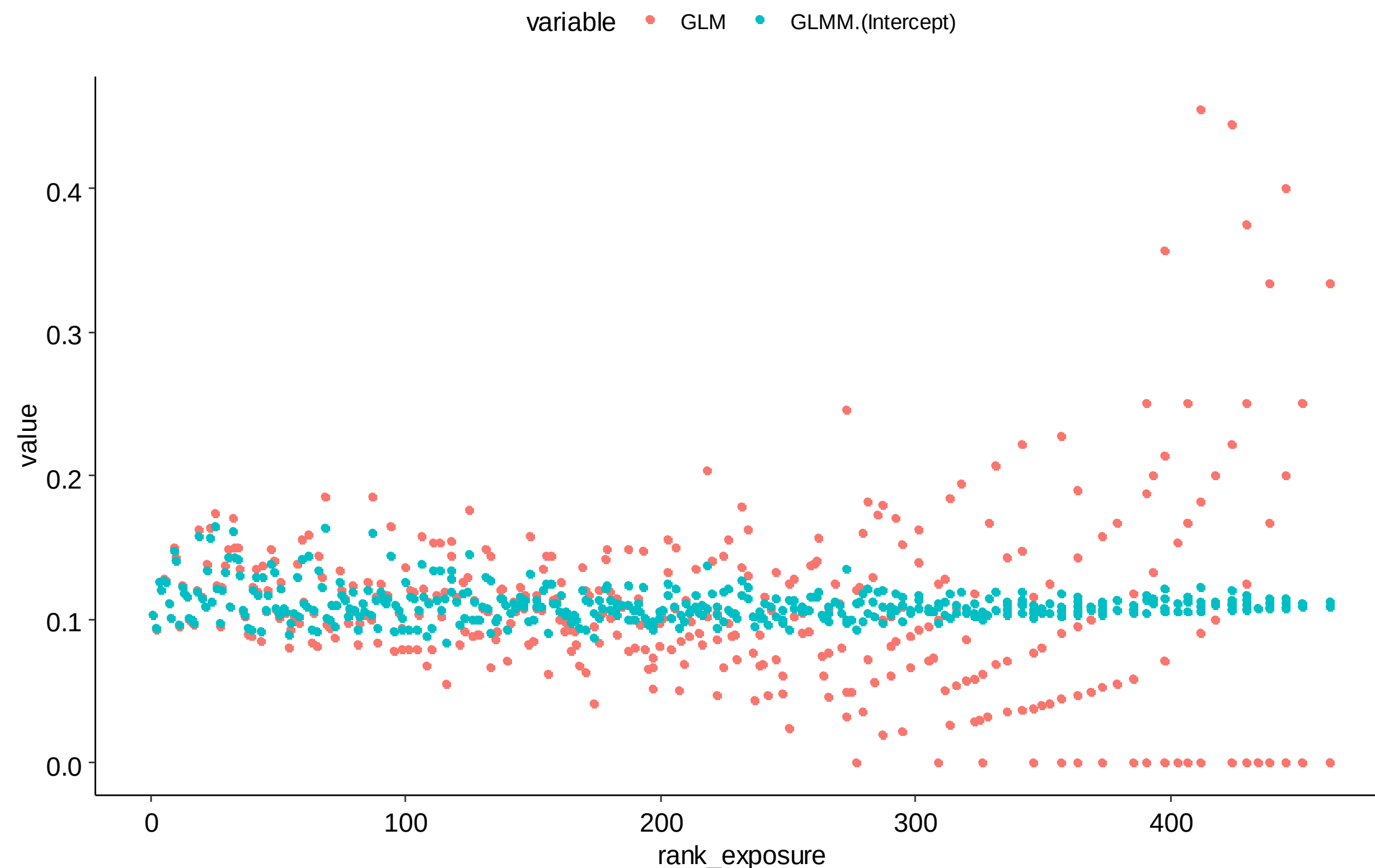


GLMMs offer a solution

- Classic actuarial solution to these problems – apply credibility theory

$$\beta_{credibility} = z * \bar{\beta} + (1 - z) * \beta_i$$

- Generalized Linear Mixed Models (GLMMs) incorporate Bühlmann-Straub credibility automatically to pool individual coefficients towards the mean



GLMMs offer a solution (2)

- What is the “mixed” model?
 - GLMMs can include “normal” regression coefficients that do not get a credibility treatment = **fixed effects**
 - In addition, coefficients with a credibility treatment = **random effects**
 - The two together = mixed model
- For more, see paper referenced below
- Disadvantages of GLMM framework:
 - Slow, especially for complex models
 - May not always converge
- How can we incorporate similar ideas into neural network models?

Generalized Linear Mixed Models for Ratemaking: A Means of Introducing Credibility into a Generalized Linear Model Setting

Fred Klinker, FCAS, MAAA

Abstract: GLMs that include explanatory classification variables with sparsely populated levels assign large standard errors to these levels but do not otherwise shrink estimates toward the mean in response to low credibility. Accordingly, actuaries have attempted to superimpose credibility on a GLM setting, but the resulting methods do not appear to have caught on. The Generalized Linear Mixed Model (GLMM) is yet another way of introducing credibility-like shrinkage toward the mean in a GLM setting. Recently available statistical software, such as SAS PROC GLIMMIX, renders these models more readily accessible to actuaries. This paper offers background on GLMMs and presents a case study displaying shrinkage towards the mean very similar to Buhlmann-Straub credibility.

Keywords: Credibility, Generalized Linear Models (GLMs), Linear Mixed Effects (LME) models, Generalized Linear Mixed Models (GLMMs).

Agenda

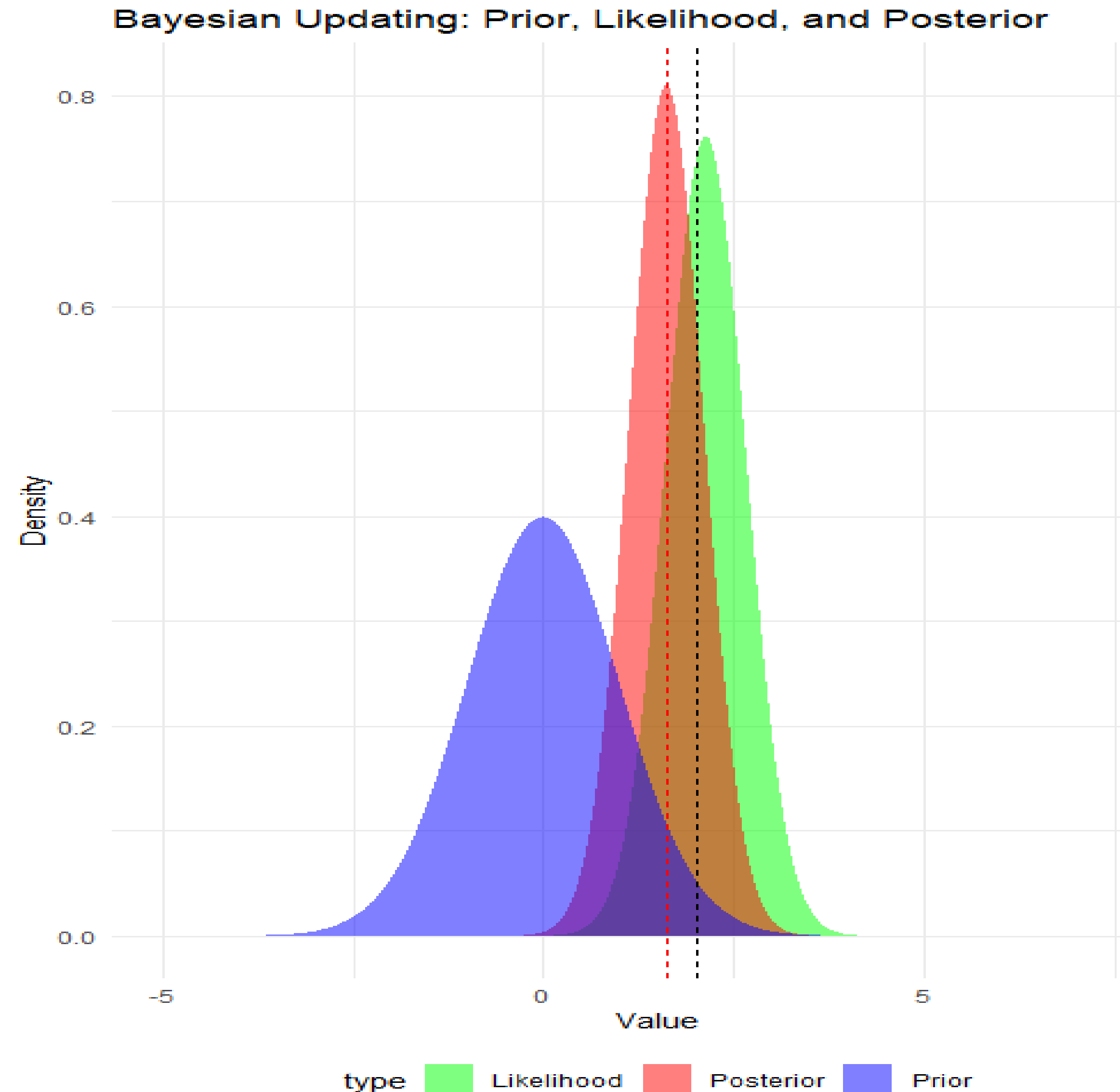
- What are embeddings?
- GLMMs
- Let's think Bayesian
- Hierarchical models based on embeddings

