Statistics and Machine Learning: Towards a Closer Integration

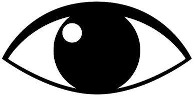
Tim van Erven

University of Amsterdam

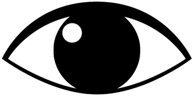
1st Workshop on AI & Mathematics, June 9, 2022

Perspectives on Data: The Two Cultures

Statistics



Machine Learning



In reference to Breiman, 2001, *Statistical Modeling: The Two Cultures*

Perspectives on Data: The Two Cultures

Statistics

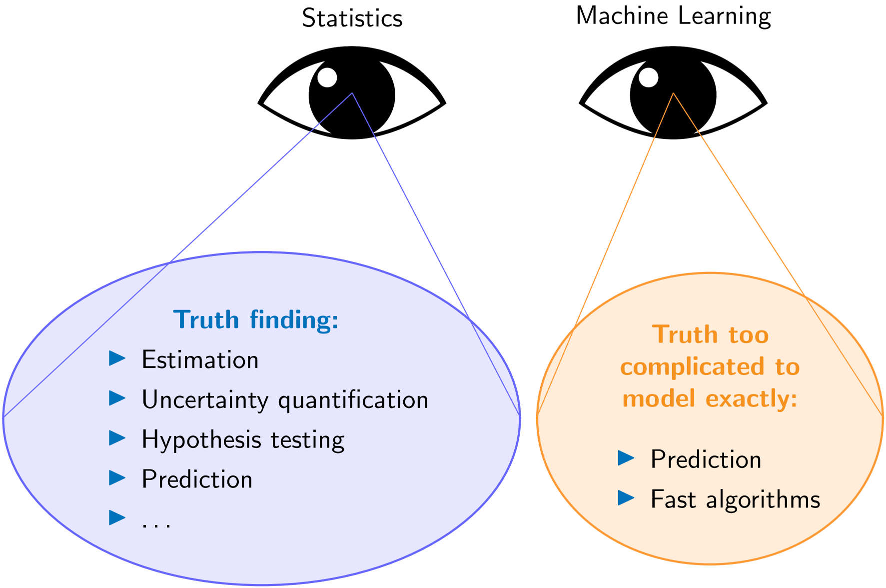
Machine Learning

Truth finding:

- Estimation
- Uncertainty quantification
- Hypothesis testing
- Prediction
- ...

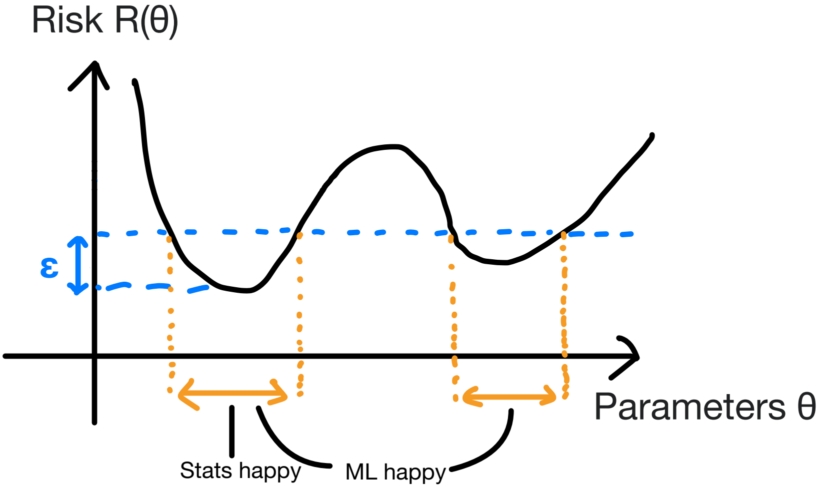
In reference to Breiman, 2001, *Statistical Modeling: The Two Cultures*

Perspectives on Data: The Two Cultures



In reference to Breiman, 2001, *Statistical Modeling: The Two Cultures*

Perspectives on Data: The Two Cultures



- Both care about **small risk**, and estimate it using empirical risk

1. The Sparse Normal Sequence Model

Want to recover signal $\theta \in \mathbb{R}^n$ from noisy observations $Y \in \mathbb{R}^n$:

$$
\begin{aligned}
&Y_i = \theta_i + \varepsilon_i, \quad i = 1, \dots, n \\
&\varepsilon_i \sim \mathcal{N}(0, 1)
\end{aligned}
$$

Sparsity: nr. of non-zero components s in θ is small: $s = o(n)$.

1. The Sparse Normal Sequence Model

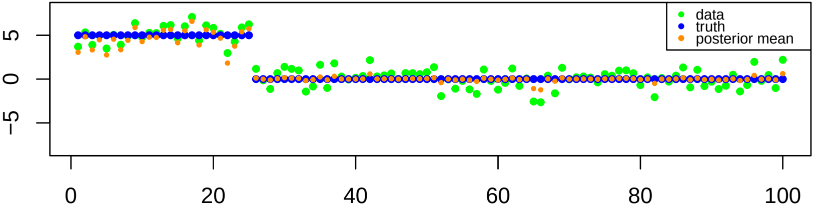
Want to recover signal $\theta \in \mathbb{R}^n$ from noisy observations $Y \in \mathbb{R}^n$:

$$Y_i = \theta_i + \varepsilon_i, \quad i = 1, \ldots, n$$
$$\varepsilon_i \sim \mathcal{N}(0, 1)$$

Sparsity: nr. of non-zero components s in θ is small: $s = o(n)$.

