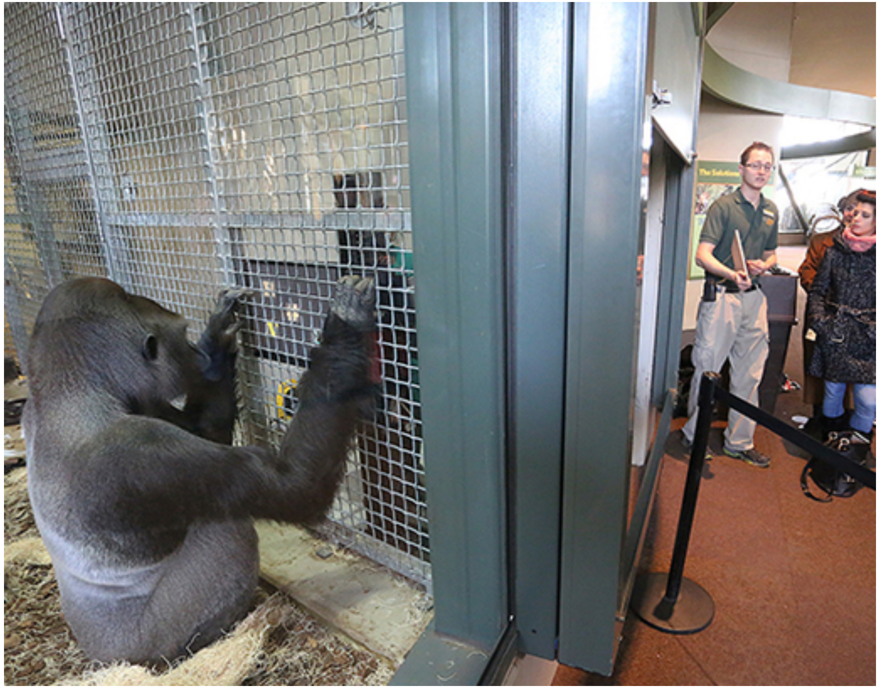
CHALMERS
UNIVERSITY OF TECHNOLOGY

Bayesian Bradley-Terry models: from primates to machine learning

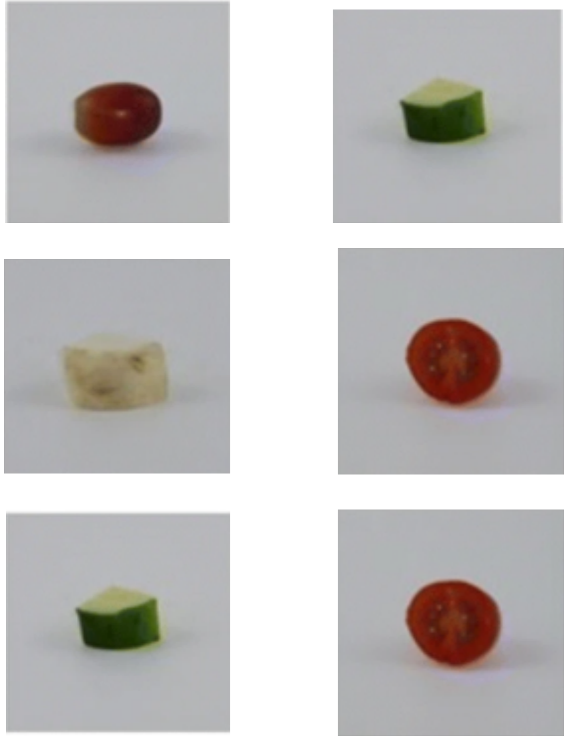
David Issa Mattos

WASP | WALLENBERG AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Gorillas with touchscreens



Lincoln zoo - Chicago



6 food images are presented in pairs

Huskisson, Sarah M., et al. "Using a touchscreen paradigm to evaluate food preferences and response to novel photographic stimuli of food in three primate species (Gorilla gorilla gorilla, Pan troglodytes, and Macaca fuscata)." *International Journal of Primatology* 41.1 (2020): 5-23.

How do we analyze this kind of data?

- We can’t do a linear/logistic model
- Frequentist non-parametric models have many limitations are hard to generalize, do not provide sufficient extensions **and are frequentist**
    - 80% of instructors in statistics in psychology made an incorrect interpretation of the p-value (H. Haller & S. Krauss 2002)
- There is a good chance that SE researchers are not good at frequentist statistics either

image1	image2	y
Grape	Cucumber	0
Grape	Turnip	0
Grape	Tomato	1
Apple	Tomato	0
Apple	Turnip	0
Carrot	Grape	0
Tomato	Turnip	0
Apple	Carrot	0
Carrot	Grape	1
Apple	Turnip	0