GLMMs and Bayesian thinking

- Bayesian explanation: models generally work by assuming that coefficients follow a normal distribution with mean = 0
=>
- Coefficients pulled towards zero unless enough posterior likelihood to support the value of the coefficient derived from Maximum Likelihood Estimation

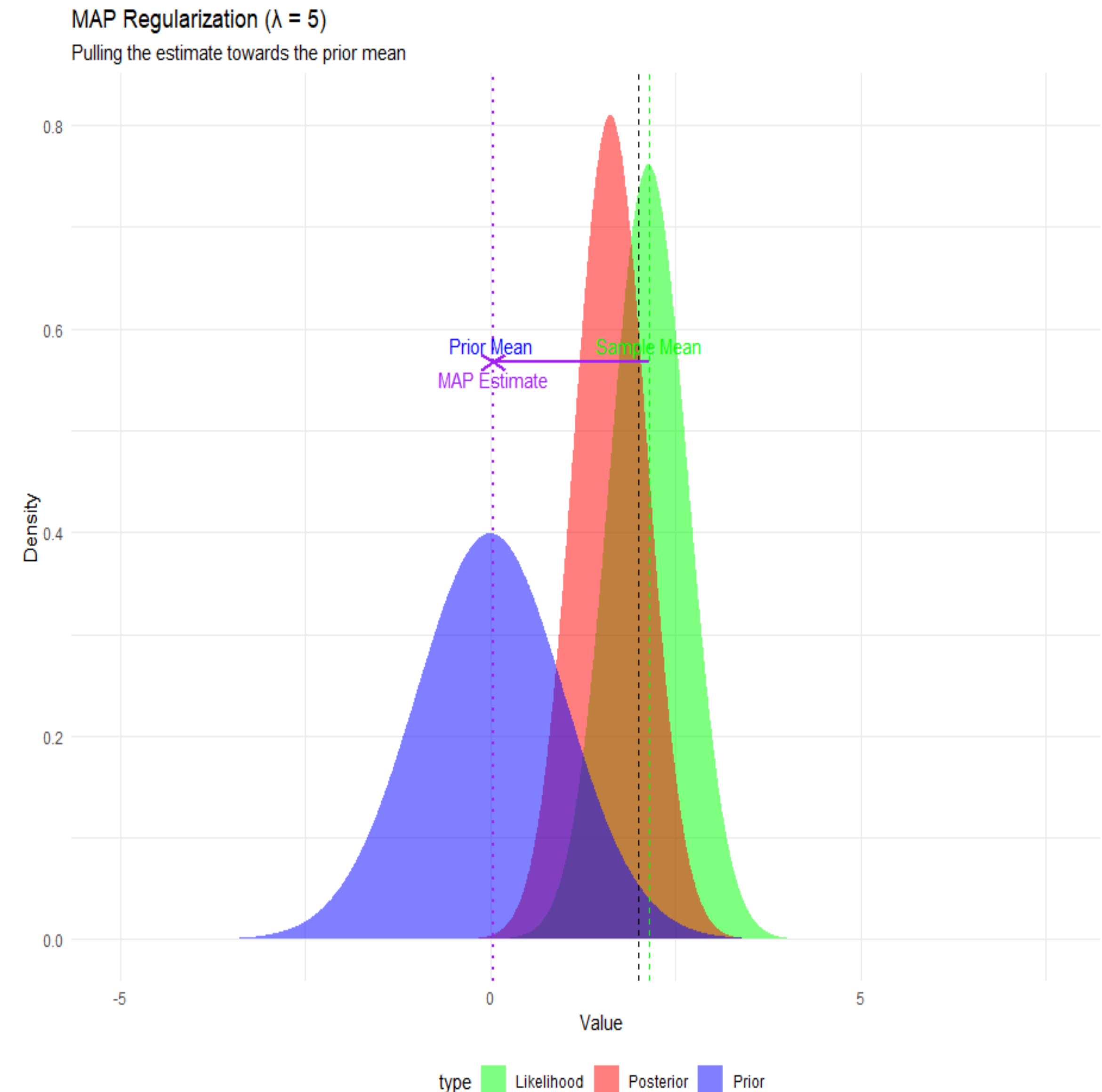
$$\log f_{\boldsymbol{\vartheta}}(\mathbf{Y}, \mathbf{U}) = \ell_{\mathbf{Y}}(\boldsymbol{\vartheta} | \mathbf{U}) + \log \pi(\mathbf{U})$$

- Can fit GLMMs using normal Bayesian methods such as MCMC...
- ... but these will be too slow for complex models such as neural networks
- Closed form solutions (i.e. credibility formula) only available in simple cases
- Need to rely on approximations to the Bayesian approach
 - Maximum a posteriori estimation
 - Variational inference



