Three Principles of Data Science: Predictability, Computability and Stability (PCS)

Bin Yu
Statistics and EECS, UC Berkeley

Dahshu Journal club
Oct. 4, 2019
What is data science?

Data science is the re-merging of computational and statistical thinking in the context of domain problems.
Biomedical data problems are pressing

https://deepmind.com/blog/alphafold/

nanalyze.com

website of S. Saria at JHU
An empirically proven subfield of ML is predictive modeling

GenenTech:

- Risk scores and predictive modeling
- Tools to improve clinical trials design/analysis
- Basic biological research using –omics data
- Medical imaging automation
- ...

2011: Movie reconstruction using fMRI signals

S. Nishimoto  J. Gallant  A. Vu  T. Naselaris

Yuval Benjamini

Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies
This project was the starting point of our work on stability – I wanted to interpret the forward model to do science and for causality.
Culmination of 3+ years of work

Iterative random forests to discover predictive and stable high-order interactions

Sumanta Basu\textsuperscript{a,b,c,1}, Karl Kumbier\textsuperscript{d,1}, James B. Brown\textsuperscript{c,d,e,f,2}, and Bin Yu\textsuperscript{c,d,g,2}

Co-authors

S. Basu \quad K. Kumbier \quad B. Brown

Culmination of 3+ years of work
2018: Chan Zuckerberg Biohub Intercampus Award

One of the 6 awards

Project leaders:
Rima Arnout and Atul Butte (UCSF)
James Priest and Euan Ashleye (Stanford)
Ben Brown and Bin Yu (UC Berkeley)

Collaborators:
Chris Re (Stanford), Deepak Srivastava (UCSF)

Image credits: Rima Arnout.
2015-2019

The DeepTune framework for modeling and characterizing neurons in visual cortex area V4

Abbasi-Asl, Chen, Bloniarz, Oliver, Willmore, Gallant, and Y. (submitted, 2018)
https://www.biorxiv.org/content/early/2018/11/09/465534

Culmination of 3+ years of work

Reza Abbasi-Asl
Yuansi Chen
Adam Bloniarz

In collaboration with
Mike Oliver
Ben Willmore
Jack Gallant
Scientific machine learning (sML) (Machine learning or AI for science)

- It uses machine learning for scientific research to extract, from data, discoveries, theory, and knowledge

- It builds scientific principles in machine learning algorithms

- It iterates between the above two steps

- It subjects itself to the scientific standard of the domain
Our approach to sML

“Embedded” students/postdocs work on site, in the wet lab

Seed scientific problem(s)

Generalization

Generalization: workflow, algorithms, theory
sML calls for a data science
“strong inference”

“Why should there be such rapid advances in some fields and not in others? I think the usual explanations that we tend to think of such as the tractability of the subject, or the quality or education of the men drawn into it, or the size of research contracts are important but inadequate. I have begun to believe that the primary factor in scientific advance is an intellectual one.

These rapidly moving fields are fields where a particular method of doing scientific research is systematically used and taught, an accumulative method of inductive inference that is so effective that I think it should be given the name of ‘strong inference.’”

John R. Platt in “Strong Inference” (1964)
Science, Vol. 146, 347-353
DS is conducted in a DS life cycle an iterative process with “integrated” steps

Stability is a paramount consideration
What is data science strong inference?

- It is a particular process of carrying out a data science life cycle that is systematically used and taught, an accumulative process of inductive inference.

- It devises multiple or alternative pathways of the process including validation through prediction, interpretation, and follow-up experiments and seeks consistent or stable, valid, and reproducible conclusions.
A platform to integrate a myriad of works in the literature and to develop new methods ...

It is a minimum requirement for **interpretability**, **reproducibility**, and **scientific hypothesis generation or intervention design**.
Stability Principle

Stability is fundamental after predictability – both need computability. Limiting results such as CLT are stability results.

