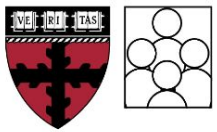
Applications of latent variable models for data exploration and uncertainty quantification

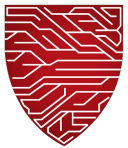
March 15, 2019

Melanie F. Pradier



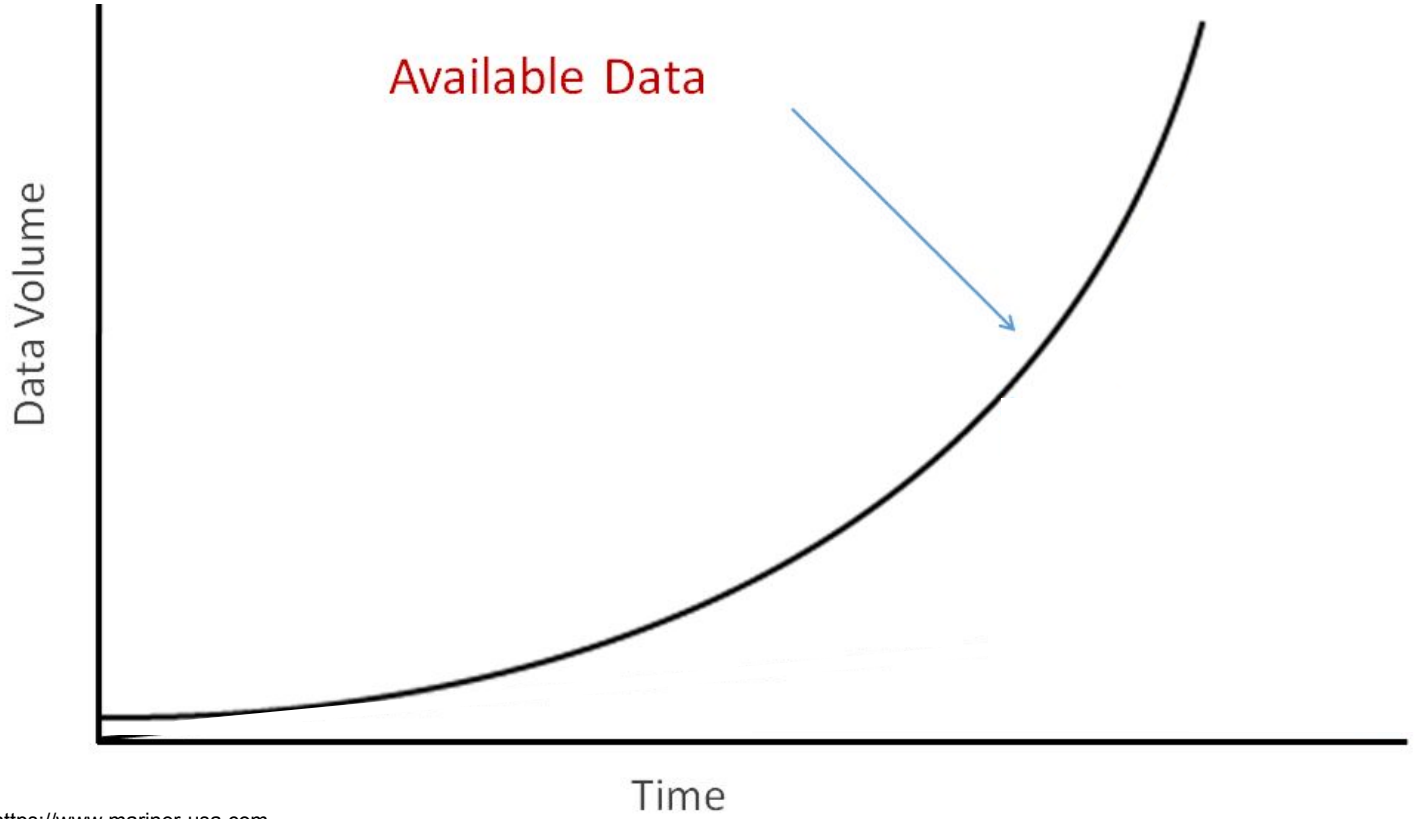
CRCS Center for Research on Computation and Society

at Harvard John A. Paulson School of Engineering and Applied Sciences



HDSI | Harvard Data Science Initiative

Data everywhere!



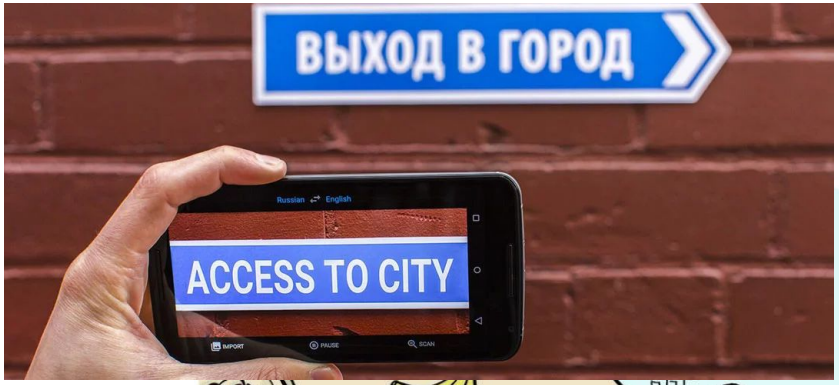
Adapted from https://www.mariner-usa.com

Huge amount of opportunities...

Huge amount of opportunities...

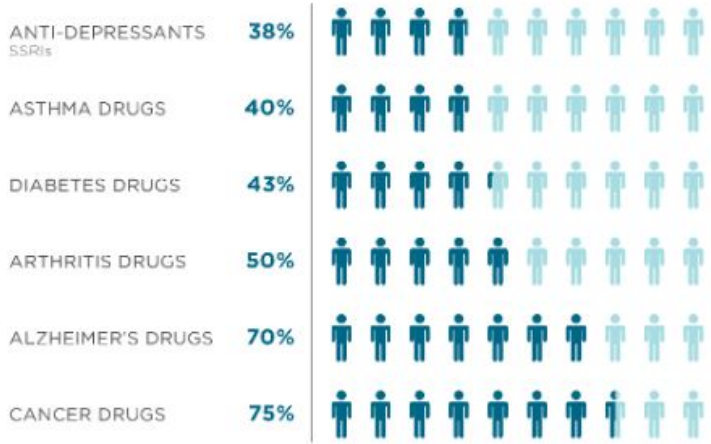
Huge amount of opportunities...

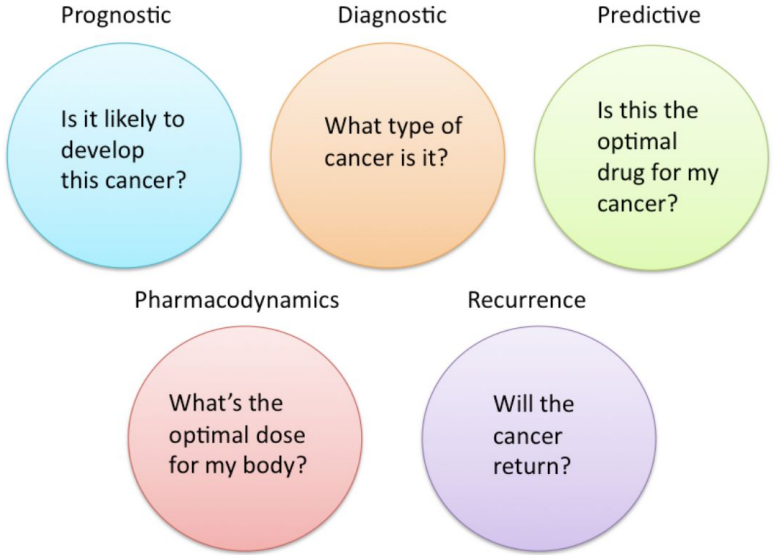
Huge amount of opportunities...



Huge amount of opportunities...