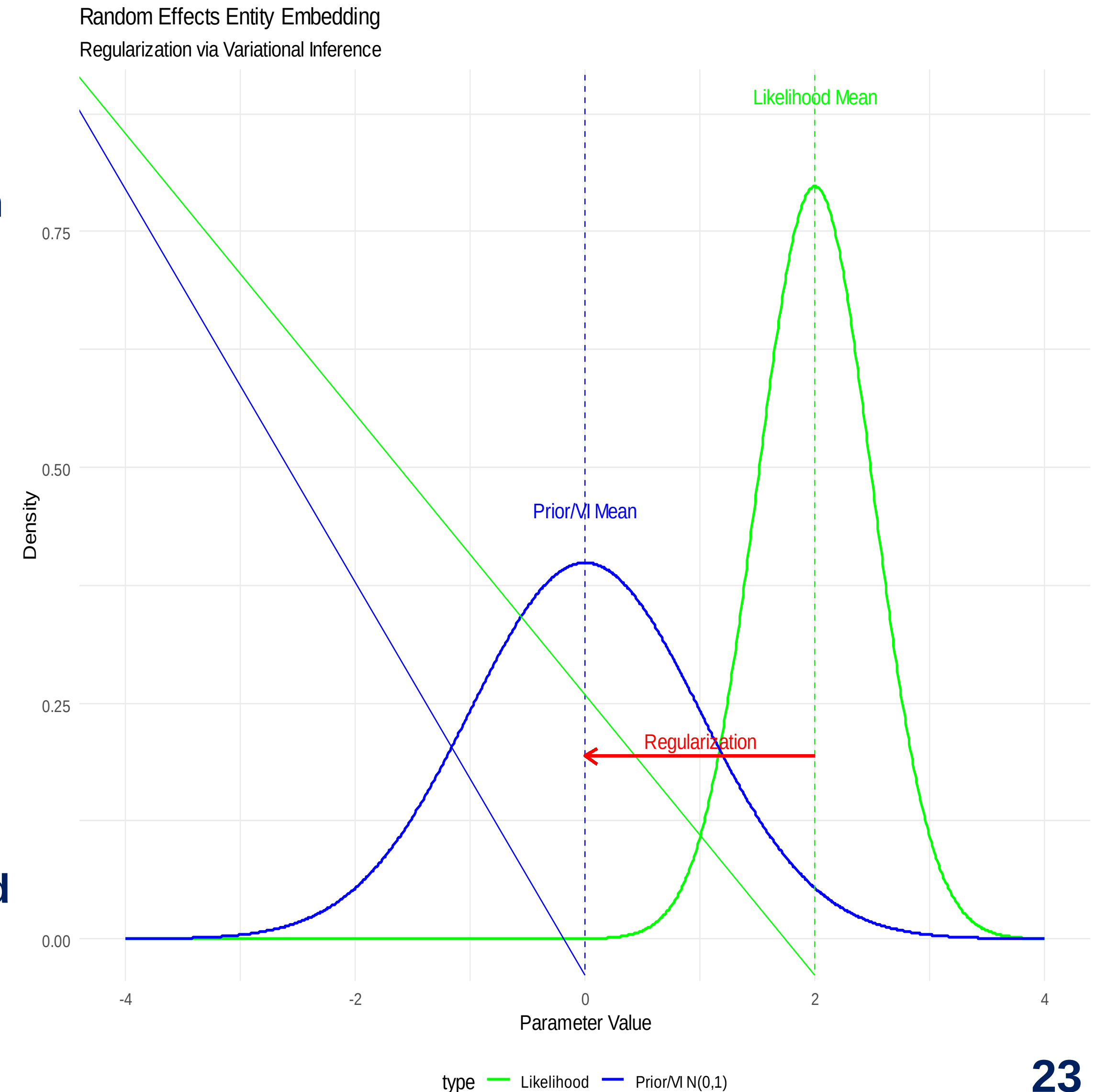
MAP estimation

- Equivalent to ridge regularization – i.e. L2 regularization
 - Adds a penalty term to the log-likelihood to force coefficients towards to mean of the prior function – which is 0
- $$\frac{1}{\varphi} \sum_{i=1}^n v_i \left[Y_i h(\text{NN}_{\vartheta}(\mathbf{x}_i, \mathbf{u}_{j[i]})) - \kappa(h(\text{NN}_{\vartheta}(\mathbf{x}_i, \mathbf{u}_{j[i]}))) - \frac{1}{w_{j[i]}} \frac{\varphi}{2\tau^2} \|\mathbf{u}_{j[i]}\|^2 \right],$$
- Regularization strength inversely proportional to case weights => low exposure means more regularization
 - Penalizes large embedding values for rare categories more strongly
 - Simple to implement and computationally efficient
 - Due to the flexibility of the neural network, we do not need to set a hyperparameter λ to control regularization strength
 - Just need to choose if we want regularization or not!



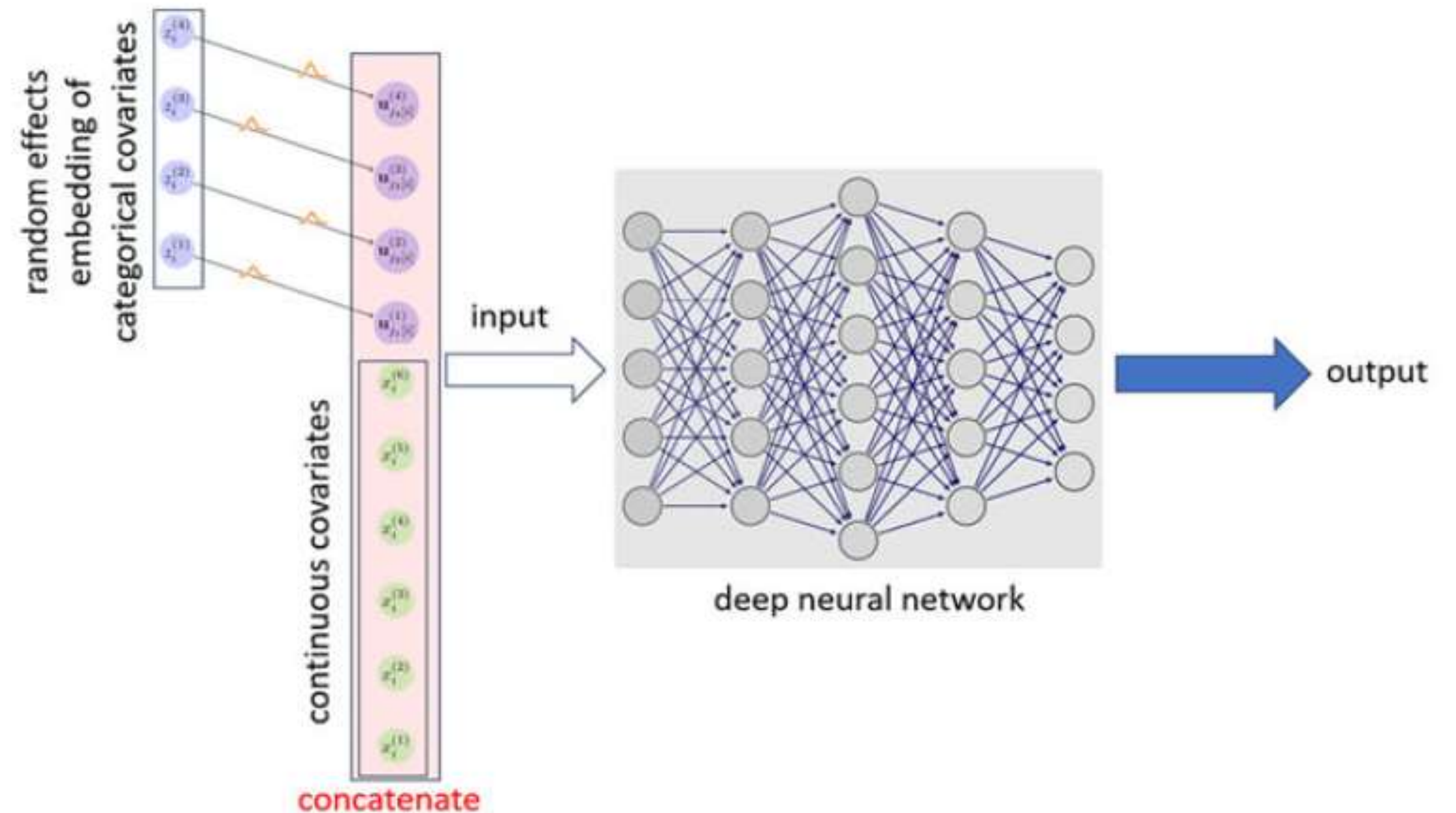
Variational Inference

- Assume we know the shape and have an idea of the posterior
- VI forces the coefficients to follow the posterior distribution => approximates the true posterior distribution of embeddings
- Gaussian prior + likelihood => gaussian posterior
- Involves optimizing both the model parameters and variational parameters
- Provides uncertainty estimates for embeddings
- More computationally intensive than MAP as it is fit using Monte Carlo sampling
- Requires selection of prior uncertainty parameter using grid search
- Can capture more complex posterior distributions