Stability Principle seeks stability based on clearly defined:

1. **Target(s) of interest** (relevant to the domain problem in the DS cycle)

2. **Appropriate perturbation(s)** to inputs to the DS cycle, including data cleaning methods, EDA, data, models/algorithms, synthetic data, and ad-hoc human decisions

3. **Appropriate Stability measure(s)** on the target(s) after perturbation

**Appropriateness** of perturbations and stability metrics is determined and debated based on subject knowledge, experience, judgment, and data collection process, resource, regulation, interpretability, …
Examples of data perturbation

• Cross-validation partition
• Bootstrap
• Subsampling
• Adding small amount of noise to data
• Bootstrapping residuals in linear regression and linear time series models
• Block-bootstrap
• *Data perturbations through synthetic data such as mechanistic simulation models
• *Adversarial examples in deep learning
• *Data under different environments/conditions (invariance)
• Differential Privacy (DP)
• ...
Examples of model/algorithm perturbation

• Robust statistics models
• Semi-parametric models
• **Lasso and Ridge models**
• Different modes of a non-convex empirical minimization
• **Different versions of Deep Learning algorithms**
• Different kernel machines
• Sensitivity analysis of Bayesian modeling
• ...
The PCS framework for DS life cycle:
  workflow and documentation

• PCS workflow:
  - predictability as a check for reality (algorithmic modeling)
  - computability as a necessity (algorithmic modeling)
  - stability as a minimum requirement for reproducibility and interpretability, and as a significant expansion of statistical inference (data modeling)

• PCS documentation: narratives and codes to explain assumptions and justify judgment calls
Remarks on P and C in PCS

• Predictability in broad sense: both global and local prediction performance and relative to different perturbations (including future data) and a first step in PCS inference

• Computability in the broad sense: computation considerations in the DS life cycle starting with data collection

• Computability in the narrow sense: computational scalability including storage, communication cost and speed, and using appropriate simulation models to algorithmic development and model validation
Dual roles of generative models (data modeling culture) in PCS

We consider both probabilistic or PDE-driven generative models

• They can concisely summarize past data and prior domain knowledge with parameters in them estimated by current data

• They can also be used to generate synthetic data as a form of regularization with current data to add stability
6-step PCS documentation is the bridge

Quantitative and qualitative narratives

Mental construct

Reality

Models

PCS documentation in Rmarkdown: narratives and codes

Stability formulation

Bootstrap sampling is a widely accepted perturbation scheme for problems in genomics that is a useful baseline for data where we have limited understanding of the dependencies. However, sequences located in similar regions of genome space (i.e. nearby on the DNA) exhibit dependent behavior that is possible to account for. In particular, enhancers that perform redundant tasks known as “shadow enhancers” are believed to confer robustness to regulatory processes (Hong, Hendrix, and Levine 2008). (Cannavò et al. 2016) studied shadow enhancers in detail and found that over 70% of loci they examined have anywhere from 2-5 shadow enhancers (Cannavò et al. 2016) with highly overlapping patterns of activity. To account for this potential dependency along the genome, we also consider block bootstrap perturbations using blocks of 5 and 10 sequences. We define the stability of an interaction to be the proportion of times it is recovered by RIT across $B = 100$ RFs trained on an outer layer of bootstrap samples using the 3 proposed perturbation schemes.

```
# Block bootstrap for blocks of size 5 and 10
block5.tr <- makeBlocks(gene.coords, idcs=train.id, size=5)
block10.tr <- makeBlocks(gene.coords, idcs=train.id, size=10)
block5.tst <- makeBlocks(gene.coords, idcs=test.id, size=5)
block10.tst <- makeBlocks(gene.coords, idcs=test.id, size=10)
```

iPython or Jupyter Notebook could also be used.
How to choose perturbations in PCS?