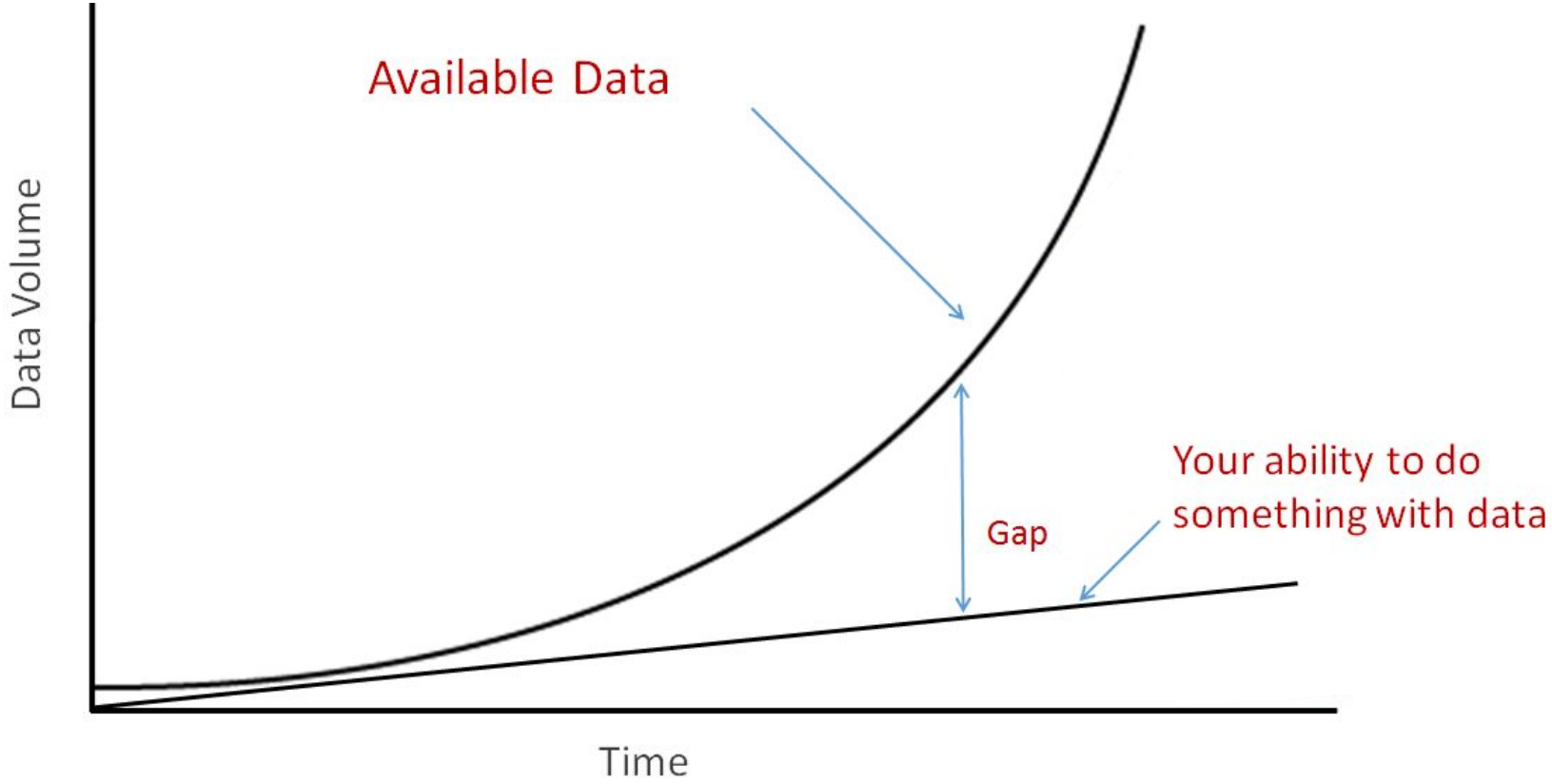
...but still many challenges



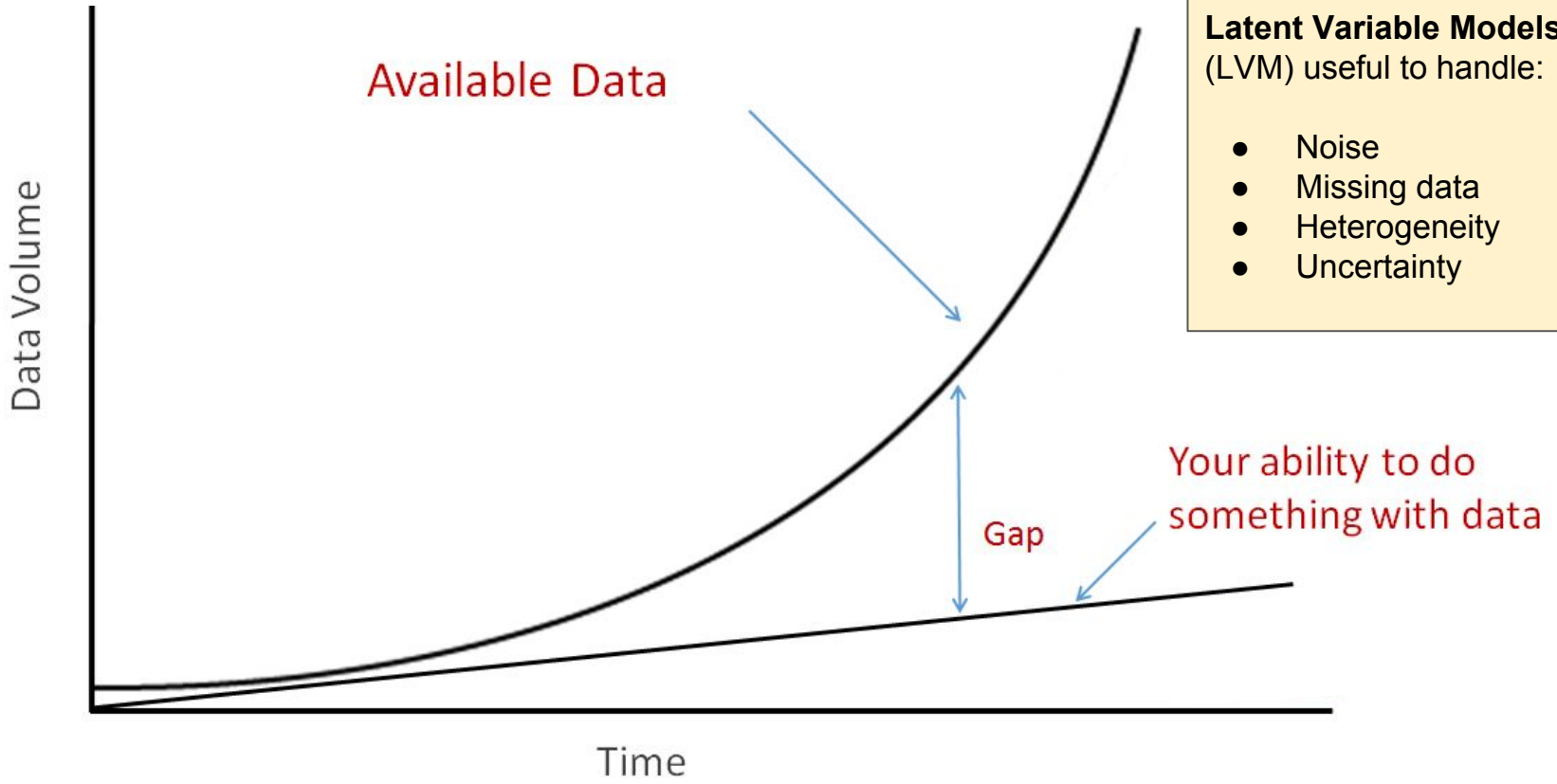


Specially in high-stake decision scenarios

Bridging the gap



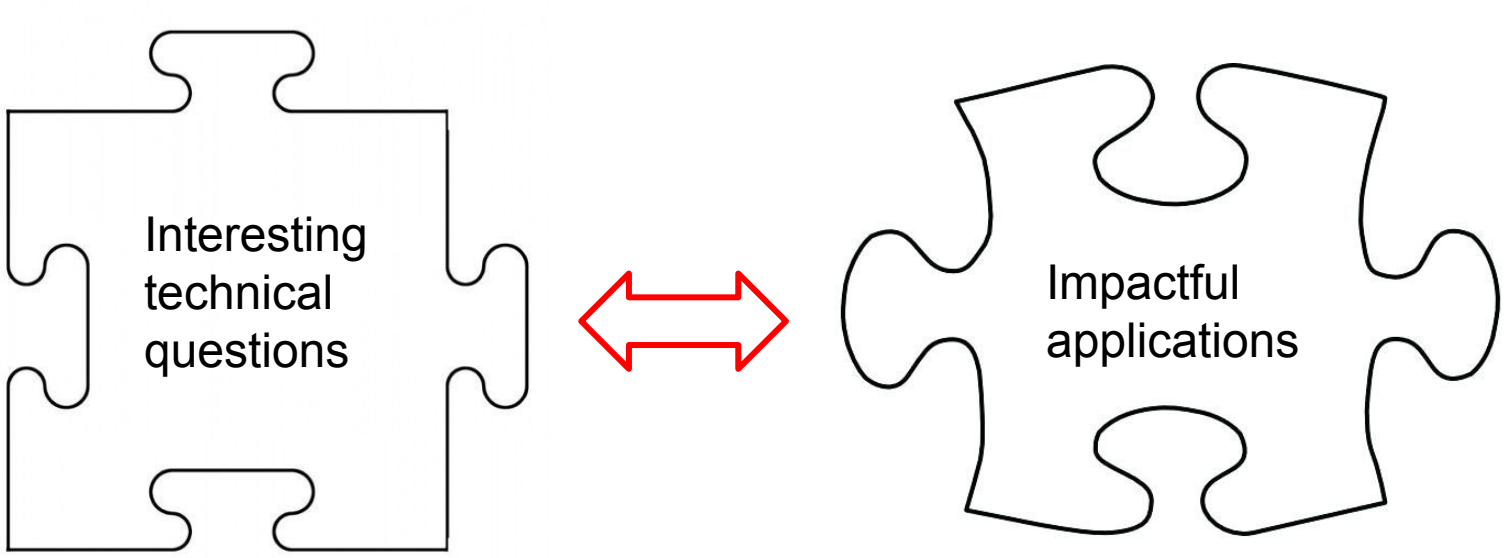
Bridging the gap



Latent Variable Models (LVM) useful to handle:

- Noise
- Missing data
- Heterogeneity
- Uncertainty

My research: probabilistic models for societal needs



- Design probabilistic models (modeling/inference) for real-world applications
- Crucial: multidisciplinary collaboration

Agenda from now on...

Applications of Latent Variable Models (LVMs) for:

1. Data Exploration
    - Biomarker discovery in clinical trials
2. Uncertainty Quantification
    - Inference framework for Bayesian neural networks

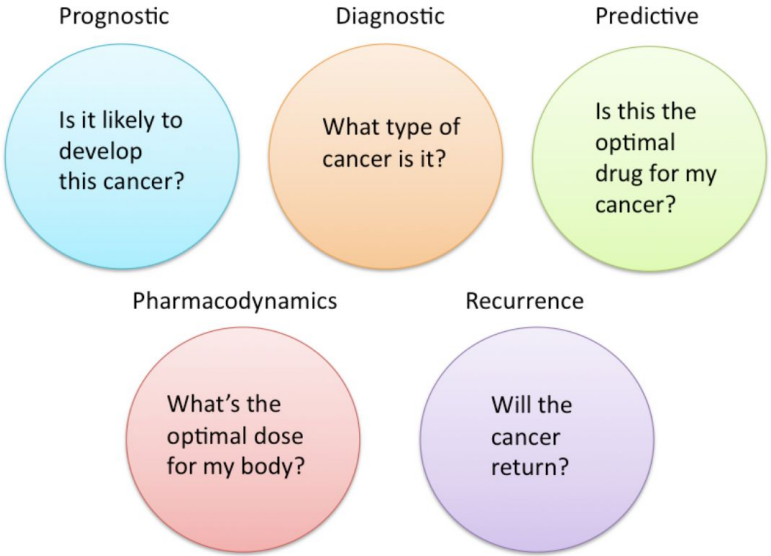
Goal 1: Data exploration

Biomarker = ”any variable that can be used as an indicator of a particular disease state”

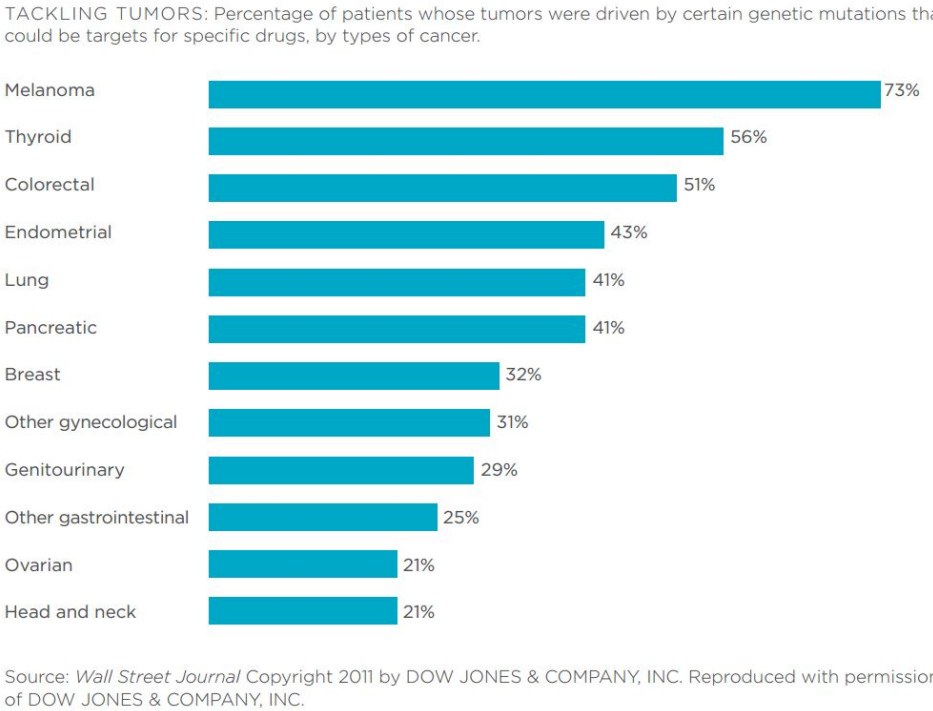
Objective: Biomarker discovery

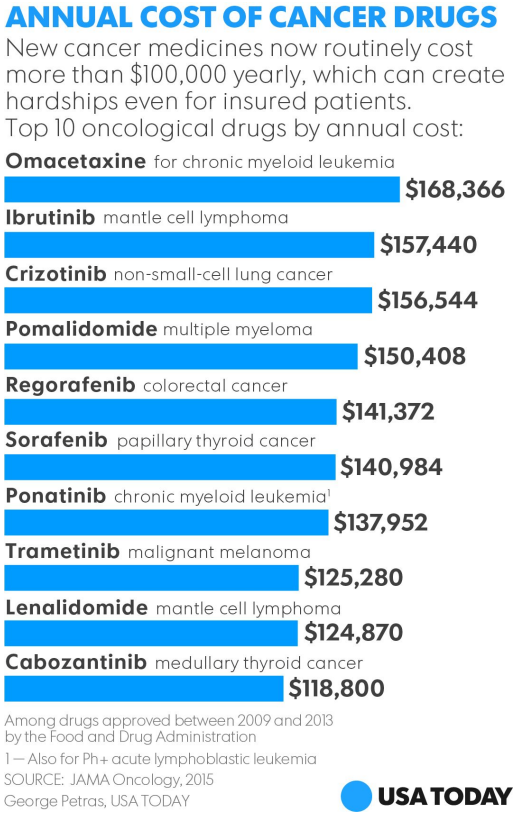
Biomarkers used everywhere, e.g.,

- Prostate-specific antigen (PSA) to diagnose prostate cancer
- Estrogen / progesterone to predict sensitivity to endocrine therapy in breast cancer
- KRAS mutation to predict resistance to EGFr antibody treatment