Putting it together

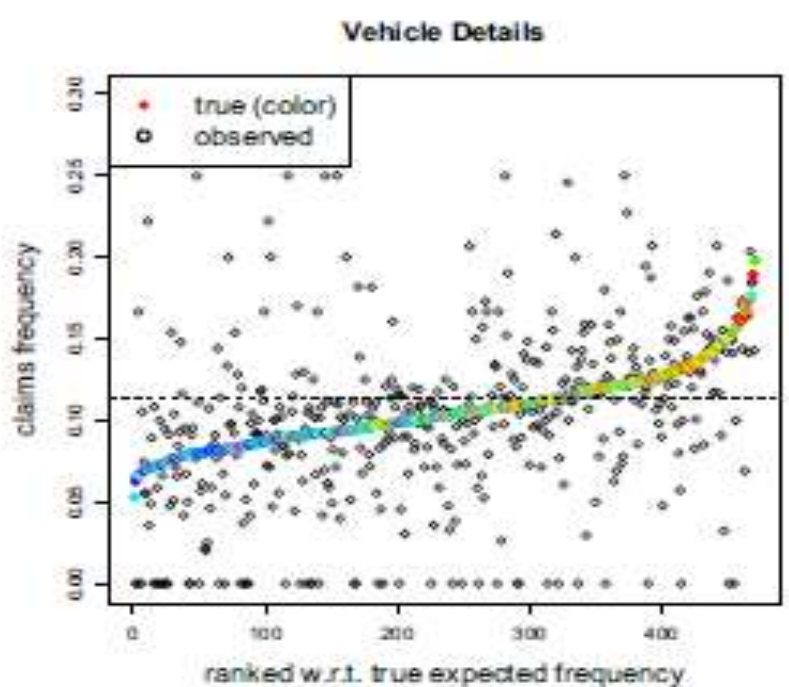
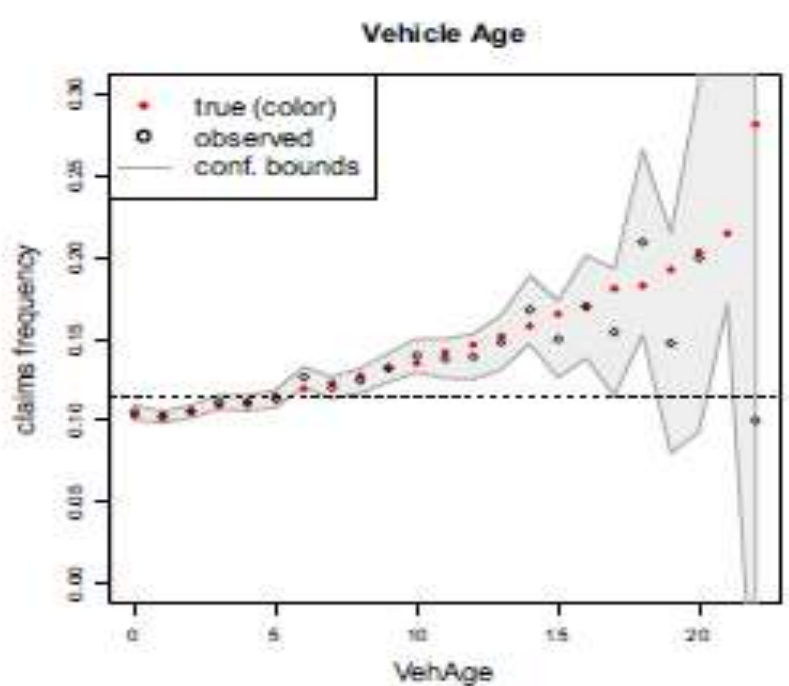
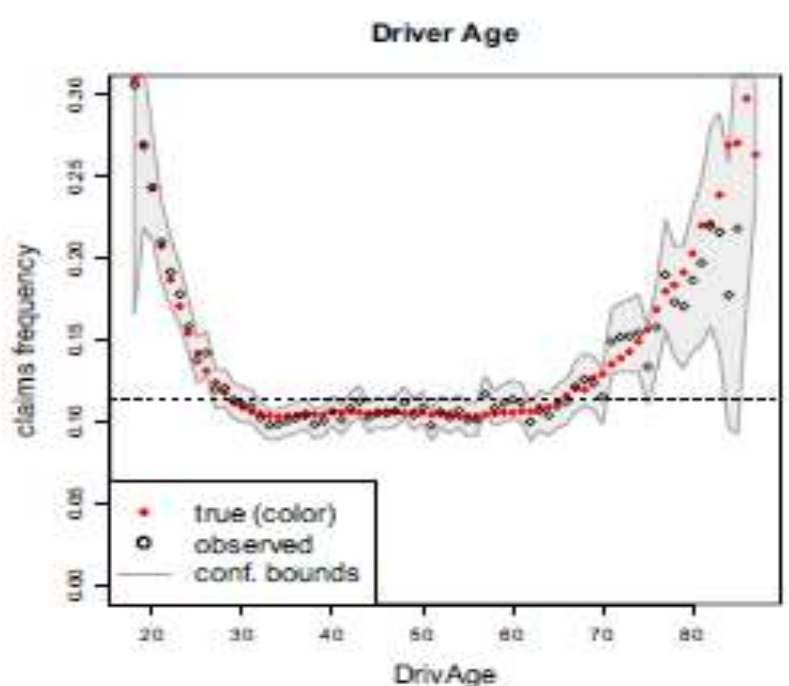
- Embeddings can be used to capture complex high dimensional information about categorical data...
- ... with the risk the we overfit to noise, particularly when there is high cardinality
- Developed Bayesian methods to credibility weight embeddings towards zero, which depend on exposure
- Incorporate these into a deep neural network so that both continuous and categorical variables can contribute to network predictions
- Diagram of network used shown on right



Results

- Recall that we know the true frequency from the simulated dataset
- Fit basic neural network with embeddings (2D), another proposed approach to incorporating embeddings (GLMMnet) and LightGBM. Note that networks suffer with VehDetail!

		Average KL divergence		
		Network (2.30)	GLMMNet (4.3)	LightGBM
(0)	Null model (empirical mean)	1.0342	1.0342	1.0342
(0)	w/o categorical covariates	0.3947	0.3970	0.3958
(1)	With VehBrand	0.2622	0.3076	0.2763
(1)	With VehModel	0.2188	0.2961	0.2499
(1)	With VehDetail	0.2615	0.3292	0.2240
(2)	With VehBrand, VehModel	0.2312	0.2958	0.2618
(3)	With VehBrand, VehModel, VehDetail	0.2694	0.3305	0.2191