The Bradley-Terry model

$$\mathcal{P}[i \text{ beats } j] = \frac{\alpha_i}{\alpha_i + \alpha_j}$$

- Alpha is a ‘strength’ variable
- This model just computes the probability of selecting i instead of j

or

$$\mathcal{P}[i \text{ beats } j] = \frac{\exp \lambda_i}{\exp \lambda_i + \exp \lambda_j}$$

- We just use a log transformation so we can compute lambda from [- infinity. + infinity]

Mattos, I. D. and Ramos, E. M. S. “Bayesian Paired-Comparison with the bpcs package” In submission (arxiv:)

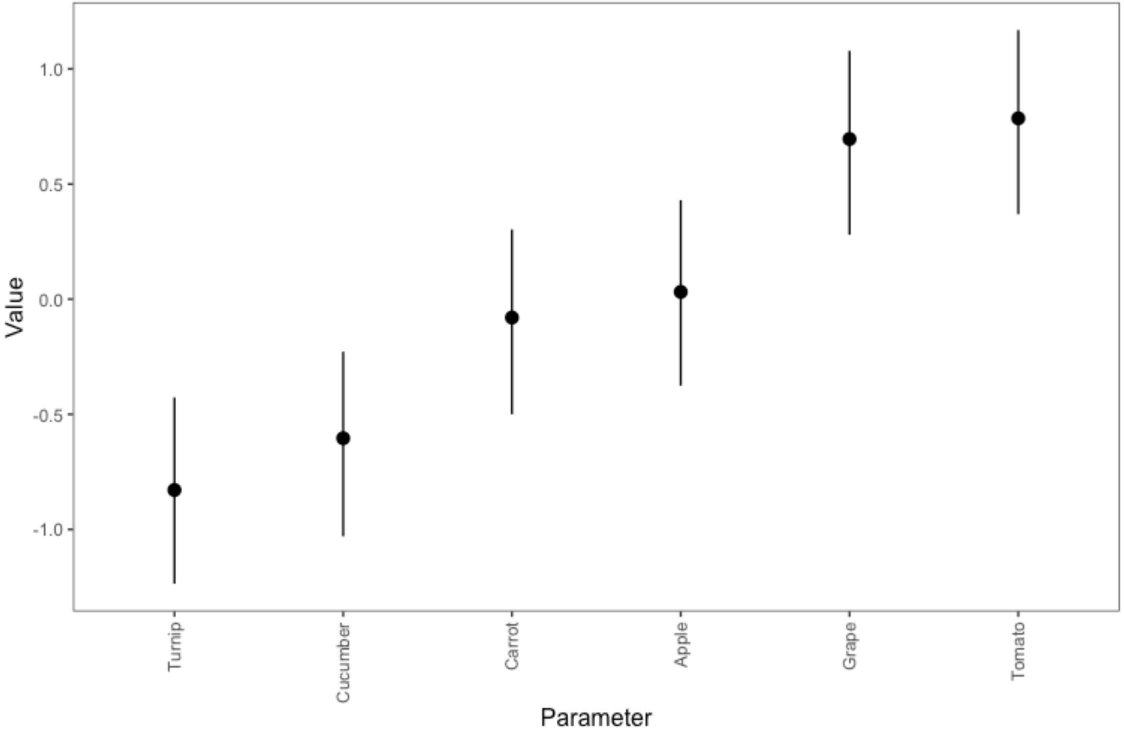
Making it Bayesian

$$\mathcal{P}[i \text{ beats } j] = \frac{\exp \lambda_i}{\exp \lambda_i + \exp \lambda_j}$$

$$y_{i,j} \sim \text{Bernoulli}(\mathcal{P}[i \text{ beats } j])$$

$$\lambda_i \sim \mathcal{N}(0, \sigma_\lambda^2), \text{ [Prior]}$$

Parameter estimates



What does it have to do with SE?

Survey design

- Likert scale is not always good (actually it has loads of problems)
  - Bias by anchoring (you answer the next question based on the previous one)
  - Faking response for social acceptability
  - Not suitable for some people/contexts
  - Using statistics wrong
  - See more: Liddell & Kruschke, 2018, Coetzee & Taylor, 1996; Luckett, Burns, & Jenkinson, 2020; Petrou, 2003
- Forced choice assessment
  - You must choose one alternative only (like the gorillas)
  - It can be out of two (more common) or out of several alternatives

What does it have to do with SE?