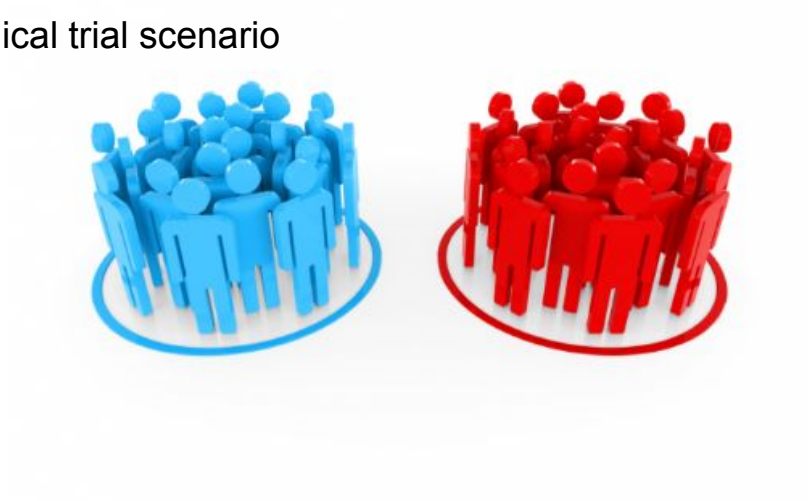
Biomarker discovery is expensive

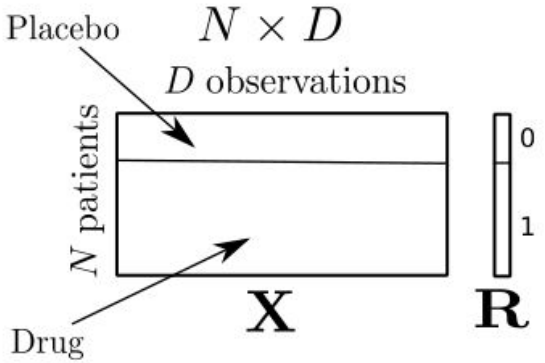




Problem formulation

Clinical trial scenario



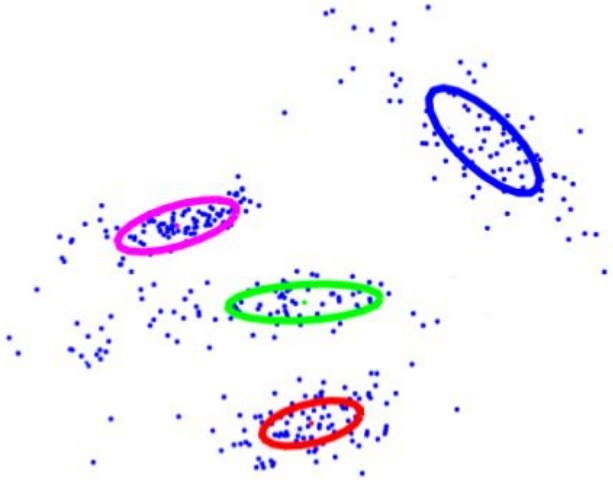


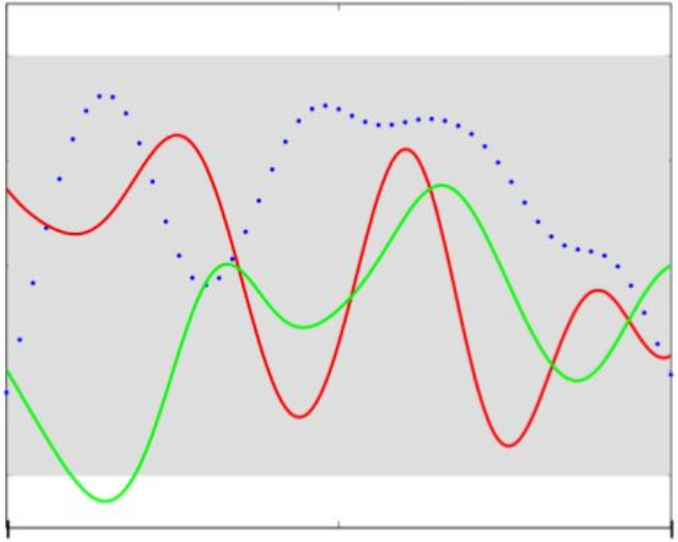
We want to discover:

1. Indicators of disease progression: prognostic biomarkers
2. Indicators of (positive) drug response: predictive biomarkers

Bayesian nonparametrics

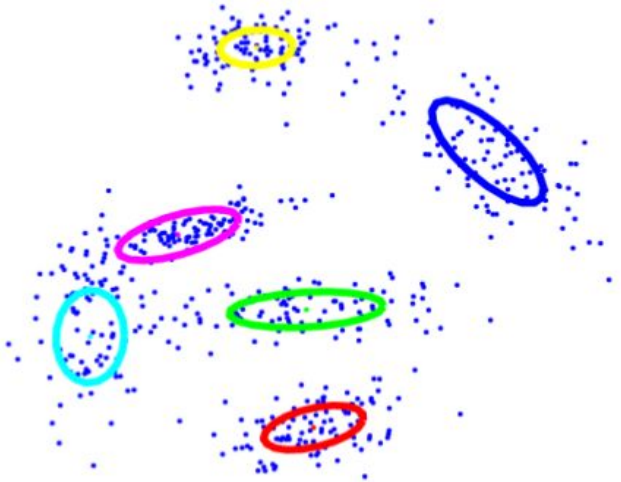
- Bayesian: to handle uncertainty $p(\mathbf{\Phi}|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{\Phi})p(\mathbf{\Phi})$
- Nonparametric: to adapt model complexity depending on input data (hypothesis generation)

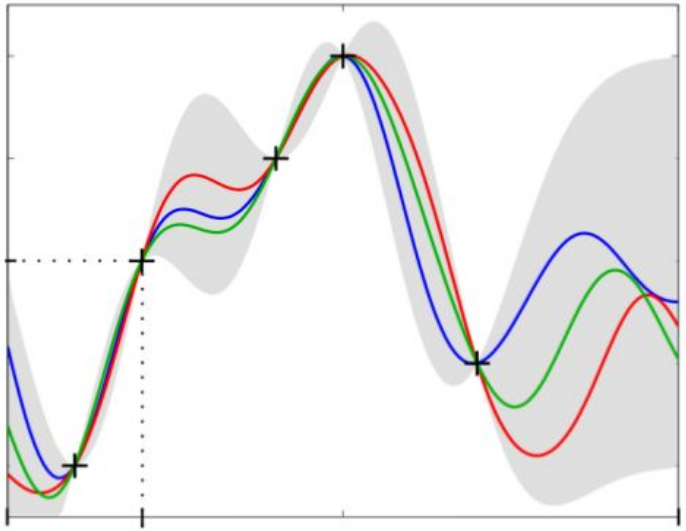




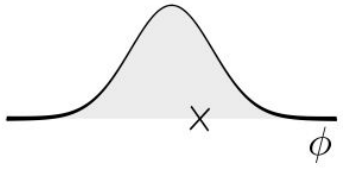
Bayesian nonparametrics

- Bayesian: to handle uncertainty $p(\mathbf{\Phi}|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{\Phi})p(\mathbf{\Phi})$
- Nonparametric: to adapt model complexity depending on input data (hypothesis generation)

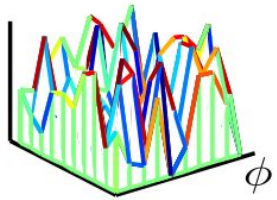




Indian Buffet Process (Ghahramani et.al, 2006)



$$\xrightarrow[x \sim \mathcal{N}(\mu_0, \sigma_0^2)]{} \quad x = 0.356$$

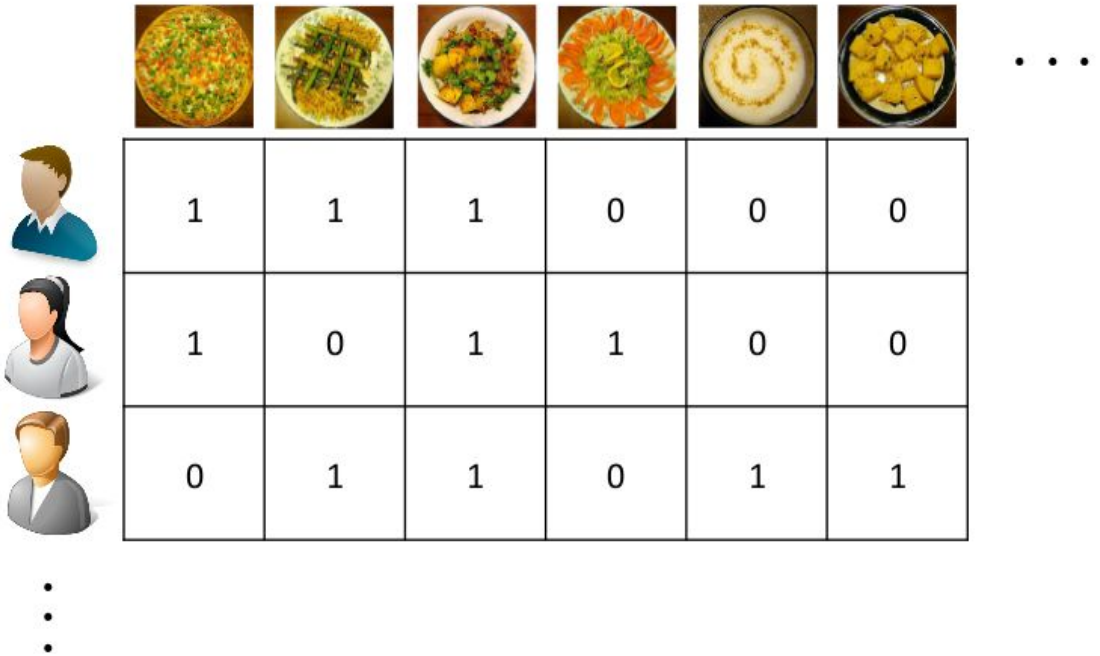


$$\xrightarrow[Z \sim \mathrm{IBP}(\alpha)]{} \quad Z = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \vdots$$

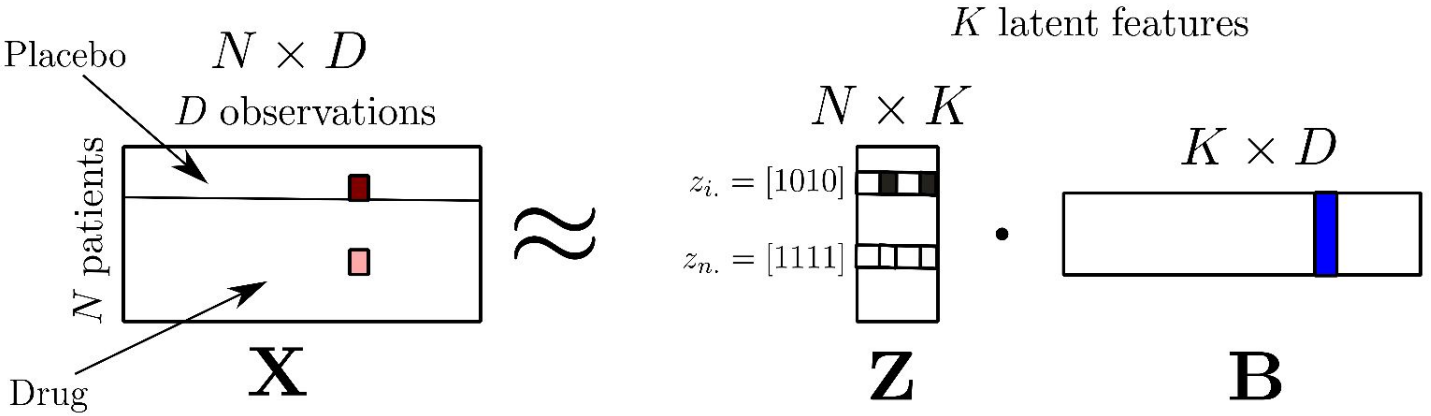
- Prior over binary matrices with infinite number of columns
- Rows $\equiv$ observations; columns $\equiv$ features
- $\mathbf{Z} \sim \mathrm{IBP}(\alpha)$
- α: concentration parameter

Indian Buffet Process (Ghahramani et.al, 2006)

Indian Buffet Process (Ghahramani et.al, 2006)



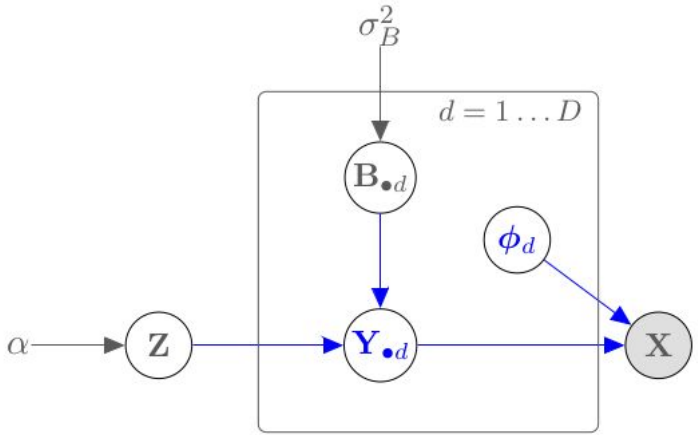
Infinite latent feature model (intuition)



- $x_{id} = 173\,\mathrm{ml/dL} = 73 + 0 + 100\,\mathrm{ml/dL}$
- $x_{nd} = 136\,\mathrm{ml/dL} = 86 + 40 + 60 - 50\,\mathrm{ml/dL}$

General Latent Feature Model (GLFM)

Latent feature model for heterogeneous datasets



- Link functions T_d depend on type of data for each dimension d

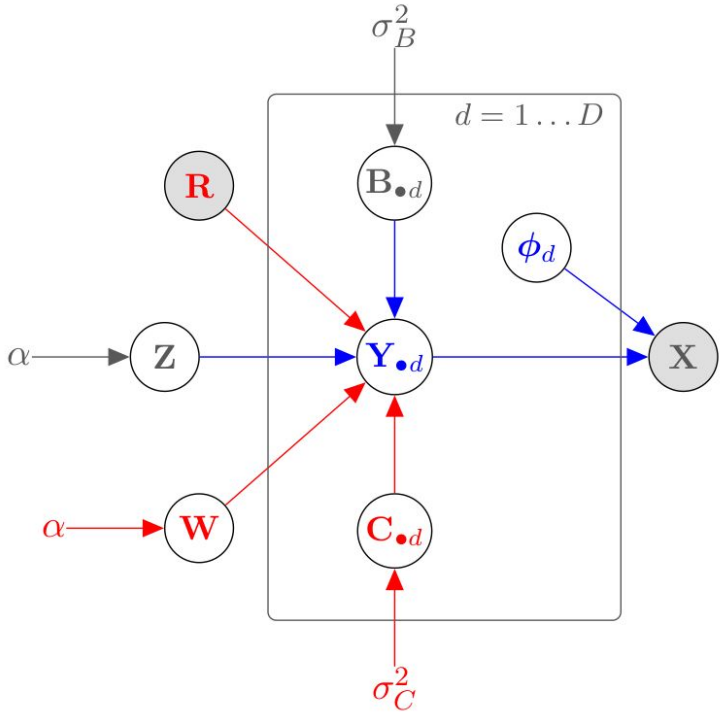
$$
\begin{aligned}
x_{nd} &= T_d(y_{nd}; \boldsymbol{\phi_d}) \\
y_{nd}|\mathbf{Z}, \mathbf{B} &\sim \mathcal{N}(\mathbf{Z}_{n\bullet}\mathbf{B}_{\bullet d}, \sigma_y^2) \\
B_{kd} &\sim \mathcal{N}(0, \sigma_B^2) \\
\mathbf{Z} &\sim \text{IBP}(\alpha)
\end{aligned}
$$

Open-source python code

https://github.com/ivaleraM/GLFM

I. Valera, M. F. Pradier, M. Lomeli, and Z. Ghahramani. **General Latent Feature Models for Heterogeneous Datasets**. *In submission to Journal of Machine Learning Research*. 2018.