Bayesian prior ideal to model sparsity:

1. Draw sparsity level $s \sim \pi_n$
2. Draw subset of non-zero coordinates $\mathcal{S} \subset \{0, 1, \ldots, n\}$ of size $|\mathcal{S}| = s$ uniformly at random.
3. $\theta_i \sim G$ for $i \in \mathcal{S}$, $\quad \theta_i = 0$ for $i \notin \mathcal{S}$



1. The Sparse Normal Sequence Model

Want to recover signal $\theta \in \mathbb{R}^n$ from noisy observations $Y \in \mathbb{R}^n$:

$$Y_i = \theta_i + \varepsilon_i, \quad i = 1, \ldots, n$$
$$\varepsilon_i \sim \mathcal{N}(0,1)$$

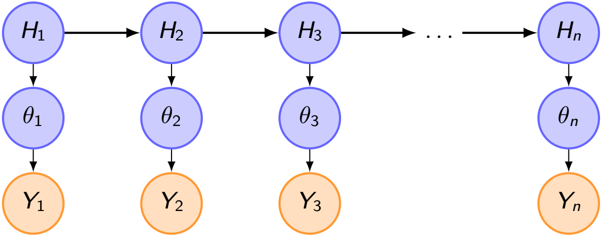
Sparsity: nr. of non-zero components s in θ is small: $s = o(n)$.

Bayesian prior ideal to model sparsity:

1. Draw sparsity level $s \sim \pi_n$
2. Draw subset of non-zero coordinates $\mathcal{S} \subset \{0, 1, \ldots, n\}$ of size $|\mathcal{S}| = s$ uniformly at random.
3. $\theta_i \sim G$ for $i \in \mathcal{S}$, $\quad \theta_i = 0$ for $i \notin \mathcal{S}$

- Under suitable conditions on π_n and G, the Bayes posterior distribution on θ contracts around the true θ at the **optimal rate** [Castillo & Van der Vaart, 2012].
- But **cannot compute this posterior efficiently** for $n \gg 300$ ...

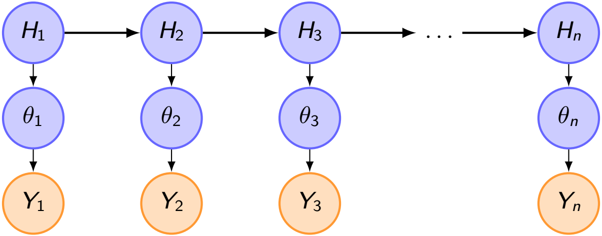
1. The Sparse Normal Sequence Model: Computation [Van Erven, Szabo, 2021]



- Hidden Markov model going back to [Volf, Willems, 1998] in the context of data compression and online machine learning:

$$H_i = \left(|\{j \in \mathcal{S} : j \leq i\}|, \mathbf{1}_{[i \in \mathcal{S}]}\right)$$

1. The Sparse Normal Sequence Model: Computation [Van Erven, Szabo, 2021]

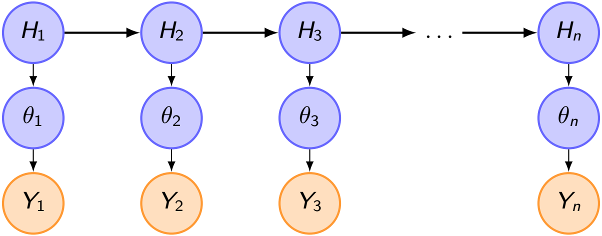


- Hidden Markov model going back to [Volf, Willems, 1998] in the context of data compression and online machine learning:

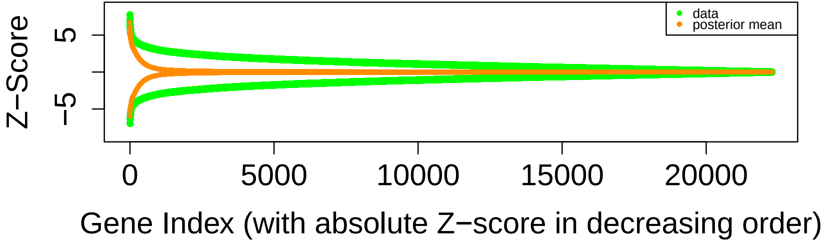
$$H_i = \left(|\{j \in \mathcal{S} : j \leq i\}|, \mathbf{1}_{[i \in \mathcal{S}]}\right)$$

- Can choose transition probabilities s.t. this **HMM is equivalent to the Bayesian model**, with $\mathcal{S}$ encoded in hidden states $H_1, \ldots, H_n$
- For HMMs with small hidden state there are **efficient algorithms**...