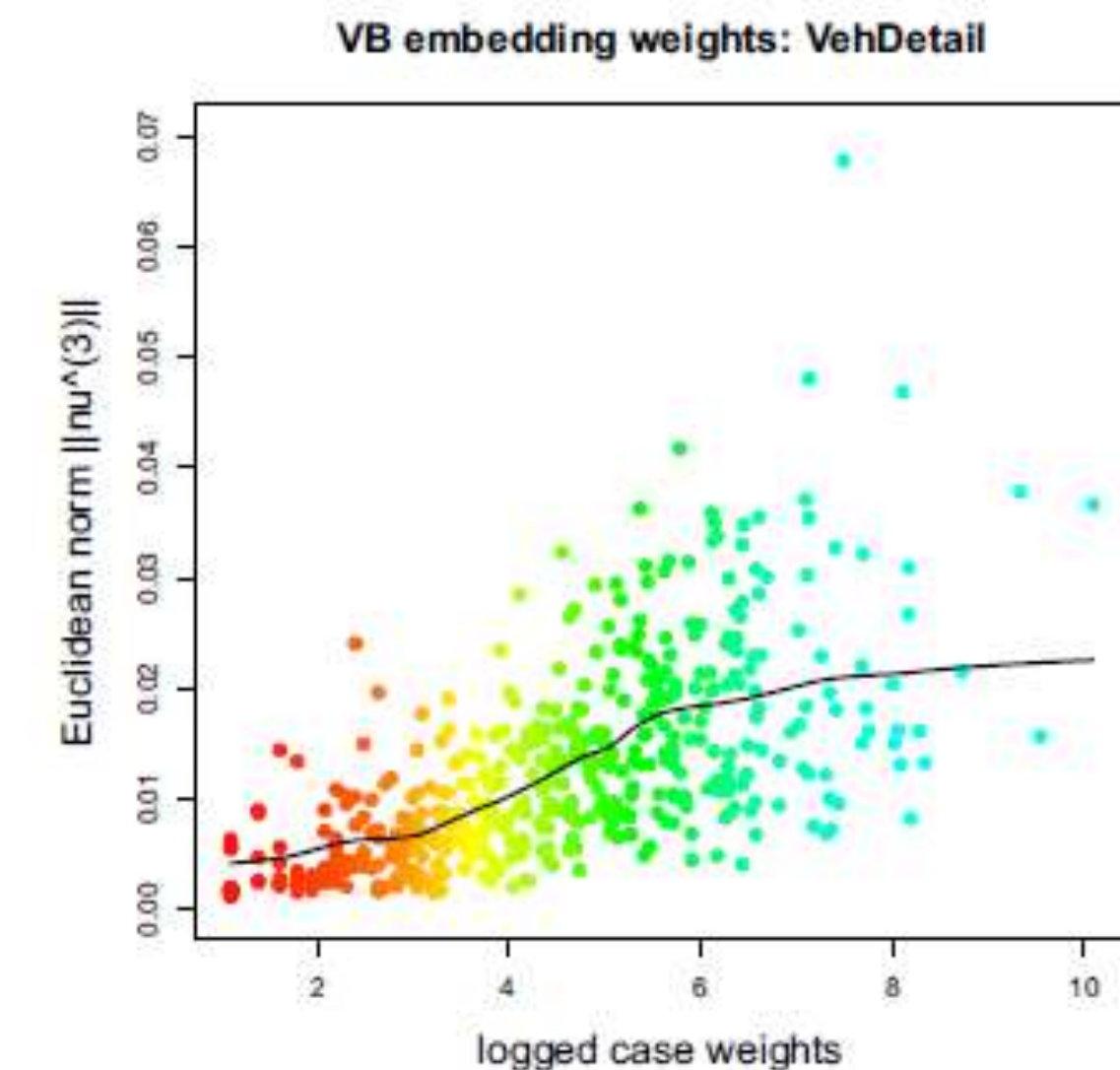
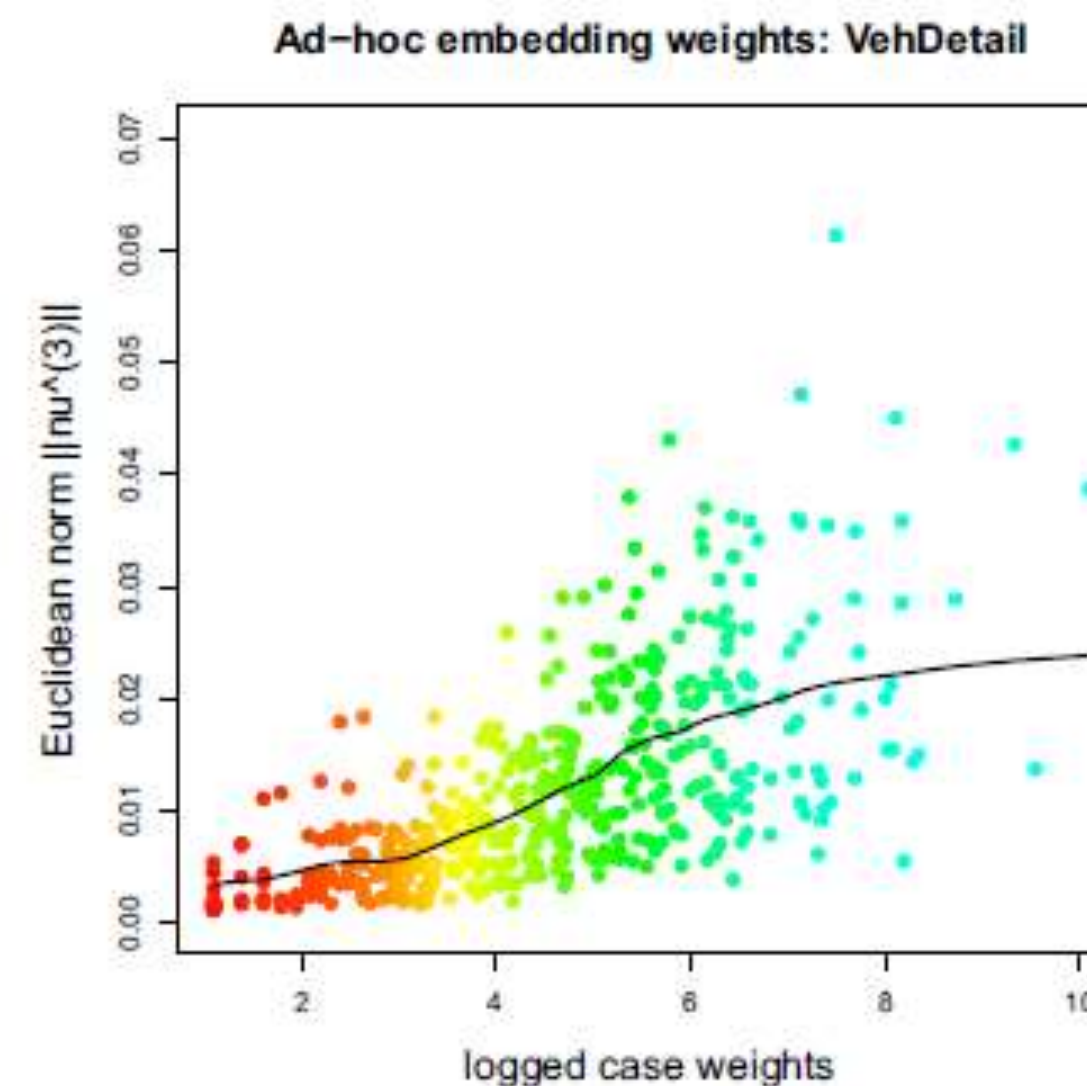
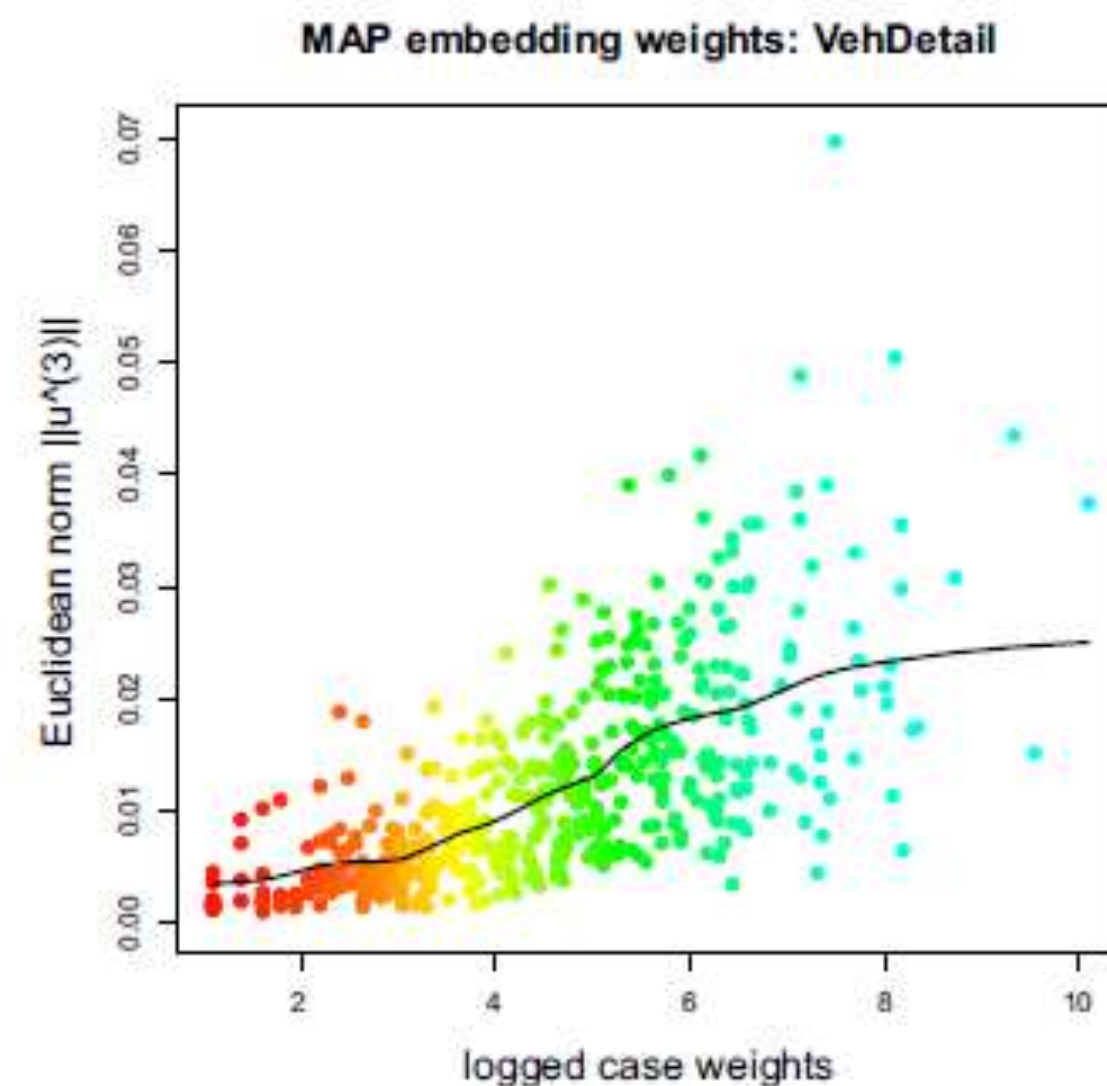