Case-Control Indian Buffet Process (C-IBP)



R_n: drug indicator por patient n

$$x_{nd} = T_d(y_{nd}; \boldsymbol{\phi_d})$$
$$y_{nd}|\mathbf{Z}, \mathbf{W}, \mathbf{B}, \mathbf{C}, \mathbf{R} \sim$$
$$\mathcal{N}(\mathbf{Z}_{n\bullet}\mathbf{B}_{\bullet d} + \mathbb{1}[\mathbf{R}_n = 1]\mathbf{W}_{n\bullet}\mathbf{C}_{\bullet d}, \sigma_y^2)$$
$$B_{kd} \sim \mathcal{N}(0, \sigma_B^2)$$
$$\mathbf{Z} \sim \text{IBP}(\alpha)$$
$$C_{kd} \sim \mathcal{N}(0, \sigma_C^2)$$
$$\mathbf{W} \sim \text{IBP}(\alpha)$$

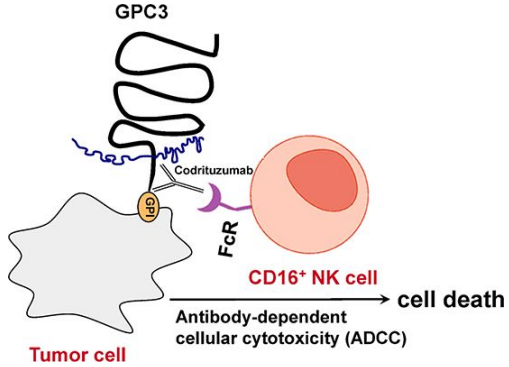
- **Inference**: MCMC approach with accelerated Gibbs sampling
- **Biomarker discovery**: statistical multiple hypothesis testing

M. F. Pradier, B. Reis, L. Jukofsky, F. Milletti, T. Ohtomo, F. Perez-Cruz, and O. Puig. **Case-control Indian Buffet Process identifies biomarkers of response to Codrituzumab**. *Accepted to BMC Cancer*. 2019.

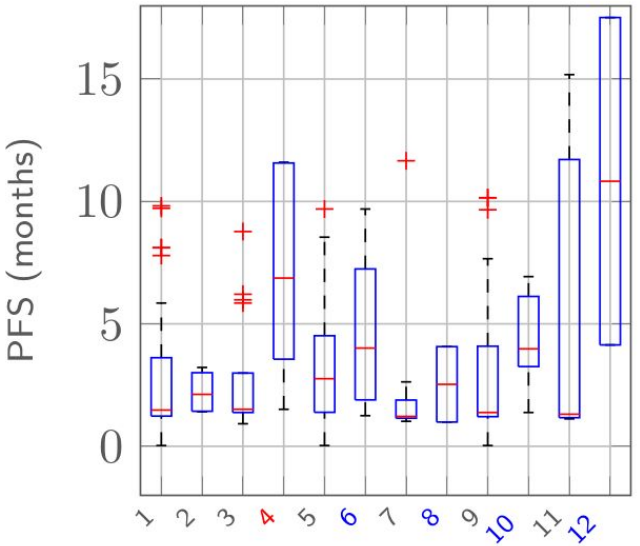
Results: subpopulations

GPC3 Antibody Treatment against Liver Cancer (J. Hepatology. 2016 Apr, Abou-Alfa et.al.)

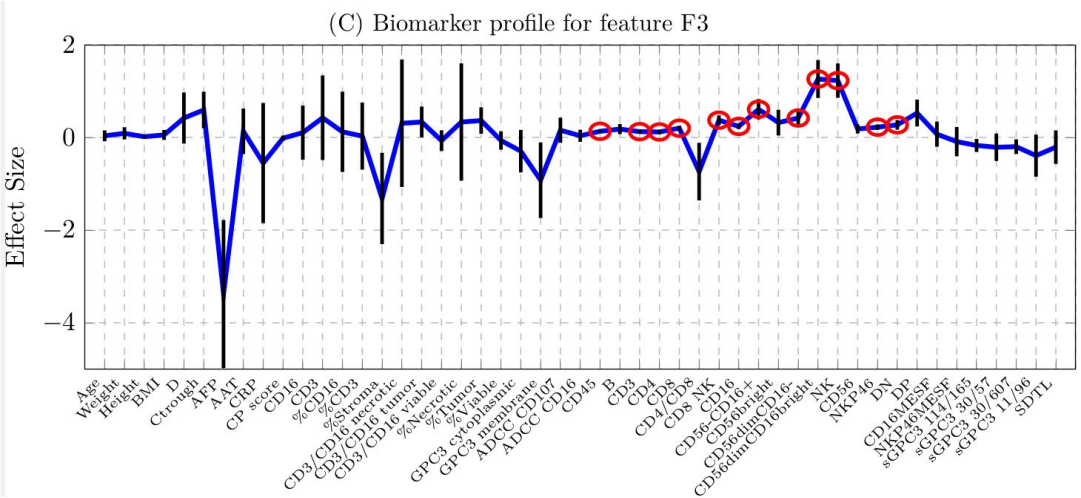
- 180 patients: 60 took a placebo, 120 took the drug
- PFS: Progression Free Survival

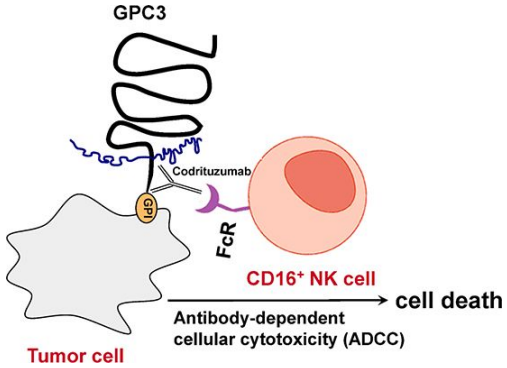


Sub-population	Drug Identifier	F1	F2	F3	Size (number of patients)	Mean PFS (months)	Median PFS (months)
1.	0	0	0	0	33.37	3.06	1.65
2.	0	0	1	0	4.07	2.29	2.24
3.	0	1	0	0	17.84	2.72	1.81
4.	0	1	1	0	4.72	7.05	7.18
5.	1	0	0	0	51.52	3.22	2.55
6.	1	0	0	1	16.77	4.17	3.65
7.	1	0	1	0	8.38	1.74	1.33
8.	1	0	1	1	2.07	2.69	2.65
9.	1	1	0	0	29.88	3.36	2.03
10.	1	1	0	1	4.90	4.44	4.34
11.	1	1	1	0	4.53	6.31	5.31
12.	1	1	1	1	1.94	10.04	10.01



Results: biomarker profiles





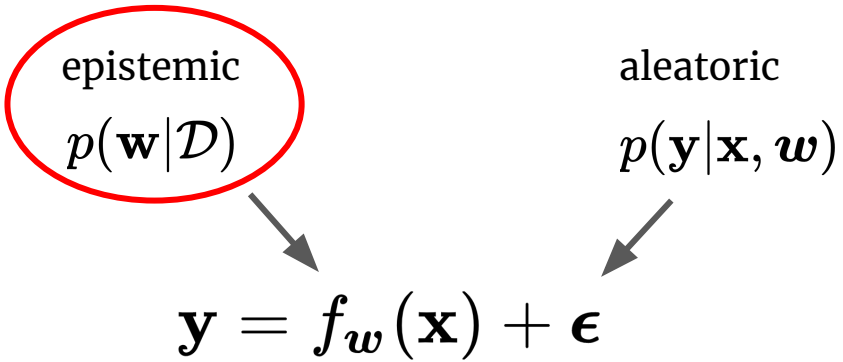
Take-away message:

- LVMs useful to identify hidden patterns underlying data
- Challenge addressed: data heterogeneity (both across dimensions and observations)

M. F. Pradier, B. Reis, L. Jukofsky, F. Milletti, T. Ohtomo, F. Perez-Cruz, and O. Puig. **Case-control Indian Buffet Process identifies biomarkers of response to Codrituzumab**. *Accepted to BMC Cancer*. 2019.

Goal 2: Uncertainty quantification

Two sources of uncertainty



[Depeweg et.al, 2017]

Goal 2: Uncertainty quantification

Two sources of uncertainty

epistemic

$p(\mathbf{w}|\mathcal{D})$

aleatoric

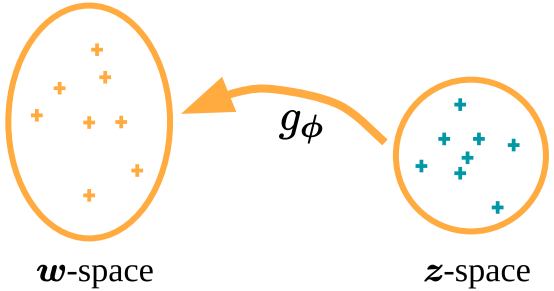
$p(\mathbf{y}|\mathbf{x}, \boldsymbol{w})$

$$\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \boldsymbol{\epsilon}$$

[Depeweg et.al, 2017]

High-level idea

- Approximate $f_{\boldsymbol{w}}$ with a Bayesian Neural Network

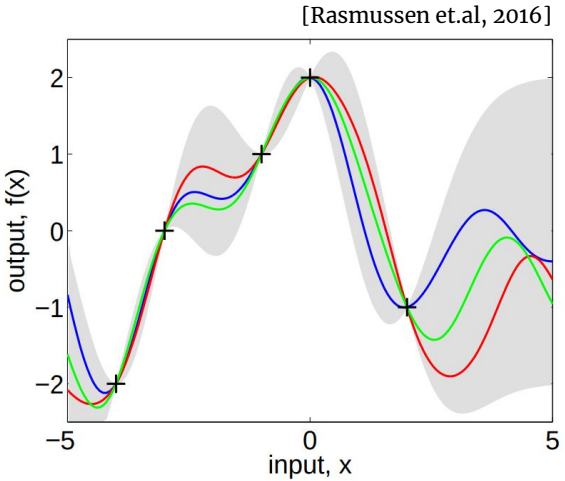


- Modeling + inference contributions

How to estimate function uncertainty?