1. The Sparse Normal Sequence Model: Computation [Van Erven, Szabo, 2021]



Compute posterior on differential gene expression data with $n = 22\,283$ genes in just 2 minutes:



2. Computation & Generalization in Deep Learning

Non-convex Optimization:

- Millions of images: too many to process all at once
- Process one image at a time using stochastic gradient descent

2. Computation & Generalization in Deep Learning

Non-convex Optimization:

- Millions of images: too many to process all at once
- Process one image at a time using stochastic gradient descent

High-dimensional Setting:

- Still many more parameters than images (e.g. 25 times as many)
- Statistically obvious: we cannot estimate so many parameters unless we add constraints (e.g. restrict to L_p ball)

2. Computation & Generalization in Deep Learning

Non-convex Optimization:

- Millions of images: too many to process all at once
- Process one image at a time using stochastic gradient descent

High-dimensional Setting:

- Still many more parameters than images (e.g. 25 times as many)
- Statistically obvious: we cannot estimate so many parameters unless we add constraints (e.g. restrict to L_p ball)
- But even if you disable all standard regularization, it still works! [Zhang, Bengio, Hardt, Recht, Vinyals, 2017]
- So how are the parameters restricted?

2. Computation & Generalization in Deep Learning

Non-convex Optimization:

- Millions of images: too many to process all at once
- Process one image at a time using stochastic gradient descent

High-dimensional Setting:

- Still many more parameters than images (e.g. 25 times as many)
- Statistically obvious: we cannot estimate so many parameters unless we add constraints (e.g. restrict to L_p ball)
- But even if you disable all standard regularization, it still works! [Zhang, Bengio, Hardt, Recht, Vinyals, 2017]
- So how are the parameters restricted? **By the behavior of the optimization algorithm!**

2. Computation & Generalization in Deep Learning

Non-convex Optimization:

- Millions of images: too many to process all at once
- Process one image at a time using stochastic gradient descent

High-dimensional Setting:

- Still many more parameters than images (e.g. 25 times as many)
- Statistically obvious: we cannot estimate so many parameters unless we add constraints (e.g. restrict to L_p ball)
- But even if you disable all standard regularization, it still works! [Zhang, Bengio, Hardt, Recht, Vinyals, 2017]
- So how are the parameters restricted? **By the behavior of the optimization algorithm!**

Big open question: Can we **characterize subspace** searched by optimization methods (on realistic inputs) and prove it is **small enough to generalize**? See e.g. [Belkin et al., 2019].

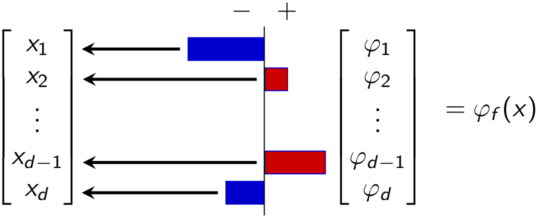
Related work in STAR: Schmidt-Hieber studies generalization of sparse statistical estimators for neural networks.

3. Explainable Machine Learning

Very new area:

- Classifier $f : \mathbb{R}^d \to \{-1, +1\}$
- User with features x is unhappy about $f(x)$
- Goal: explain why $f(x)$

Attribution methods indicate feature importance:



There is **no consensus** on what **importance** should mean,
so people focus on **necessary requirements**...

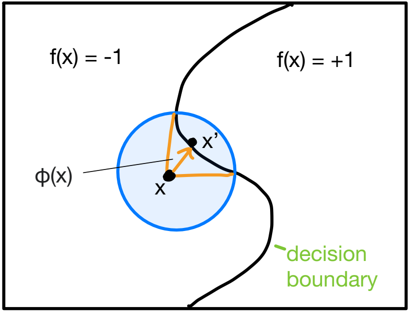
3. Explainable Machine Learning: Requirements

Suppose the user wants **Recourse**:

- User has limited ability to change x into x'
  - E.g. increase their credit score if bank loan was refused
- Then $\phi_f(x)$ should be a direction that tells them how to flip the class

Robustness:

- Similar users should get similar explanations, so want ϕ_f to be continuous.



3. Explainable Machine Learning: Requirements

Suppose the user wants **Recourse**:

- User has limited ability to change x into x'
  - E.g. increase their credit score if bank loan was refused
- Then $\phi_f(x)$ should be a direction that tells them how to flip the class

Robustness:

- Similar users should get similar explanations, so want ϕ_f to be continuous.

Theorem (Fokkema, De Heide, Van Erven, 2022)

There exist classifiers f for which it is **impossible** *for any attribution method ϕ_f to both provide recourse and be continuous.*

- See **poster** by Hidde Fokkema today!
- Result generalizes beyond classification
- Under (a restrictive) condition, we provide an exact characterization of the classifiers f that cause problems

Conclusion

Examples of fruitful interaction between Stats and ML:

1. Normal sequence model: idea from ML solves computational problem in Statistics
2. Generalization of deep learning: ideas from ML and Stats can fruitfully combine
3. Explainable machine learning: important new direction with room to be the Fisher of explainability

Did you know there is a **machine learning Netherlands mailing list**?

- Subscribe via my website: `www.timvanerven.nl`
- Use it to announce seminars, vacancies, etc.!