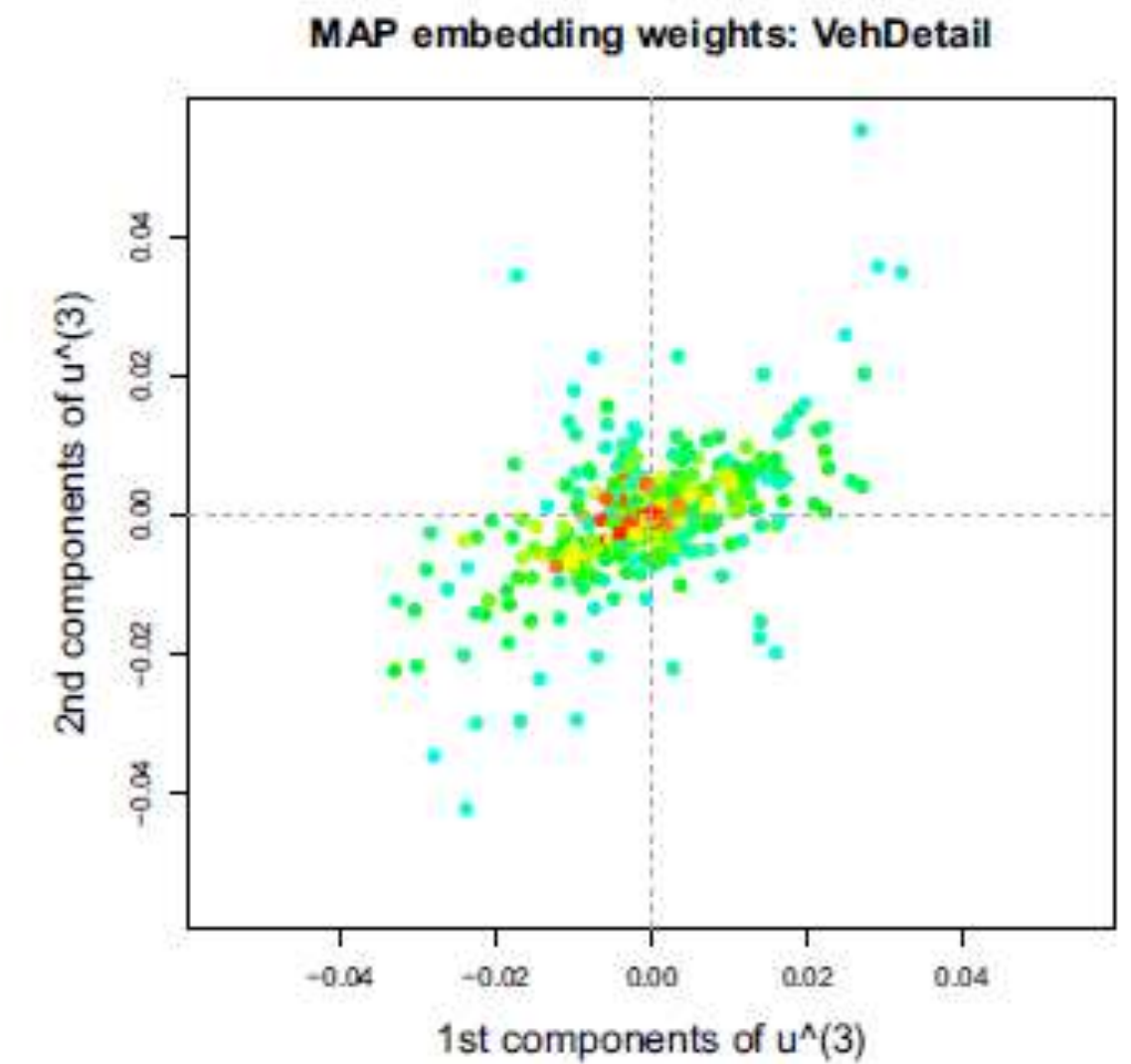
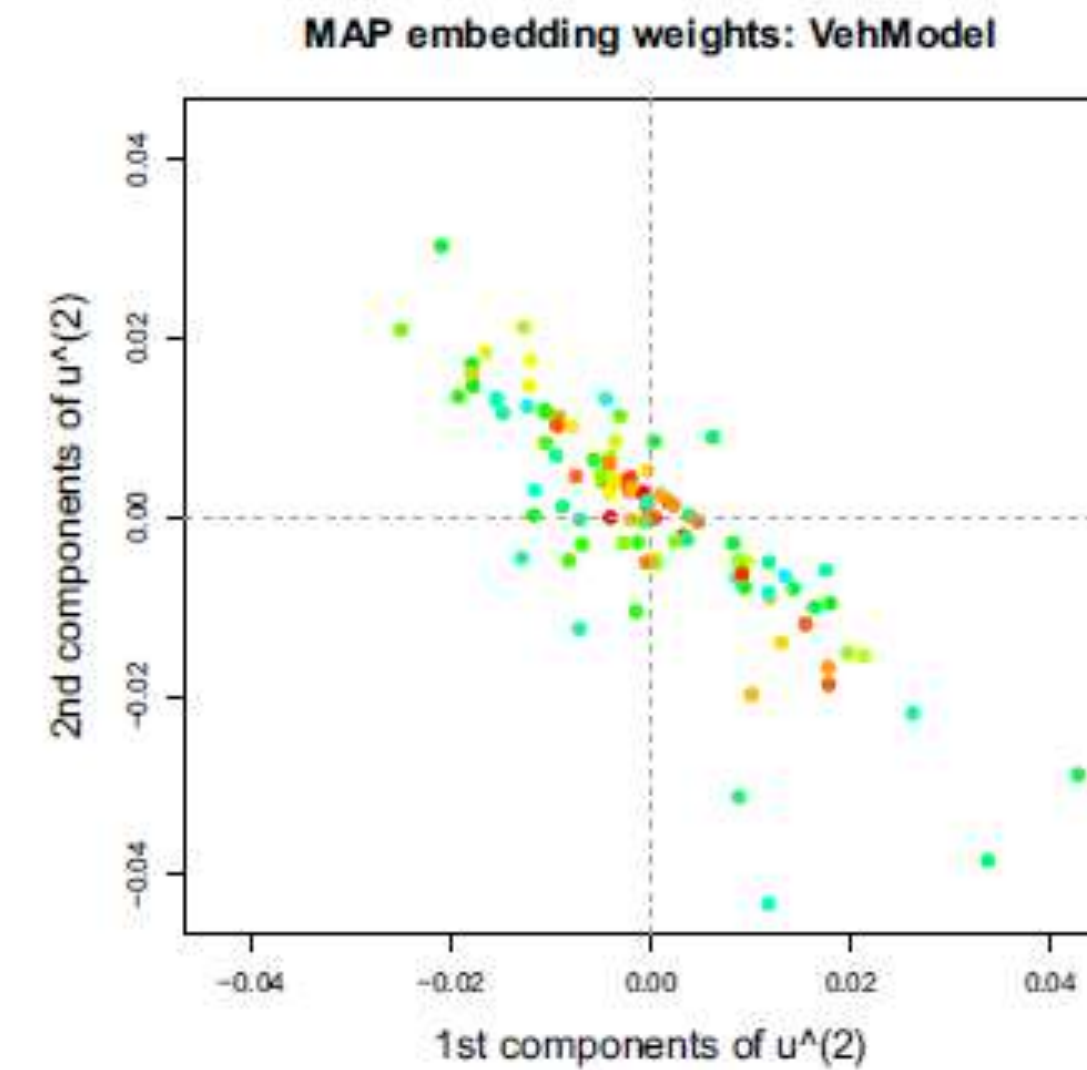
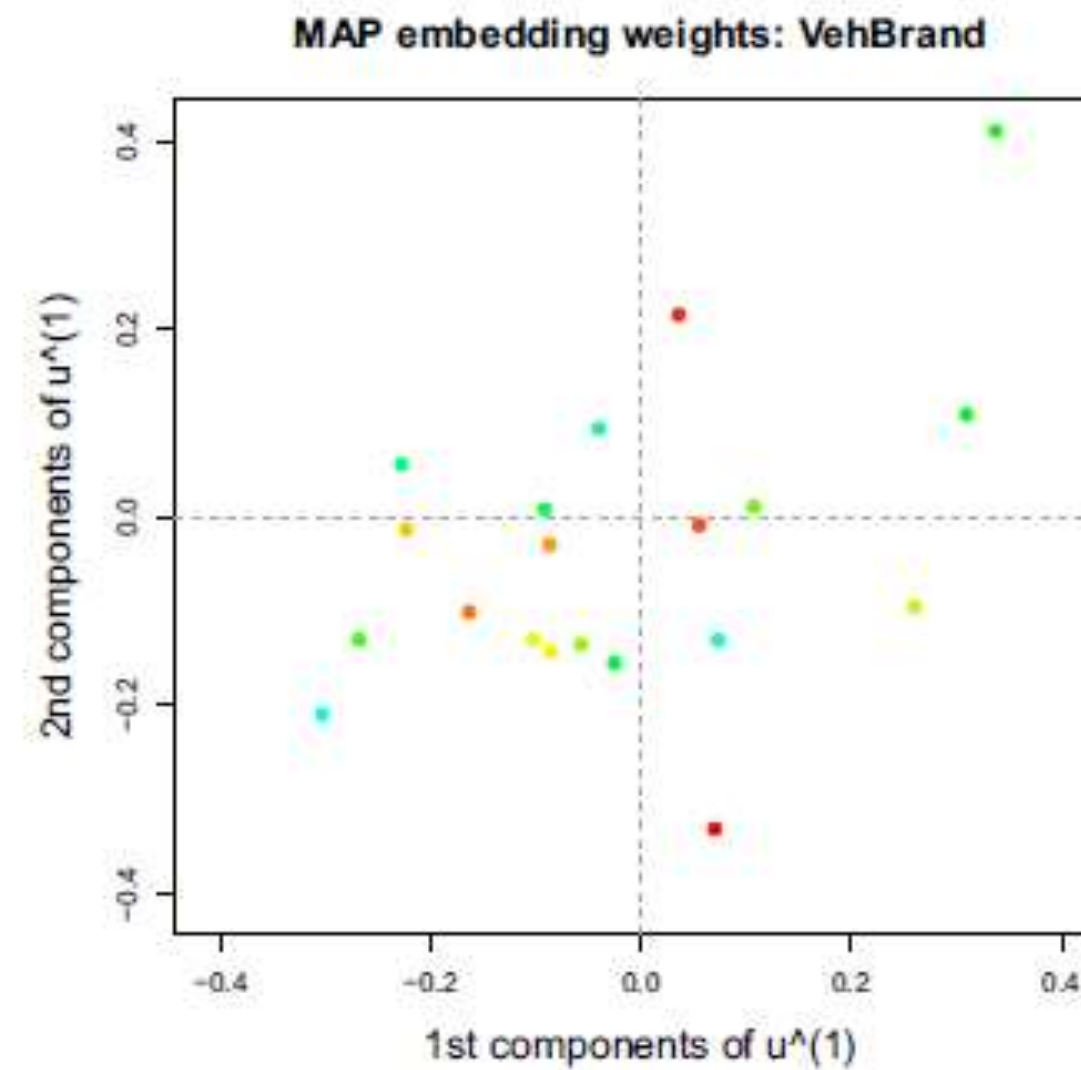