Gaussian Process (GP)

[Rasmussen et.al, 2016]



$$f(x) \sim \mathrm{GP}\left(m(x), k(x, x')\right)$$

Drawbacks of GPs

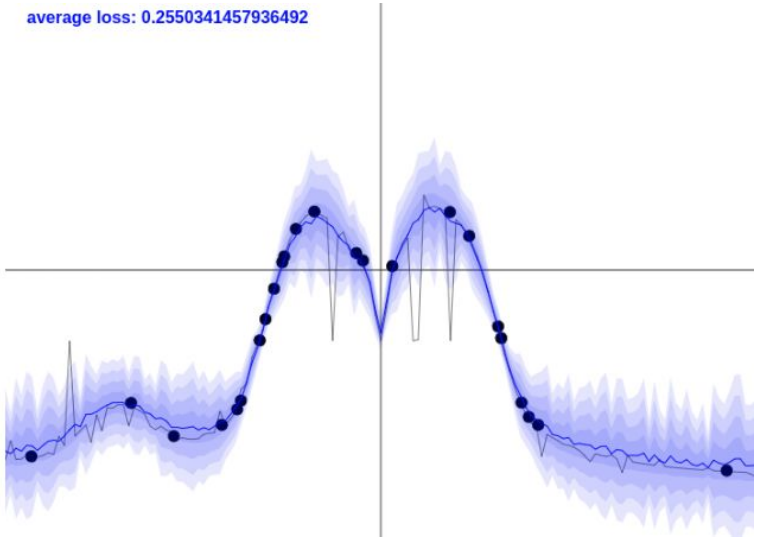
- Scalability
- Kernel learning is not trivial

Alternative: Neural Networks with uncertainty

- Ensemble of Neural Networks
  [Lakshminarayanan et al., 2017; Pearce et.al, 2018]

- Bayesian Neural Networks
  [Buntine et al., 1991; MacKay, 1992; Neal, 1993]

Bayesian Neural Network (BNN)



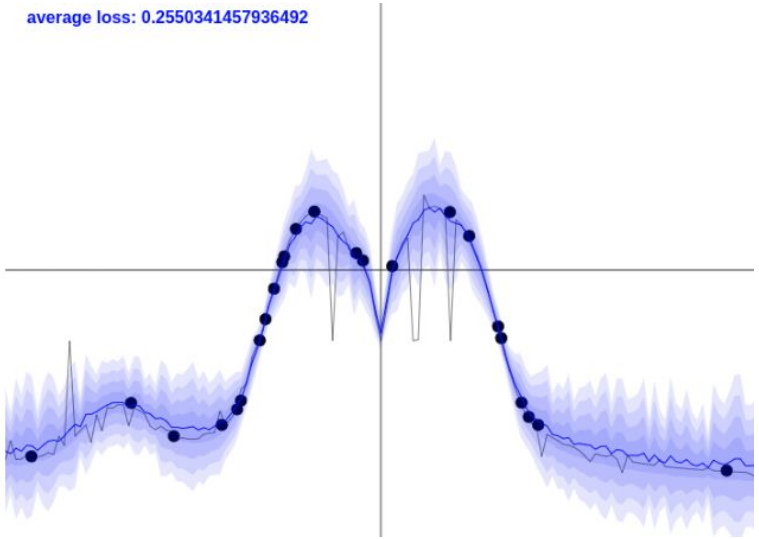
[What my deep model does not know, post of Yarin Gal, 2015]

$$\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$$

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

$$\boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

Bayesian Neural Network (BNN)



[What my deep model does not know, post of Yarin Gal, 2015]

$$\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$$

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

$$\boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

Quantities of interest:

- Posterior of the weights $p(\boldsymbol{w}|\mathcal{D})$
- Predictive distribution

$$p(\mathbf{y}^\star|\mathbf{x}^\star, \mathcal{D}) = \int p(\mathbf{y}^\star|\mathbf{x}^\star, \boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}$$

$$p(\boldsymbol{w}|\mathcal{D})$$

is intractable!

Inference options:

- Markov Chain Monte Carlo
  Hamiltonian Monte Carlo [Neal, 1993]
- Variational Inference
  [Graves, 1993] [Blundell et.al, 2015]

Variational Inference for BNNs

[Blundell et.al, 2015]

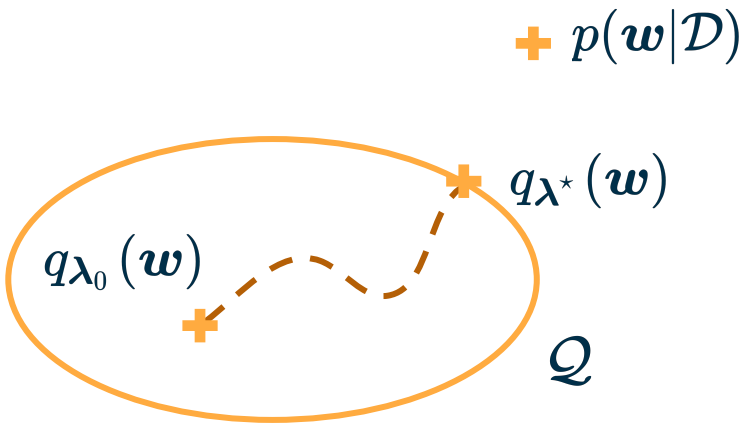
Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}}\, D_{\mathrm{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$

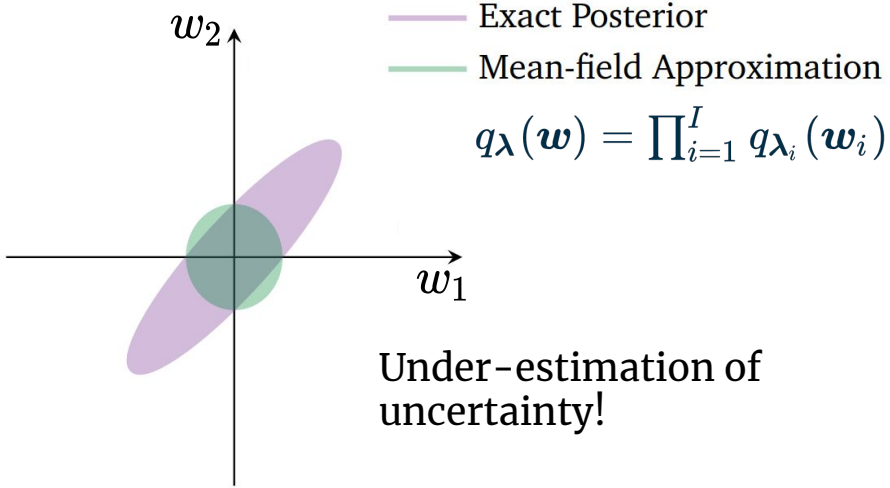
$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})\Big] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

Is mean-field VI good enough?

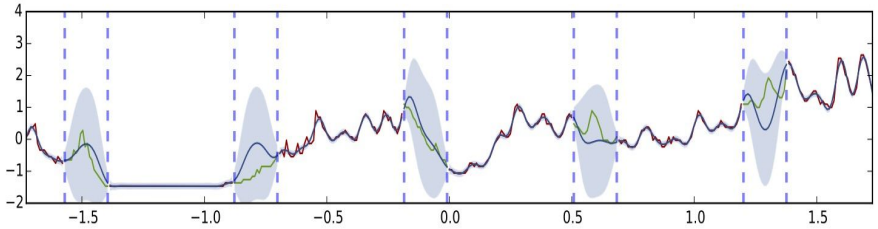


Under-estimation of uncertainty!

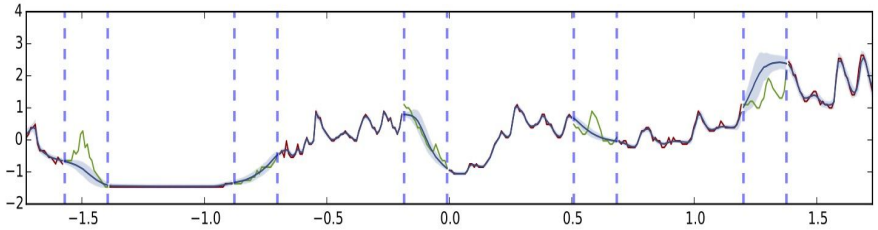
- More **flexible variational** approx. in weight Space [Louizos et al, 2016; 2017]

Example on solar irradiance dataset [Gal et.al, 2015]

GP



BNN



Standard BNN

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \;\; \boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}),$$

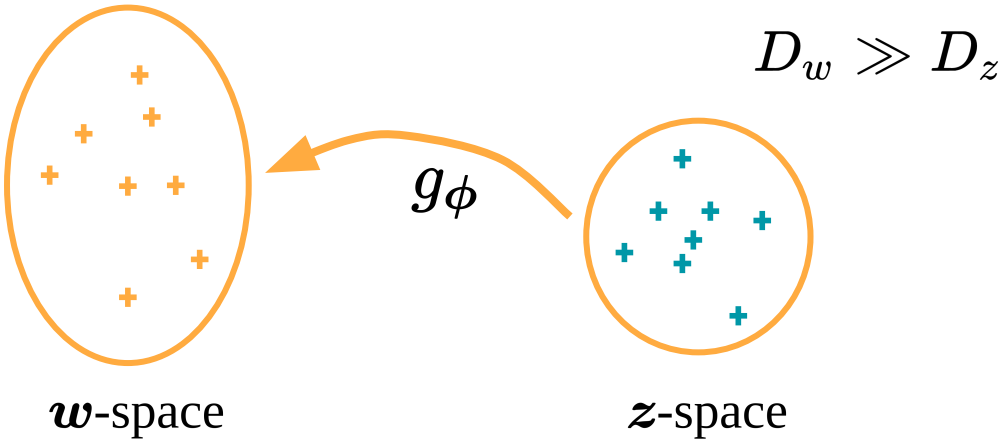
$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

Weight redundancy
[Denil et.al, 2013]

Projected BNN

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \; \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z}), \quad \boldsymbol{\phi} \sim p(\boldsymbol{\phi}),$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$



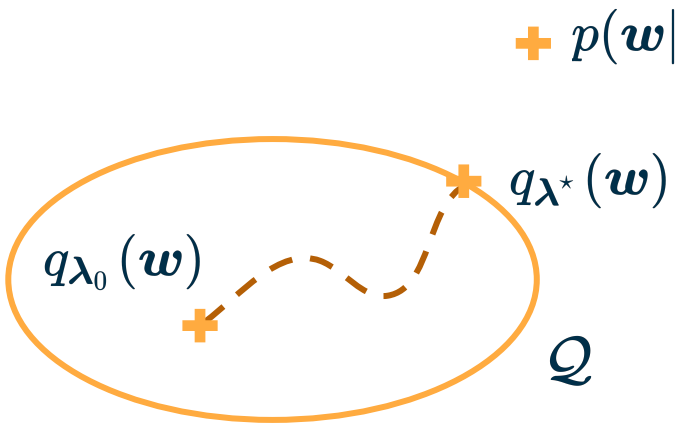
How about inference?

Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}}\, D_{\mathrm{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})\Big] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

How about inference?

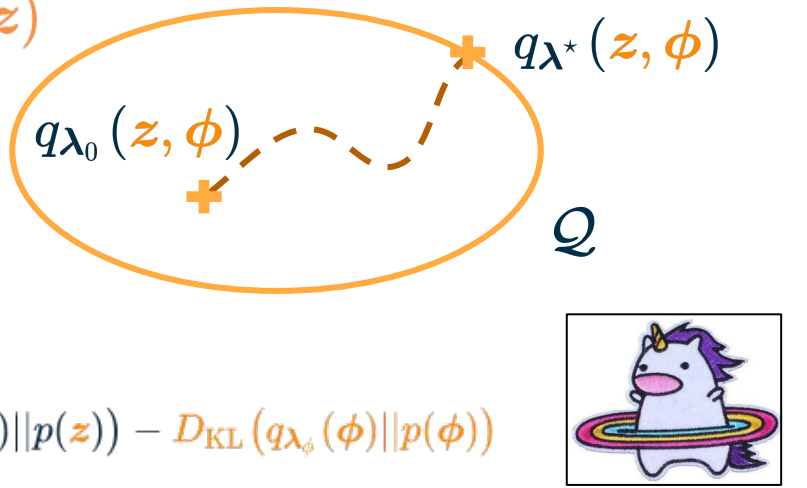
Objective: approximate $p(\boldsymbol{z}, \boldsymbol{\phi}|\mathcal{D})$

$$\boldsymbol{z} \sim q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}), \quad \boldsymbol{\phi} \sim q_{\lambda_\phi}(\boldsymbol{\phi}), \quad \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z})$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}} \, D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})||p(\boldsymbol{z}, \boldsymbol{\phi}|\mathcal{D})\right)$$

$$\updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmax}} \, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p\big(\boldsymbol{y}|\boldsymbol{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\big)\Big] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})||p(\boldsymbol{z})\right) - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi})||p(\boldsymbol{\phi})\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

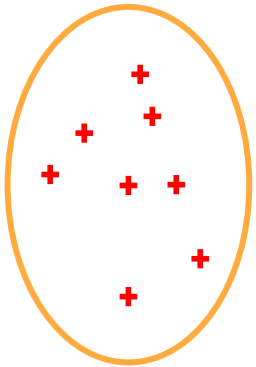
$$\underset{\boldsymbol{\lambda}^{\star}}{\operatorname{argmin}}\, D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) || p(\boldsymbol{z}, \boldsymbol{\phi} | \mathcal{D})\right)$$

jointly is hard!