• One can never consider all possible perturbations

• A pledge to the stability principle in PCS would lead to null results if too many perturbations were considered

• PCS requires documentation on the appropriateness of all the perturbations

• To avoid null results, PCS encourages careful and well-founded choices of the perturbations through PCS documentation.
Causality evidence spectrum

Mechanistic
Individual level

Stable, replicable

Average effect
Group level

Effect depends on the group
Stability implicit in causal inference: e.g. SUTVA

PCS works towards causality:

Predictability + stability (+ computability)

interpretability and hypothesis generation
Frontier in ML/Stats: interpretation

EU's General Data Protection Regulation (GDPR) (2016) gives a “right” to explanation, and demands ML/Stats algorithms to be human interpretable

Image credit: https://christophm.github.io/interpretable-ml-book/
What is interpretable ML (iML)?
(Murdoch, Singh, Kumbier, Abbasi-Asl, and Y., accepted by PNAS, 2019)
“Interpretable Machine Learning: Definitions, Methods and Applications”

https://arxiv.org/abs/1901.04592

“We define interpretable machine learning as the extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model. Here, we view knowledge as being relevant if it provides insight for a particular audience into a chosen problem. These insights are often used to guide communication, actions, and discovery.”
iML-PDR in one figure

(P) Predictive accuracy

Model

(D) Descriptive accuracy

Post hoc analysis

(R) relevancy

Problem, Data, & Audience

Iterate

R is key in the trade-off of P and D
Desirable properties of model-based interpretability

• Sparsity (e.g. sparse logistic regression for lung cancer prediction)

• Simulatability (e.g. decision tree for lung cancer prediction)

• Modularity (e.g. generalized additive models, layers in DL)

• Domain-based feature engineering (e.g. credit score)

• Model-based feature engineering (e.g. clustering and dimensionality reduction like PCA)

Murdoch et al (2019) contains iML references and examples to illustrate PDR.
**PCS inference (basic)**

1. **Problem formulation:** Translate the domain question to be answered by a model/algorithm (or multiple of them and seek stability). Specify a target of interest.

1. **Prediction screening:** Filter models/algorithms based on prediction accuracy on held out test data – a sample split approach (it helps assess model bias)

1. **Target value perturbation distribution:** Evaluate the target of interest across “appropriate” data and model perturbations

1. **Perturbation interval reporting:** Summarize the target value perturbation distribution.

**PCS documentation:** transparent narratives and codes on Rmarkdown or Jupyter Notebook
Feature importance simulation study

simulation results for lasso feature selection in linear model
n=1000, p=630

Adding another method: Lasso (CV)+ asymptotic normal approx.
PCS theory after good PCS empirical evidence to analyze iterative Learning Algorithms

Algorithmic stability ↔ Equivalent ↔ Generalization

Trade-off

“Stable algorithms can not converge too fast...”

Practical computability: convergence rate

“Stability and convergence trade-off of iterative optimization algorithms”
Case-study of PCS: iRF (Basu et al, 2018)

Iterative random forests to discover predictive and stable high-order interactions

Co-authors

S. Basu  K. Kumbier  B. Brown

Culmination of 3+ years of work
**Order-4** interaction regulate *eve* stripe 2 in *Drosophila* development

Goto et al. (1989), Harding et al. (1989), Small et al. (1992), Isley et al. (2013), Levine et al. (2013)
Regulatory interactions through predictability and stability

Natural phenomenon

Active enhancer

Inactive enhancer

Prediction

Interpretation

Transcription is initiated when activating transcription factors reach sufficient DNA occupancy.
Capturing the form of genomic interactions

- Interactions are high-order and combinatorial in nature
- Interactions can vary across space and time as biomolecules carry out different roles in varied contexts
- Interactions exhibit thresholding behavior, requiring sufficient levels of constitutive elements before activating

(Hartenstein, 1993)
(Wolpert, 1969; Jaeger and Reinitz, 2006)
(Spitz and Furlong, 2006)
From genomic to statistical interactions

Transcription is initiated when a collection of activating TFs achieve sufficient DNA occupancy

\[ R(x) = \prod_{i \in S} 1\{x_i > t_i\} \]

Order-\(S\) interaction,

\[ S \subseteq \{1, \ldots, p\}, |S| = s \]
Random Forests (RFs)
Breiman (2001)

Draw $T$ bootstrap samples and fit a modified CART to each sample.