- Adding regularization to the network produces excellent results; VI slightly better than MAP

Average KL divergence		
	Network (2.30)	GLMMNet (4.3)
No regularization: with VehBrand, VehModel	0.2312	0.2958
MAP regularization: with VehBrand, VehModel	0.2212	0.2799
Ad-hoc regularization: with VehBrand, VehModel	0.2260	0.2794
VB regularization: with VehBrand, VehModel	0.2331	0.2800
No regularization: with VehBrand, VehModel, VehDetail	0.2694	0.3305
MAP regularization: with VehBrand, VehModel, VehDetail	0.1446	0.2584
Ad-hoc regularization: with VehBrand, VehModel, VehDetail	0.1449	0.2596
VB regularization: with VehBrand, VehModel, VehDetail	0.1410	0.2589

Investigating the embeddings

- We have aimed to have more regularization when exposures are low =>
- We should see smaller values of the embedding weights for lower exposure
- Exactly shown for MAP regularized embeddings, especially for VehDetail
- Size of case weights predicts overall magnitude of embeddings



Agenda

- What are embeddings?
- GLMMs
- Let's think Bayesian
- Hierarchical models based on embeddings

Hierarchical categorical data

- Above, we have treated the embeddings representing make/model/variant of the vehicle independently...
- ... BUT we know we have a hierarchical structure – cannot have a Toyota model under a Ford make!
- How can we exploit this known structure to improve our models?
- Complexity – the exposure at the lowest level of the hierarchy can be very sparse!
- A good way to think about this is that each level adds a new level of detail to the previous level =>

Can model the incremental information added in each level

Assume that each level of the hierarchy is conditionally independent of the previous levels