Our solution: find smart initialization

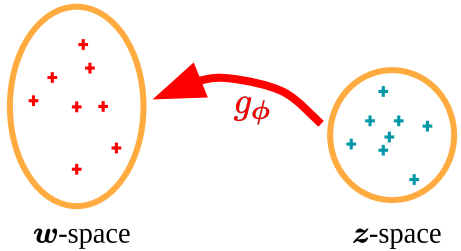
Solution: 3-stage Inference Framework

1. Characterize weight space



Train ensemble of neural networks

2. Find point estimate g_ϕ



Train an autoencoder

3. Black-box VI (BBVI)

$$D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z},\boldsymbol{\phi})||p(\boldsymbol{z},\boldsymbol{\phi}|\mathcal{D})\right)$$

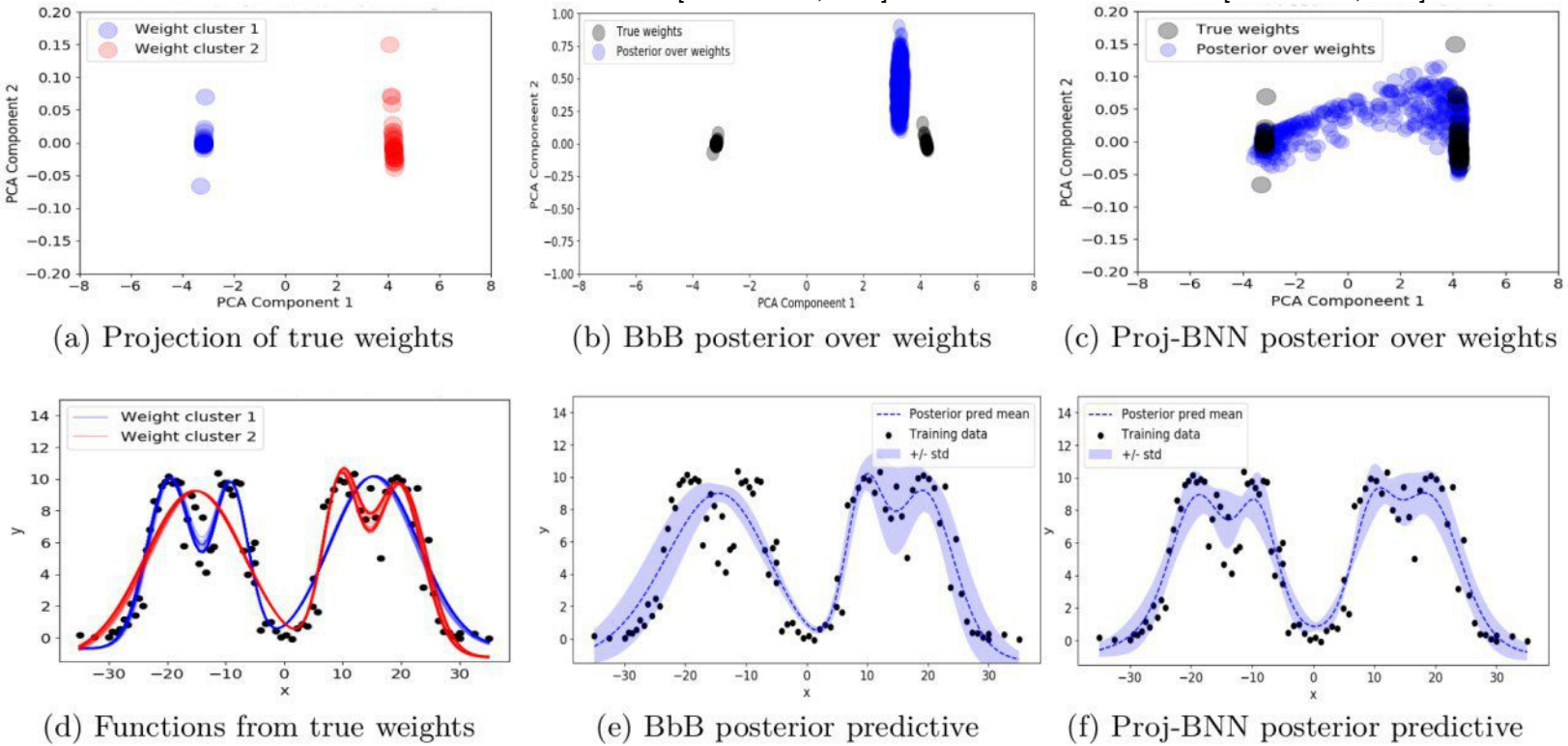
Principled BBVI with smart initialization

Results

Illustrative Toy Example

[Blundell et.al, 2015]

[Pradier et.al, 2018]



(a) Projection of true weights

(b) BbB posterior over weights

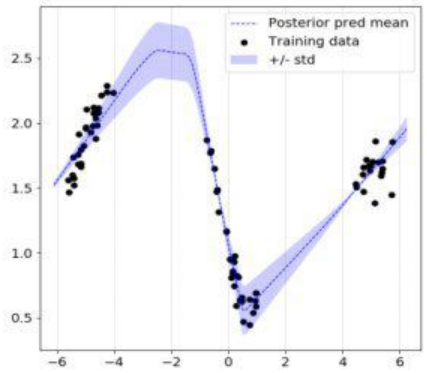
(c) Proj-BNN posterior over weights

(d) Functions from true weights

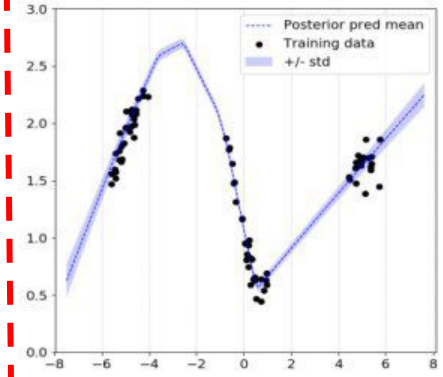
(e) BbB posterior predictive

(f) Proj-BNN posterior predictive

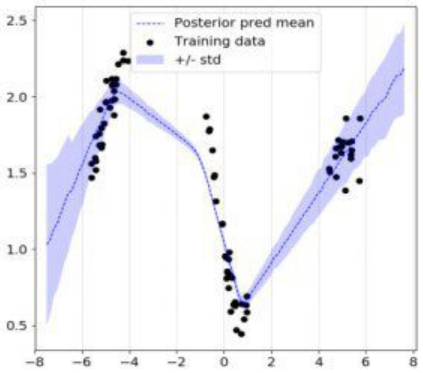
Results: Uncertainty estimation



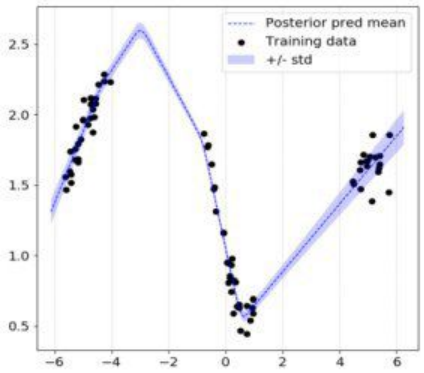
(a) Proj-BNN ($D_z = 2$)



(b) BbB



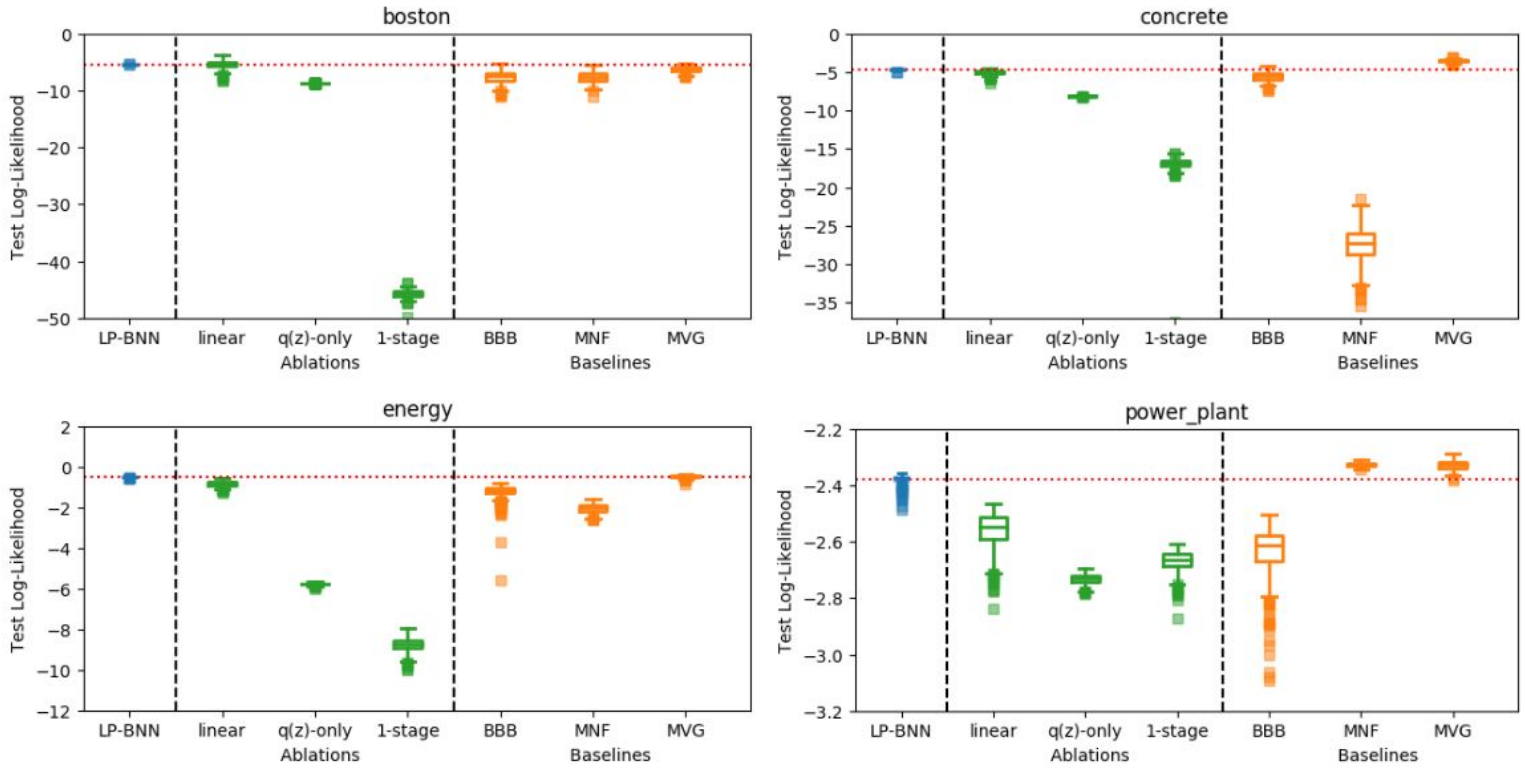
(c) MNF



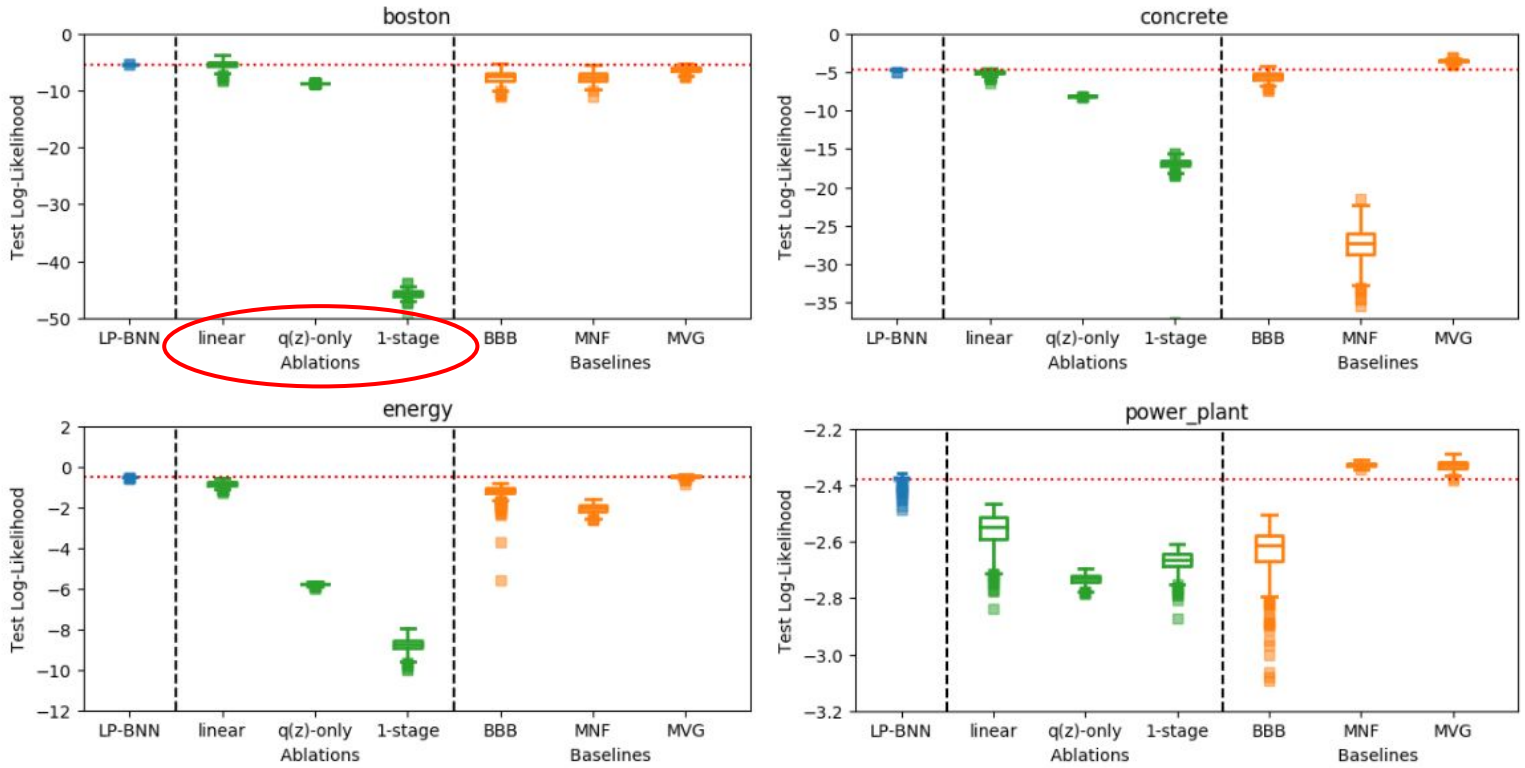
(d) MVG

- BbB: Bayes by Back Prop [Blundell et.al, 2015]
- MVG: Multivariate Gaussians [Louizos et.al, 2016]
- MNF: Multiplicative Normalizing Flow [Louizos et. al, 2017]