1. Grow CART trees to purity
2. When selecting splitting feature, choose a subset of $m_{\text{try}}$ features uniformly at random and optimize CART criterion over subsampled features.
Our iterative Random Forests (iRFs)

Core ideas

1. Interpret RF decision paths
2. Stabilize RF decision paths
1. Assess interaction stability
Interpreting RF: decrease in Gini Impurity as importance measure of a feature

\[
I_G(\pi) - \frac{N_l}{N} \cdot I_G(\pi_l) - \frac{N_r}{N} \cdot I_G(\pi_r)
\]

Mean Decrease in Impurity:
On average, how much does splitting on a feature decrease the Gini Impurity?
Feature-weighted RF
Amaratunga et al. (2014)

Random Forest:
At each node of the decision tree, uniformly sample \( m_{\text{try}} \) features to evaluate splitting criteria.

Feature-weighted Random Forest:
At each node of the decision tree, sample \( m_{\text{try}} \) features with probability proportional to

\[
w \in \mathbb{R}_+^p
\]
Iteratively re-weighted RF stabilize decision paths

Iteratively re-weighted Random Forests

Iter 1
Input: $D, w^{(1)} \leftarrow (1/p, \ldots, 1/p)$
Output: $w^{(2)} \leftarrow$ Gini Importance

Iter 2
Input: $D, w^{(2)}$
Output: $w^{(3)} \leftarrow$ Gini Importance

Iterate through $K - 1$

Feature weights

```
1   2   3
4   5
1   2   3
4   5
...
```
Digression: Interactions in market baskets

\[ Z_1 \quad Z_2 \quad Z_3 \quad Z_4 \]

\[ I_1 \quad I_2 \quad I_3 \quad I_4 \]

Feature-index sets
Random Intersection Trees (RIT)
Shah and Meinshausen (2014): fast computation uses sparsity
Random Intersection Trees (RIT)
Shah and Meinshausen (2014)

Randomly sampled class-\(C\) observation \rightarrow \text{“Survived” interaction}
Random Intersection Trees (RIT)
Shah and Meinshausen (2014)

Randomly sampled class-$C$ observation → "Survived" interaction

Diagram showing the flow of reasoning or classification in Random Intersection Trees (RIT) with a focus on the interaction aspect.
Our Generalized RIT for Decision Trees
fast computation uses sparsity

\[ \mathcal{I}_{it} \subseteq \{1, \ldots, p\} \]  
*Feature-index set* for leaf node containing observation \( i = 1, \ldots, n \) in tree \( t = 1, \ldots, T \)

\[ \mathcal{Z}_{it} \in \{0, 1\} \]  
*Prediction* for the leaf node containing observation \( i = 1, \ldots, n \) in tree \( t = 1, \ldots, T \)

\[ S \leftarrow \text{RIT}(\{\mathcal{I}_{it}, \mathcal{Z}_{it}\}, C') \]
Stability bagging

Output feature interaction sets with stability scores:

\[
\{S, sta(S)\}
\]

\[
S \subseteq \{1, \ldots, p\}
\]

\[
sta(S) = \frac{1}{B} \cdot \sum_{b=1}^{B} 1(S \in S_b)
\]

Reference: (Breiman, 1996)
Computability of iRFs

- Same order as RFs \( O(p \times n \log n) \)
- Key difference between iRFs and RFs:

  **RIT** (Random Intersection Trees) \( O(p^\kappa) \) (\( \kappa \sim 1 \) for very sparse data)

  RIT is similar to **Stochastic Gradient Descent (SGD)** but for sparse 0-1 vectors in two ways:

  -- it uses one data point at each iteration
  -- updates are local (using the the current data point and a previous fit)

  RIT is also dissimilar to SGD in the sense that RIT uses a **tree construction** for updates, not a sequential updates – this eliminates possible solutions very quickly under sparsity
Case study: Enhancer activity in *Drosophila*

- **Response:** enhancer status
- **Drosophila blastoderm embryos:**
  - n=7809 genomic sequences
  - p=80 ChIP assays (TF binding, histone modifications)
  - Response: enhancer activity

(Bermen et al., 2002; Frise et al., 2010; Fisher et al., 2012; Kvon et al. 2014)