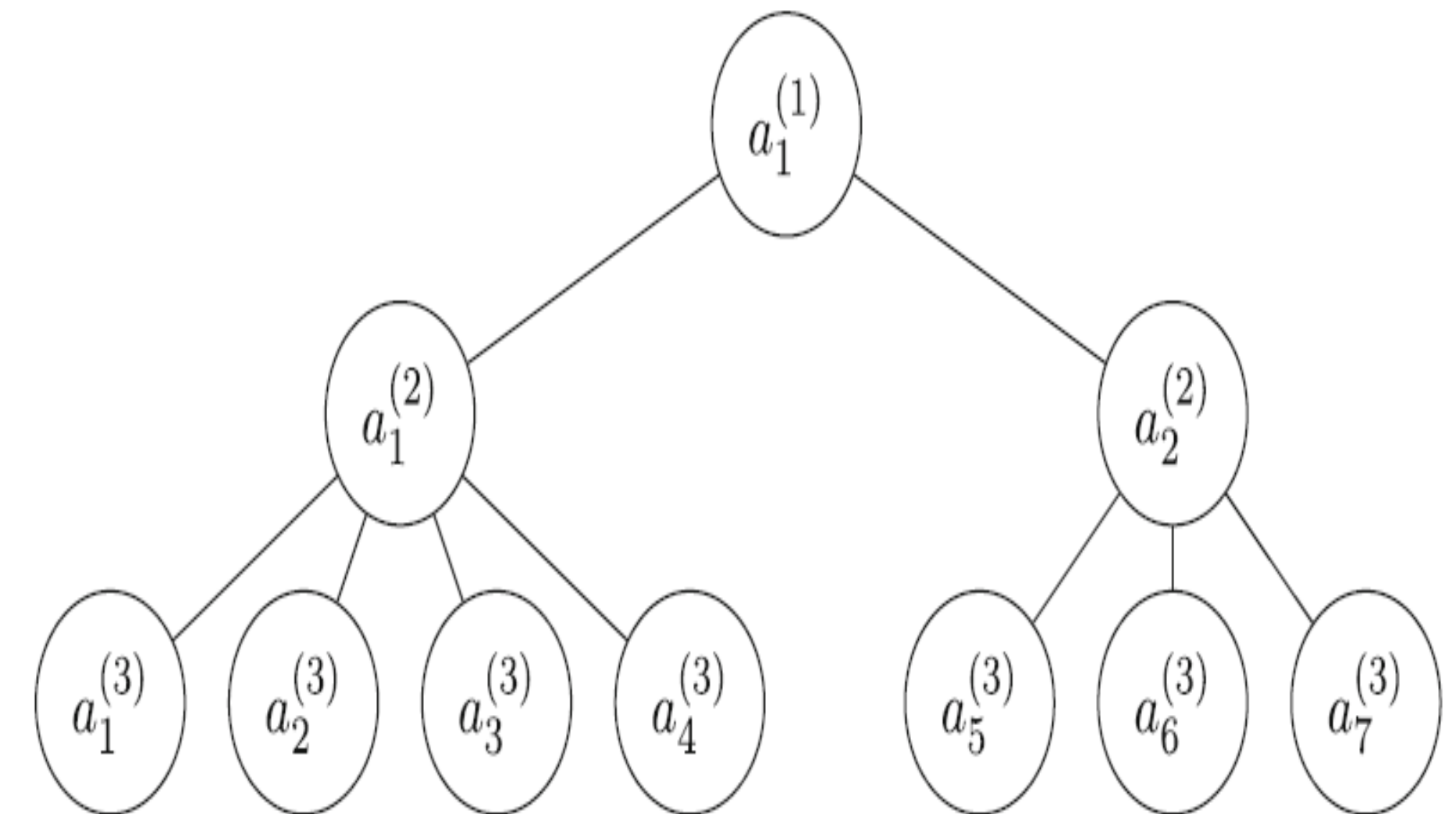


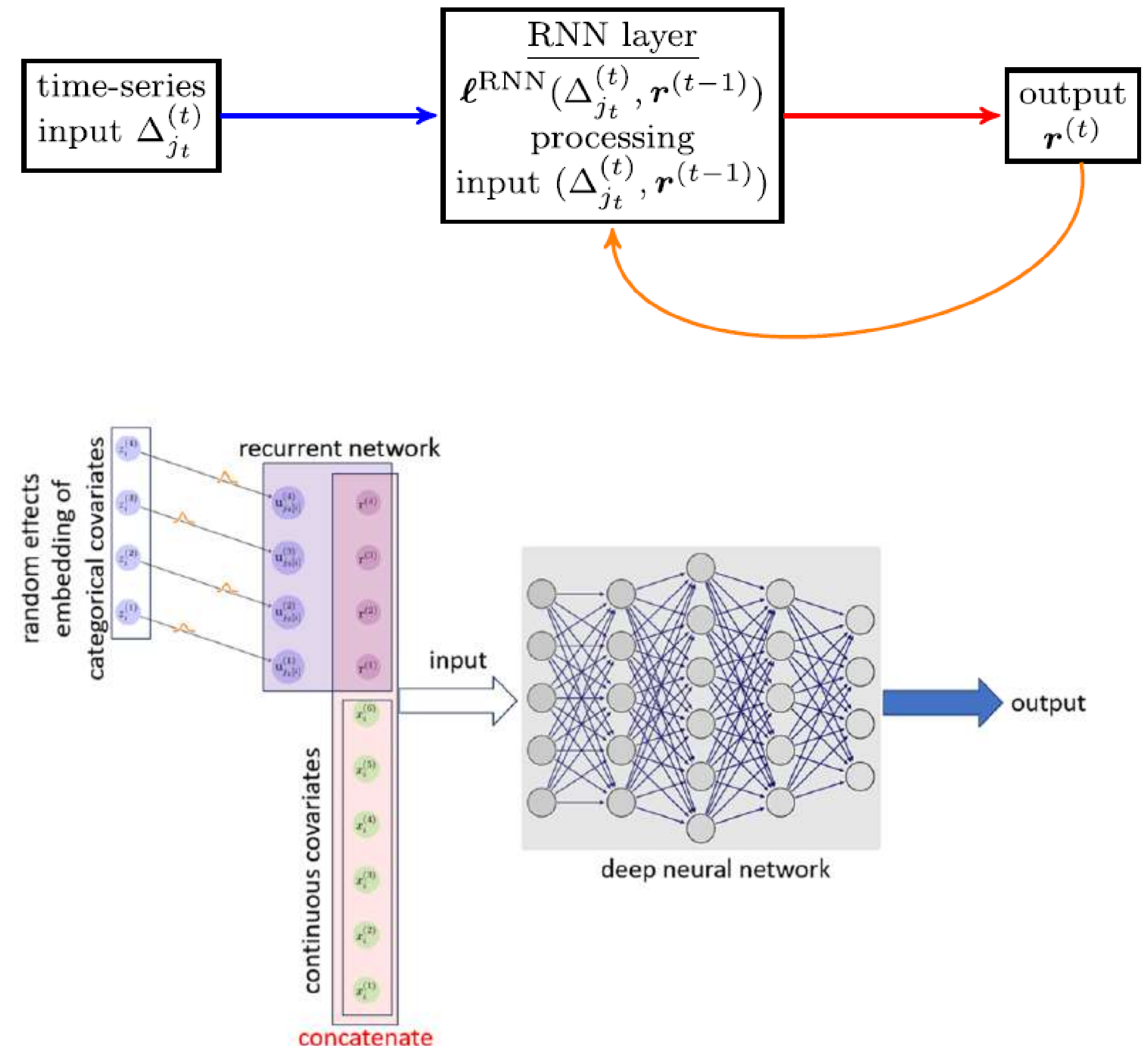
Fig.3 Descendants of $a_1^{(1)} \in A^{(1)}$ with $T = 3$ generations

$$\Delta_{j'_t}^{(t)} = \mathbf{u}_{j'_t}^{(t)} - \mathbf{u}_{j_{t-1}[j'_t]}^{(t-1)},$$

$$\Delta_j^{(t)} \Big|_{\mathbf{U}^{(t-1)}} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\tau_t^2))$$

Models for hierarchical embeddings

- We have some options for processing the hierarchical embeddings:
- Could simply put the embeddings for each level of the hierarchy into the model – leads to model that is the same as above
- Add the embeddings for each level up and input the last level of the hierarchy into the model – H1 model
Advantage: Compact representation that still captures hierarchical information
- Process each level using a neural network
 - Recurrent Neural Network – RNN model
 - Transformer Network
- Main idea for RNN/Transformer is to process the hierarchy of embeddings together so that information is shared among these via the network mechanism
Input the augmented embeddings back into the rest of the neural network

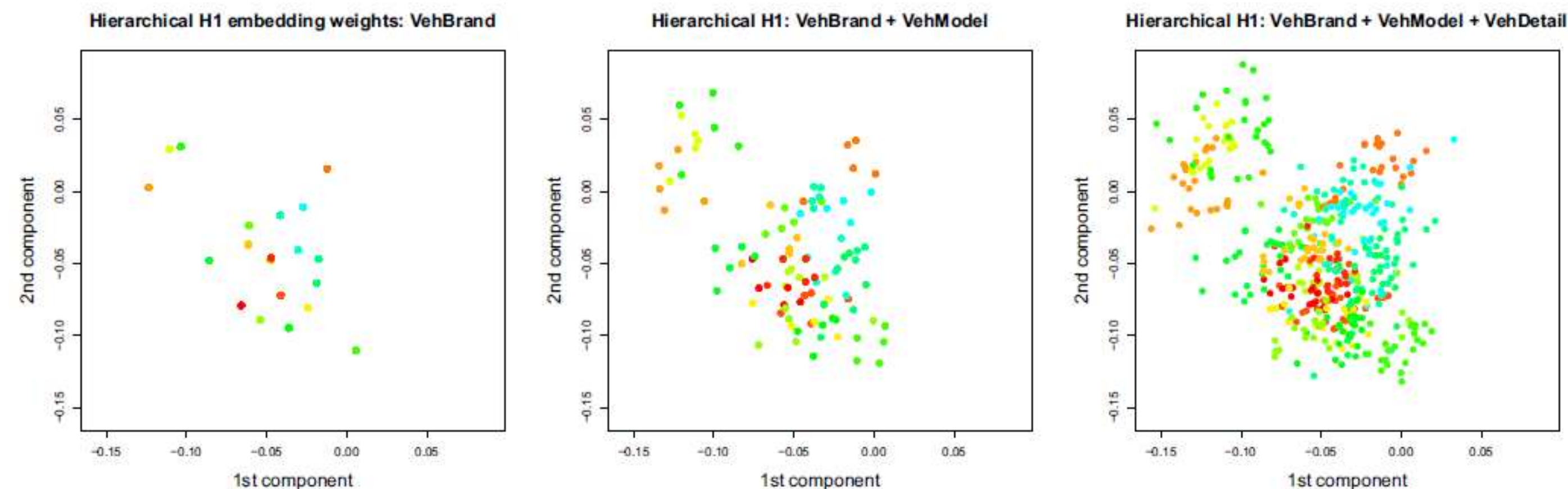


Results

- Only fit models using MAP as the previous results show VI doesn't make a huge difference
- Results show that incorporating hierarchical approaches improves on previous models!

		MAP case		
		KL div.	RMSE	MAE
(2)	Non-hierarchical: with VehBrand, VehModel	0.2212	2.4583	1.7592
(2)	Hierarchical H1: with VehBrand, VehModel	0.2144	2.4258	1.7415
(2)	RNN layer: with VehBrand, VehModel	0.2222	2.4695	1.7621
(2)	Transformer layer: with VehBrand, VehModel	0.2041	2.3759	1.7039
(3)	Non-hierarchical: with VehBrand, VehModel, VehDetail	0.1446	2.0501	1.3653
(3)	Hierarchical H1: with VehBrand, VehModel, VehDetail	0.1351	1.9941	1.3066
(3)	RNN layer: with VehBrand, VehModel, VehDetail	0.1354	2.0013	1.3020
(3)	Transformer layer: with VehBrand, VehModel, VehDetail	0.1294	1.9463	1.2802

- Evolution of embeddings in H1 model shown below



Summary

- **High-cardinality categorical variables pose significant challenges in regression modeling, particularly in insurance pricing.**
- **Entity embedding coupled with regularization techniques offers a powerful solution to handle these variables in neural network models.**
- **Two main regularization approaches were explored: MAP estimation and Variational Bayesian inference**
- **Hierarchical structures in categorical variables can be leveraged to further improve model performance.**
- **RNN and Transformer architectures show promise in processing hierarchical categorical data.**

Conclusions

- **Regularization of high-cardinality categorical embeddings significantly improves model performance compared to unregularized approaches**
- **The choice of regularization method (MAP or VB) has less impact than the act of regularizing itself**
- **Leveraging hierarchical structure in categorical variables further enhances predictive accuracy**
- **Transformer-based approach performed best overall, suggesting that complex, non-sequential relationships in hierarchical data are important**
- **Directions for Future Research:**
 - Investigate the integration of these methods with other machine learning techniques, such as gradient boosting or ensemble methods
 - Develop more sophisticated priors that incorporate domain knowledge or external data sources
 - Explore adaptive regularization techniques that adjust the strength of regularization based on data characteristics
 - Explore the potential of transfer learning using pre-trained categorical embeddings across related tasks or domains.