Results: Generalization

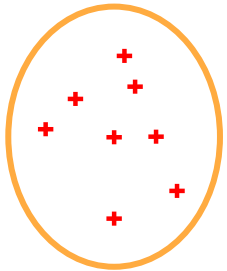


Results: Generalization

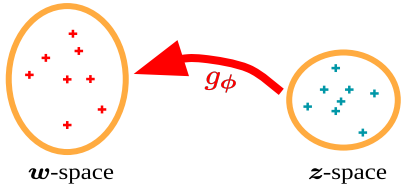


Results: Generalization (Ablations)

1. Characterize w-space



2. Find point estimate g_ϕ



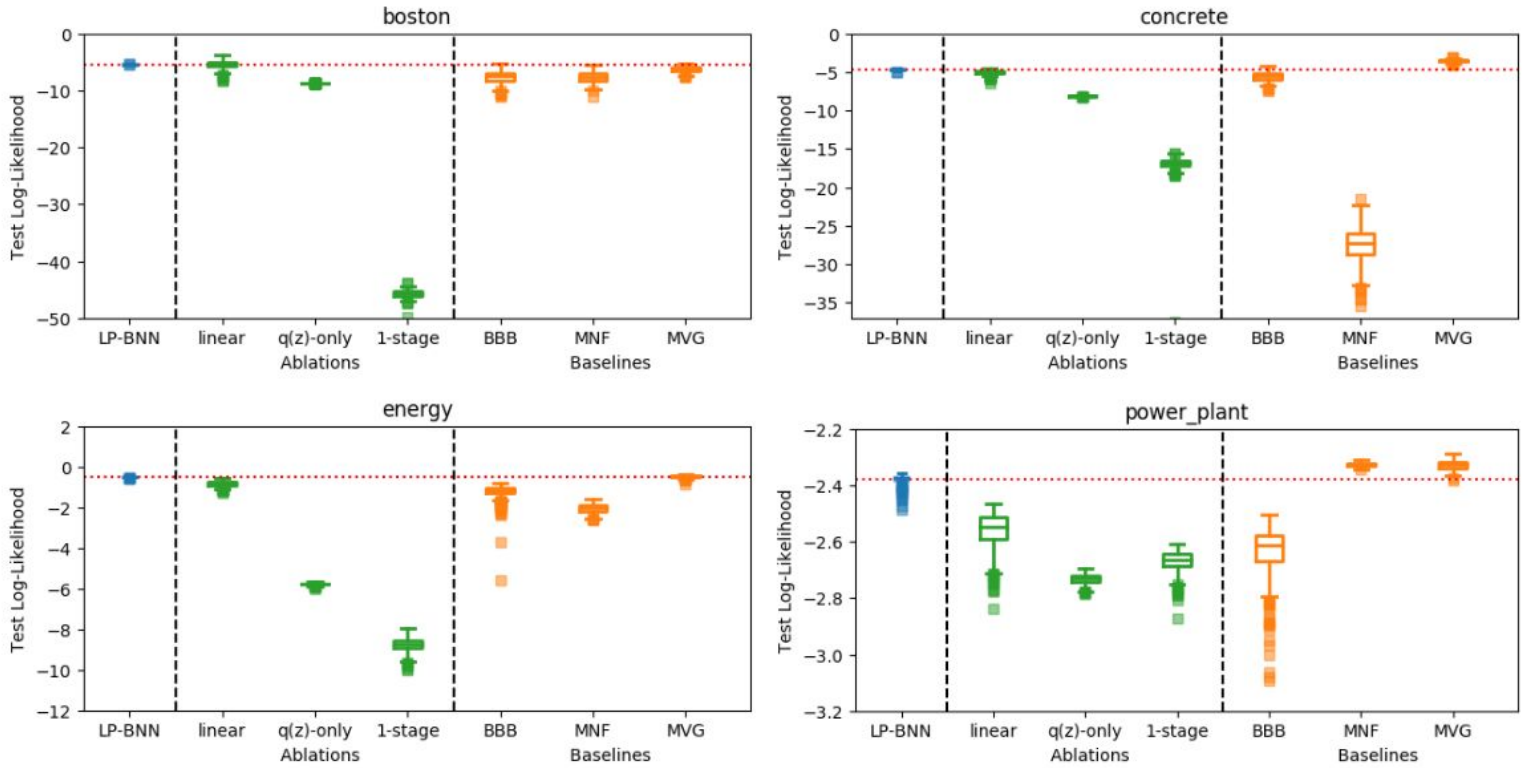
3. Black-box VI (BBVI)

$$D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z},\boldsymbol{\phi})||p(\boldsymbol{z},\boldsymbol{\phi}|\mathcal{D})\right)$$

1-stage	✗	✗	✓
linear	✓	linear	✓
q(z) only	✓	✓	$q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$

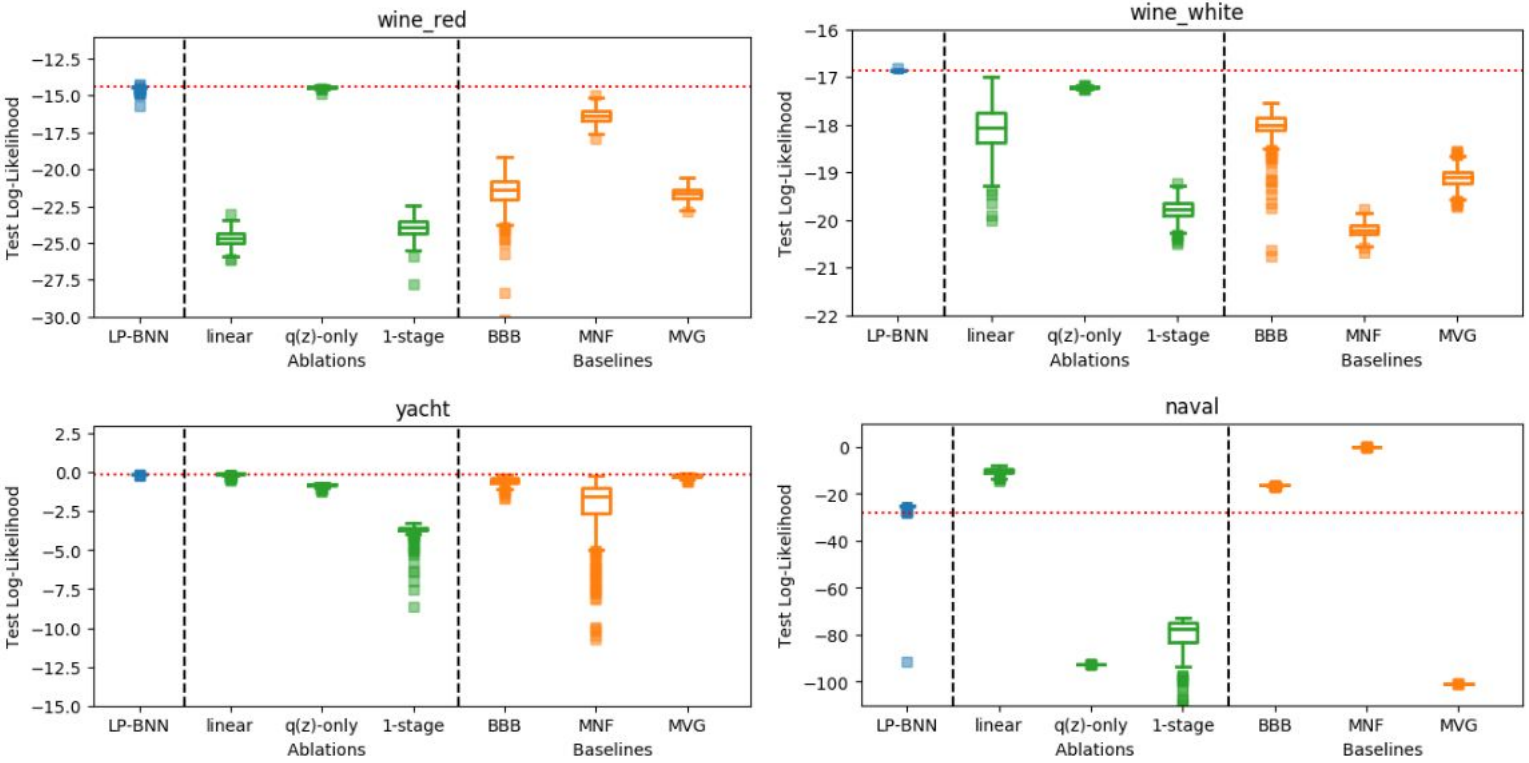
Results: Generalization

https://arxiv.org/abs/1811.07006

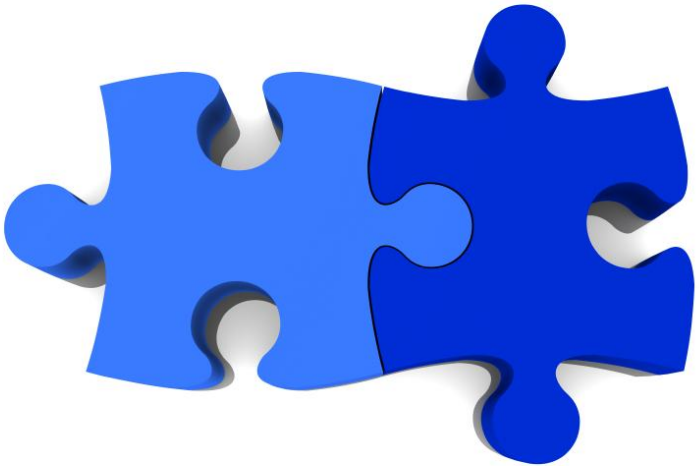


Results: Generalization

https://arxiv.org/abs/1811.07006



Conclusions



In this talk, two applications of LVMs

1. **Data exploration**
   a. Infinite latent feature model for heterogeneous datasets
   b. Global and group specific factors

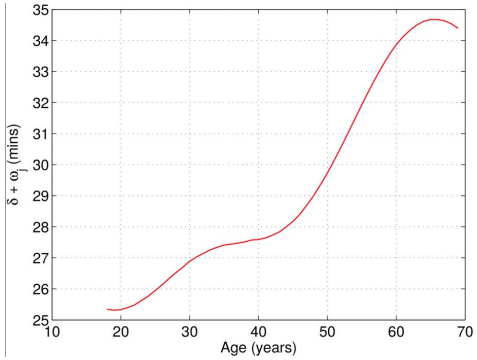
   https://ivaleram.github.io/GLFM/

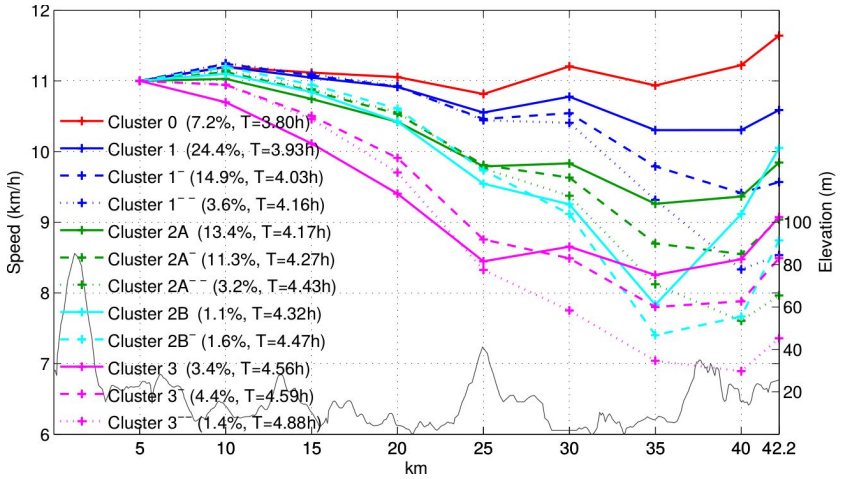
2. **Uncertainty quantification**
   a. Alternative modeling for BNNs
   b. Better approximate inference

   https://arxiv.org/abs/1811.07006

Other projects...