Benchmark experiments

- You want to compare tools/methods tested against a benchmark suite
- Many repetitions of the same tool in the same benchmark test
  - Introduces dependency in the data
- Some benchmarks are not comparable in term of response
  - Use of a ranking system or data transformation
- There can be clusters in the benchmarks
  - Introduces dependency in the data

Benchmark experiments

- If one tool (i) performs better than the other (j) in a benchmark test, we can say that:
  - Tools i beats tool j in benchmark s
- Extending the Bradley-Terry model for random effects
  - We want to compensate for dependency in the data
  - This is not really optional

$$\mathcal{P}[i \text{ beats } j] = \frac{\exp \lambda_{i,s}}{\exp \lambda_{i,s} + \exp \lambda_{j,s}}$$

$$\lambda_{i,s} = \lambda_i + U_{i,s}$$

$$y_{i,j} \sim \text{Bernoulli}(\mathcal{P}[i \text{ beats } j])$$

$$\lambda_i \sim \mathcal{N}(0, \sigma_\lambda^2), \text{ [Prior]}$$

$$U_{i,s} \sim \mathcal{N}(0, U_{\text{std}}^2), \text{ [Prior]}$$

$$U_{\text{std}} \sim \mathcal{N}(0, \sigma_U^2), \text{ [Prior]}$$

Classification of algorithms for automated labelling

Teodor's research

- There are different semi-supervised learning algorithms (ask Teodor)
  - Label Spreading KNN and RBF
  - Label Propagation KNN and RBF
- Experimental setup with 18 datasets in three categories and 10 repetitions of each algorithm in each dataset (with different seeds)
- How do we rank them in terms of accuracy?

Fredriksson T. Mattos D. I., Bosch J., Olsson. H. H. "An Empirical Evaluation of Algorithms for Data Labeling" In submission 2021

- First, we expand the data into paired comparisons

Value	Iteration	Variable	Model	Dataset	DataType
0.6967609	7	50%	LabelSpreading_rbf	digits	image
0.3333333	2	50%	LabelSpreading_knn	corpus	text
0.1666667	3	50%	LabelPropagation_knn	reuters	text
0.0181818	8	10%	LabelPropagation_rbf	cifar	image
0.0004808	8	50%	LabelPropagation_rbf	cifar2	image
0.2756910	4	50%	LabelPropagation_knn	german	numeric
0.0181818	4	10%	LabelPropagation_knn	mnist	image
0.1666667	9	10%	LabelPropagation_knn	reuters	text
0.1280959	1	10%	LabelPropagation_rbf	20news	text
0.2635986	0	50%	LabelPropagation_knn	musk1	numeric

model0	model1	y	Iteration	Dataset	DatasetType
LabelPropagation_rbf	LabelSpreading_rbf	1	0	wine	numeric
LabelSpreading_knn	LabelSpreading_rbf	0	8	wine	numeric
LabelPropagation_rbf	LabelSpreading_knn	0	5	corpus	text
LabelPropagation_rbf	LabelSpreading_rbf	0	6	wine	numeric
LabelPropagation_knn	LabelPropagation_rbf	1	3	cifar	image
LabelPropagation_rbf	LabelSpreading_rbf	0	0	corpus	text
LabelPropagation_knn	LabelSpreading_rbf	0	7	20news	text
LabelPropagation_knn	LabelSpreading_rbf	0	8	reuters	text
LabelPropagation_rbf	LabelSpreading_knn	0	5	musk1	numeric
LabelSpreading_knn	LabelSpreading_rbf	1	9	cifar	image
LabelPropagation_knn	LabelSpreading_rbf	0	1	20news	text

- Then we can fit the model

Estimated posterior ranks

Parameter	MedianRank	MeanRank	StdRank
lambda[LabelPropagation_rbf]	1	1.426	0.756
lambda[LabelPropagation_knn]	3	2.700	0.976
lambda[LabelSpreading_knn]	3	2.812	0.959
lambda[LabelSpreading_rbf]	3	3.062	0.976

Estimated posterior probabilites

i	j	i_beats_j	j_beats_i
LabelPropagation_knn	LabelPropagation_rbf	0.33	0.67
LabelPropagation_knn	LabelSpreading_knn	0.45	0.55
LabelPropagation_knn	LabelSpreading_rbf	0.45	0.55
LabelPropagation_rbf	LabelSpreading_knn	0.62	0.38
LabelPropagation_rbf	LabelSpreading_rbf	0.56	0.44
LabelSpreading_knn	LabelSpreading_rbf	0.41	0.59

What about ties?