[Link to data source](http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg19)
iRF increases stability hence interpretability while maintaining predictive accuracy
iRF identifies 20 stable pairwise interactions in *Drosophila* – **80%** are proven physical interactions in the literature

<table>
<thead>
<tr>
<th>interaction (S)</th>
<th>sta(S)</th>
<th>references</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gt, Zld</td>
<td>1</td>
<td>Harrison et al. (2011); Nien et al. (2011)</td>
</tr>
<tr>
<td>Twi, Zld</td>
<td>1</td>
<td>Harrison et al. (2011); Nien et al. (2011)</td>
</tr>
<tr>
<td>Gt, Hb</td>
<td>1</td>
<td>Kraut and Levine (1991a,b); Eldon and Pirrotta (1991)</td>
</tr>
<tr>
<td>Gt, Kr</td>
<td>1</td>
<td>Kraut and Levine (1991b); Struhl et al. (1992); Capovilla et al. (1992); Schulz and Tautz (1994)</td>
</tr>
<tr>
<td>Gt, Twi</td>
<td>1</td>
<td>Li et al. (2008)</td>
</tr>
<tr>
<td>Kr, Twi</td>
<td>1</td>
<td>Li et al. (2008)</td>
</tr>
<tr>
<td>Kr, Zld</td>
<td>0.97</td>
<td>Harrison et al. (2011); Nien et al. (2011)</td>
</tr>
<tr>
<td>Gt, Med</td>
<td>0.97</td>
<td>–</td>
</tr>
<tr>
<td>Bcd, Gt</td>
<td>0.93</td>
<td>Kraut and Levine (1991b); Eldon and Pirrotta (1991)</td>
</tr>
<tr>
<td>Bcd, Twi</td>
<td>0.93</td>
<td>Li et al. (2008)</td>
</tr>
<tr>
<td>Hb, Twi</td>
<td>0.93</td>
<td>Zeitlinger et al. (2007)</td>
</tr>
<tr>
<td>Med, Twi</td>
<td>0.93</td>
<td>Nguyen and Xu (1998)</td>
</tr>
<tr>
<td>Kr, Med</td>
<td>0.9</td>
<td>–</td>
</tr>
<tr>
<td>D, Gt</td>
<td>0.87</td>
<td>–</td>
</tr>
<tr>
<td>Med, Zld</td>
<td>0.83</td>
<td>Harrison et al. (2011)</td>
</tr>
<tr>
<td>Hb, Zld</td>
<td>0.80</td>
<td>Harrison et al. (2011); Nien et al. (2011)</td>
</tr>
<tr>
<td>Hb, Kr</td>
<td>0.80</td>
<td>Nüsslein-Volhard and Wieschaus (1980); Jäckle et al. (1986); Hoch et al. (1991)</td>
</tr>
<tr>
<td>D, Twi</td>
<td>0.73</td>
<td>–</td>
</tr>
<tr>
<td>Bcd, Kr</td>
<td>0.67</td>
<td>Hoch et al. (1991, 1990)</td>
</tr>
<tr>
<td>Bcd, Zld</td>
<td>0.63</td>
<td>Harrison et al. (2011); Nien et al. (2011)</td>
</tr>
</tbody>
</table>
Stable interactions reflect Boolean-type rules

3rd or 4th or higher order interactions are suggestions for Crispr experiments.
iRF is backbone for our CZB biohub project

Get genetic information on phenotypes of interest

Next-generation statistical machine learning tools to find interacting gene variants at manageable computational cost

Image credit: Rima Arnout
Summary on PCS

“The PCS framework aims at responsible, reliable, reproducible and transparent analysis across fields... It can be used as a recommendation system for scientific hypothesis generation and experimental design. In particular, we propose (basic) PCS inference for reliability measures on data results, extending statistical inference to a much broader scope as current data science practice entails.” Y. and Kumbier (2019)

The PCS framework and iML-PDR are effective steps towards data science strong inference.

Case study: iterative Random Forests (iRF) and signed iRF (siRF) for hypothesis generation of Boolean interactions
Papers on PCS and iML

1.* Three principles of data science: predictability, computability and stability (PCS) (Y. and K. Kumbier, 2019)
https://arxiv.org/abs/1901.08152

2*. Interpretable machine learning: definitions, methods and applications
https://arxiv.org/abs/1901.04592
iRF and siRF papers and software

• iRF paper in PNAS (2019)

Open source R implementation: https://cran.r-project.org/web/packages/iRF/

• Refining interaction search through signed iterative Random Forests (s-iRF) by Kumbier, Basu, Brown, Celniker and Yu (2019)


Software: https://github.com/sumbose/iRF containing both iRF and s-iRF
Thanks to my group members and grants

Goal: quality research even if it is often slow
The primary skills taught are:

- DSIA teaches the reader skills that are adaptable to any data-based problem.
- Reliable and trustworthy solutions.
- Communication skills required to effectively formulate problems and find solutions.

Data Science In Action (DSIA) will teach the critical thinking, analytic, and communication skills required to effectively formulate problems and find reliable and trustworthy solutions.

What skills do we teach?

Data Science In Action (DSIA) will teach the critical thinking, analytic, and communication skills required to effectively formulate problems and find reliable and trustworthy solutions.

Core guiding principles

The DS Lifecycle

- Question formulation and data collection
- Communication of results
- Data cleaning and exploration
- Predictive

The Data Science Lifecycle is an iterative process that takes the analyst from problem formulation, data cleaning, exploration, algorithmic analysis, and finally to obtaining a verifiable solution that can be used for future decision-making.

Blending together concepts from statistics, computer science and domain knowledge, the data science life cycle is an iterative process that involves human analysts learning from data and refining their project-specific questions and analytic approach as they learn.

Three realms

Data → Algorithms → Decisions

Readers will learn to view every data problem through the lens of connecting the three realms:

1. The question being asked and the data collected (and the reality the data represents)
2. The algorithms used to represent the data
3. Future data on which these algorithms will be used to guide decision-making

Guiding the reader to connect the three realms is a means of guiding the reader through the data science lifecycle.

The PCS framework provides concrete techniques for finding evidence for the connections between the three realms.

- Predictability: if the patterns found in the original data also appear in withheld or new data, they are said to be predictable. If an analysis or algorithm finds predictable patterns, then these patterns are likely to be capturing real phenomena.
- Computability: algorithmic and data efficiency and scalability is essential to ensuring that the results and solutions (e.g. a predictive algorithm) can be applied to new data.
- Stability: minimum requirement for reproducibility. If results change in the presence of minor modifications of the data (e.g. via perturbations) or human analytic decisions, then there might not be a strong connection between the analysis/algorithms and the reality that underlies the data.

Intended Audience

Anyone who wants to learn the intuition and critical thinking skills to become a data scientist or work with data scientists.

Neither a mathematical nor a coding background is required.

DSIA could form the basis of a semester- or multi-semester-long introductory data science university course, either as an upper-division undergraduate or early graduate-level course.

An upcoming book: Data science in action
Berkeley’s DS Intellectual and Organizational Vision

Summary of the 2016 Report by the Faculty Advisory Board of the Data Science Planning Initiative

Prepared: 19 August 2016
Cathryn Carson, FAB Chair

Contents
A. Rationale for action: Why Berkeley, why now
B. Recommendations
   1. Organizational form: Core and connections
   2. Faculty FTE: Campus-wide surge and strategic foci
   3. Fundraising pillar and revenue generation
C. Situational challenges and next steps
D. The Faculty Advisory Board

CS/Stat Faculty co-creating and co-teaching data8.org and ds100.org

DS Major, Fall 2018 (first class graduated in 2019)

New Associate Provost of Div. of Data Science and Dean of I-school: Jennifer Chayes

Data8 Spring19 – 1500 students

Data100 Spring19: 1,100 students
Thank You!

Questions?

PCS

Prediction

Computability

Stability

Data  Data  Data  Data  Data  Data  Data  Data  Data  Data