Sport Science

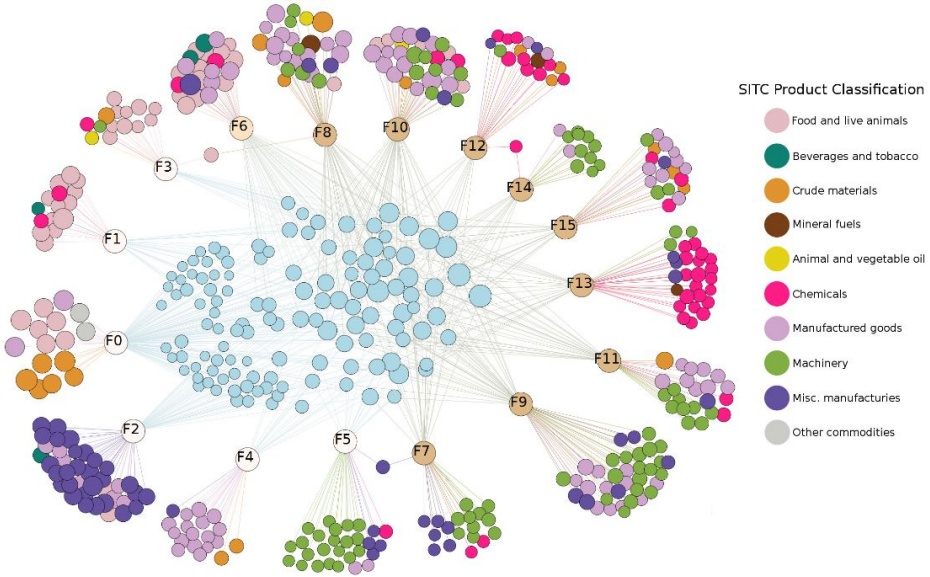


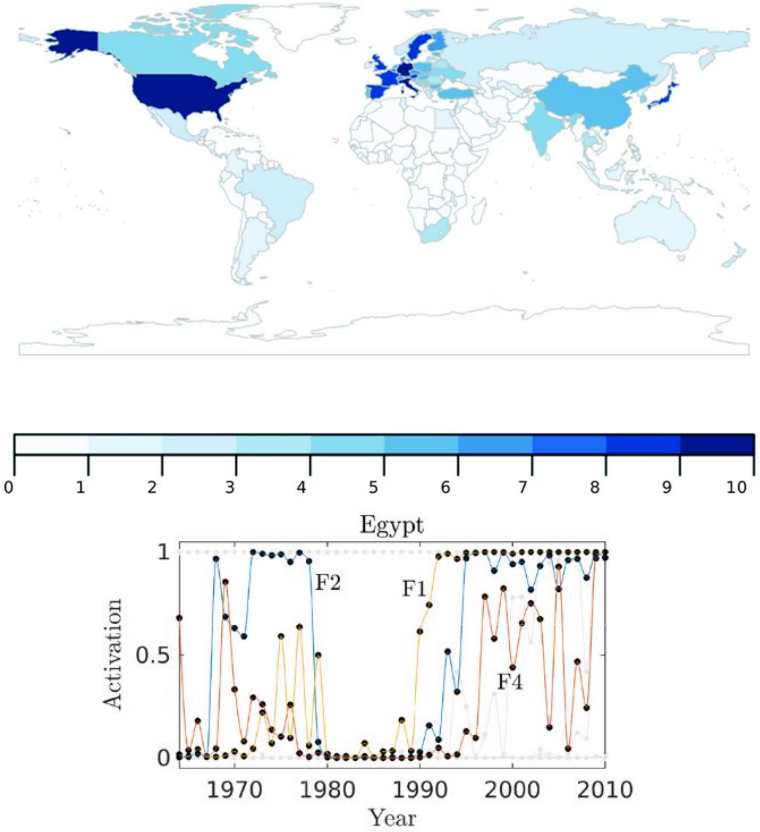


M. F. Pradier, F. J. R. Ruiz, and F. Perez-Cruz. **Prior Design for Dependent Dirichlet Processes: An Application to Marathon Modeling**. *PlosONE*. 2016.

Other projects...

Economics

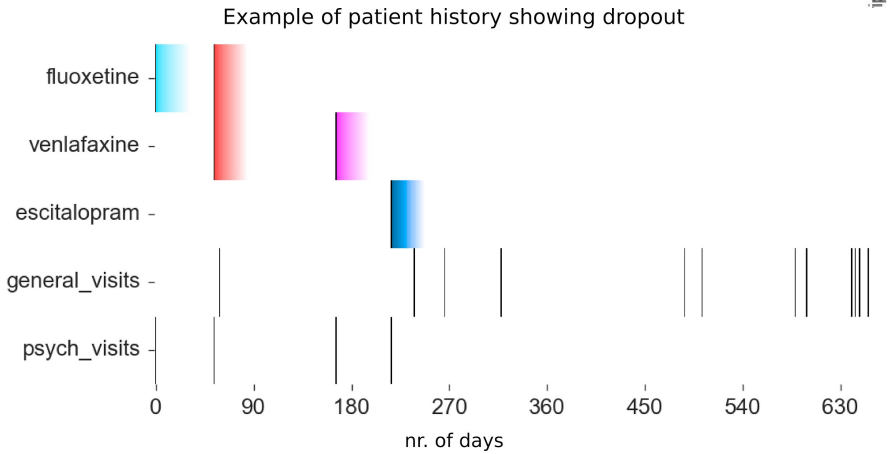


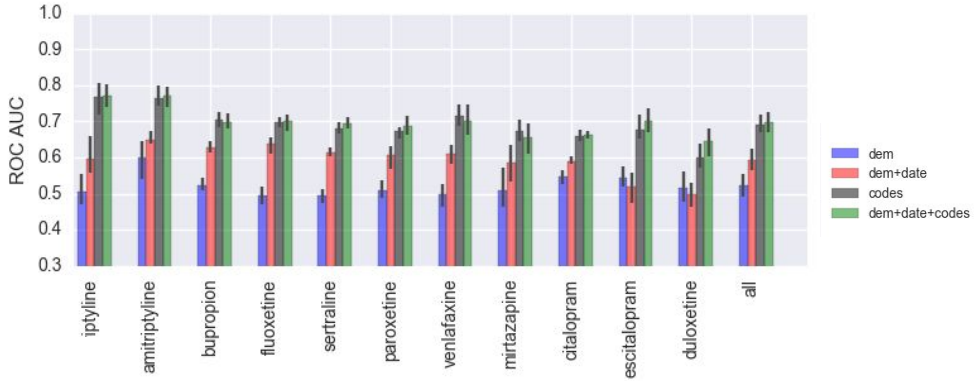


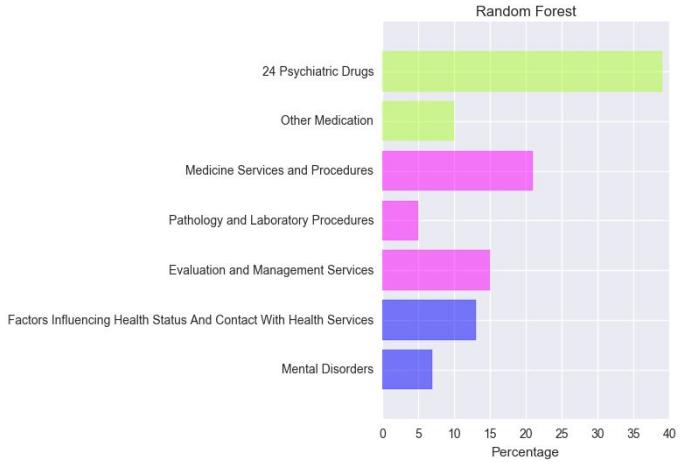
Z. Utkovski, M. F. Pradier, V. Stojkoski, L. Kocarev and F. Perez-Cruz. **Economic Complexity Unfolded: An Interpretable Model for the Productive Structure of Economies**. *PlosONE*. 2018.

Other projects...

Medicine: healthcare in psychiatry







M. F. Pradier, T. H. McCoy, M. Hughes, R. H. Perlis and F. Doshi-Velez. **Predicting Treatment Discontinuation after Antidepressant Initiation**. *In submission to JAMA Psychiatry*. 2018.

Current research agenda

Impact in real-world problems:

- Personalize prescription of antidepressants

ML research questions:

- How to adapt model complexity automatically?
- How to better quantify model uncertainty?
- How to make models easy-to-interpret?
- How to combine expert knowledge with insights from data?

Contact: *melanie@seas.harvard.edu*

https://melaniefp.github.io/

Thank you!

Special thanks to:

- Finale Doshi-Velez
- Weiwei Pan
- Michael Hughes
- All members of dtak!
- Francisco Rodriguez Ruiz
- Fernando Perez-Cruz
- Isabel Valera
- Maria Lomeli
- Zoubin Ghahramani
- Oscar Puig
- Francesca Milletti


Weiwei Pan


Jiayu Yao


Soumya Ghosh


Maia Jacobs


Finale Doshi-Velez

CRCS Center for Research on Computation and Society
at Harvard John A. Paulson School of Engineering and Applied Sciences

HDSI | Harvard Data Science Initiative

https://melaniefp.github.io/

Interpretable Machine Learning

Interpretability

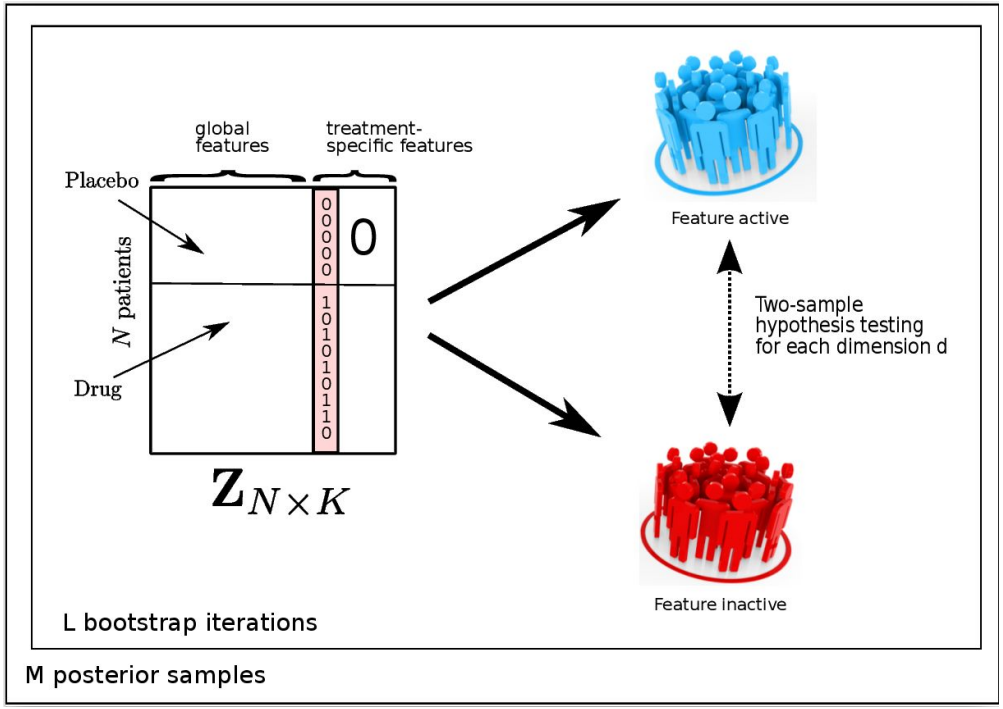
- "ability to explain or to present in understandable terms to a human" (Doshi-Velez and Kim, 2017)
- requirement in the 2018 EU General Data Protection Regulation (Goodman et.al. 2016)

Interpretable Machine Learning

- Interpretable models to explain black-boxes
  - Local Interpretable Explanations (Ribeiro et.al, 2016)
  - Interpretable Decision Sets (Lakkaraju et.al, 2016)
- Interpretable models from scratch
  - Tree-regularization of deep models (Wu et.al, 2017)
  - Input-gradient regularization (Ross et.al, 2017)

In this talk, interpretability via prob. graphical models

Statistical methodology for biomarker discovery



M. F. Pradier, B. Reis, L. Jukofsky, F. Milletti, T. Ohtomo, F. Perez-Cruz, and O. Puig. **Case-control Indian Buffet Process identifies biomarkers of response to Codrituzumab**. *Accepted to BMC Cancer*. 2019.

Prediction-constrained Autoencoder

$$\{\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star\} = \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmin}}\ \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min_{\boldsymbol{\theta},\boldsymbol{\phi}} \left\{ \frac{1}{R}\sum_{r=1}^{R} \left(\mathbf{w_c}^{(r)} - g_{\boldsymbol{\phi}}\left(f_{\boldsymbol{\theta}}\left(\mathbf{w_c}^{(r)} \right) \right) + \gamma^{(r)} \right)^2 \right.$$
$$\left. + \beta\, \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\frac{1}{R}\sum_{r=1}^{R} \log p\left(y | x, g_{\boldsymbol{\phi}}\left(f_{\boldsymbol{\theta}}\left(\mathbf{w_c}^{(r)} \right) \right) \right) \right] \right\},$$