Surveys

- If you have a survey with questions such as:
  - Which do you prefer? A, B or doesn’t matter
- You need to take ties (doesn’t matter) into consideration

Benchmark experiments

- If your benchmarks have ties:
  - Solve then randomly (if very few ties)
  - You model the ties (many ties)

The Davidson model

- Extension of the Bradley-Terry model to handle ties
- Extra parameter nu controls how much ties you have, if they depend on the strengt values or if the data have ties independentl of the strength

$$\mathcal{P}[i \text{ beats } j | \text{not tie}] = \frac{\exp \lambda_i}{\exp \lambda_i + \exp \lambda_j + \exp\left(\nu + \frac{\lambda_i + \lambda_j}{2}\right)}$$

$$\mathcal{P}[i \text{ ties } j] = \frac{\exp\left(\nu + \frac{\lambda_i + \lambda_j}{2}\right)}{\exp \lambda_i + \exp \lambda_j + \exp\left(\nu + \frac{\lambda_i + \lambda_j}{2}\right)}$$

$$y_i \sim \text{Bernoulli}(\mathcal{P}[i \text{ beats } j | \text{not tie}])$$

$$\text{tie}_{i,j} \sim \text{Bernoulli}(\mathcal{P}[i \text{ ties } j])$$

$$\lambda_i \sim \mathcal{N}(0, \sigma_\lambda^2), \ [\text{Prior}]$$

$$\nu \sim \mathcal{N}(0, \sigma_\nu^2), \ [\text{Prior}]$$

Modeling the Brazilian football league

- Similar to the other models, we can use the Davidson model to model the ties and obtain a posterior rank estimation

It indicates that ties occur a bit more often than by just the strength of the teams

Parameters estimates

Parameter	Mean	HPD_lower	HPD_higher
lambda[Avaí]	-1.912	-3.331	-0.627
lambda[Internacional]	0.162	-1.031	1.440
lambda[Cruzeiro]	-0.683	-1.921	0.520
lambda[Botafogo-rj]	-0.584	-1.862	0.637
lambda[Vasco]	-0.067	-1.286	1.137
lambda[Corinthians]	0.310	-0.867	1.576
lambda[CSA]	-1.101	-2.368	0.143
lambda[Goiás]	-0.061	-1.335	1.122
lambda[Fortaleza]	0.133	-1.107	1.352
lambda[Santos]	1.070	-0.217	2.291
lambda[Grêmio]	0.631	-0.573	1.912
lambda[Palmeiras]	1.245	-0.011	2.514
lambda[Atlético-MG]	-0.048	-1.340	1.133
lambda[Chapecoense]	-0.986	-2.280	0.272
lambda[Ceará]	-0.729	-2.022	0.433
lambda[Athlético-PR]	0.717	-0.580	1.919
lambda[Flamengo]	2.042	0.696	3.372
lambda[São Paulo]	0.616	-0.654	1.849
lambda[Bahia]	-0.129	-1.353	1.098
lambda[Fluminense]	-0.378	-1.594	0.903
nu	0.385	0.159	0.624

Note that it does not necessarily indicates who won the league since there are other rules involved (such as win=3pts and tie=1pt) there are not considered in the model

There are other possible extensions such as order effects (home field advantage), i.e. the order in which you present the items matter

Estimated posterior ranks

Parameter	MedianRank	MeanRank	StdRank
lambda[Flamengo]	1	1.314	0.758
lambda[Palmeiras]	3	3.127	1.811
lambda[Santos]	3	3.778	2.082
lambda[Athlético-PR]	5	5.481	2.659
lambda[Grêmio]	6	5.943	2.751
lambda[São Paulo]	6	6.062	2.820
lambda[Corinthians]	8	8.218	3.214
lambda[Fortaleza]	9	9.560	3.523
lambda[Internacional]	9	9.175	3.285
lambda[Atlético-MG]	11	10.718	3.482
lambda[Goiás]	11	10.932	3.312
lambda[Vasco]	11	10.962	3.334
lambda[Bahia]	12	11.406	3.427
lambda[Fluminense]	14	13.467	3.251
lambda[Botafogo-rj]	15	14.792	2.990
lambda[Ceará]	16	15.629	2.723
lambda[Cruzeiro]	16	15.392	2.786
lambda[Chapecoense]	18	16.953	2.412
lambda[CSA]	18	17.479	2.028
lambda[Avaí]	20	19.612	0.860

How to do this (very cool) type of analysis?

- Download the bpcs package (https://github.com/davidissamattos/bpcs) or in CRAN
- Simple interface
- Inspect the posterior distribution and measure effect size (the actual probabilities)
- Publication-ready functions (export tables to Latex and HTML) and generate figures

```
m1_autol <-
  bpc(
    d_acc_bt,
    player0 = 'model0',
    player1 = 'model1',
    result_column = 'y',
    model_type = 'bt-U',
    cluster = c("Dataset"),
    priors = list(prior_lambda_std = 0.5, prior_U1_std=1.0),
    iter = 3000
  )
```

```
get_probabilities_table(m1_gor, format='html')
```

```
get_parameters_table(m_football, format = 'html')
```

```
plot(m1_gor, rotate_x_labels = T)
```

Reproducible code and this presentation

- https://davidissamattos.github.io/SESeminarBT2021/
- https://github.com/davidissamattos/SESeminarBT2021

CHALMERS
UNIVERSITY OF TECHNOLOGY

Thanks!

davidis@chalmers.se

WASP | WALLENBERG AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM