Lecture 1: Introduction

Statistical Learning (BST 263)

Jeffrey W. Miller

Department of Biostatistics
Harvard T.H. Chan School of Public Health

Outline

Outline

Course overview

Course website and syllabus

Choosing among methods

Welcome to Statistical Learning (BST 263)

- Course website:
  https://canvas.harvard.edu/courses/55674
  - If you can't access the website, let me know ASAP.
  - Please read the syllabus (under Files / Course information).

- Instructor: Dr. Jeff Miller
- TAs: Yuri Ahuja, Kareem Carr, and Greyson Liu

- Textbook: James et al. (2013)
- Supplementary text: Friedman et al. (2009)

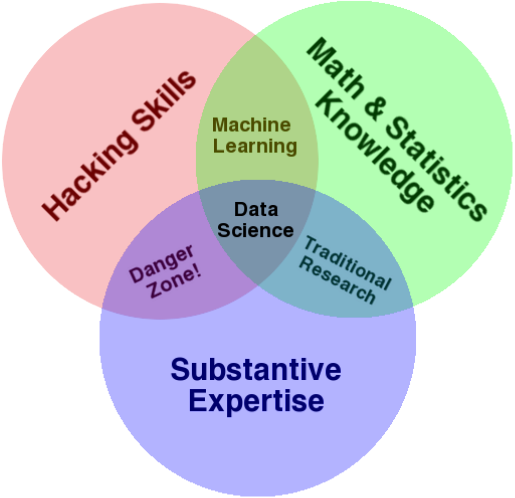
- Video supplements: My YouTube channel mathematicalmonk has over 250 videos on machine learning, probability, and information theory.

What is “statistical learning”?

- Statistical learning is a mixture of stats and machine learning.
- So. . . what is the difference between statistics and ML?
- Cynic: “ML is people in CS departments doing statistics.”
- There is huge overlap. . . main differences are in emphasis.

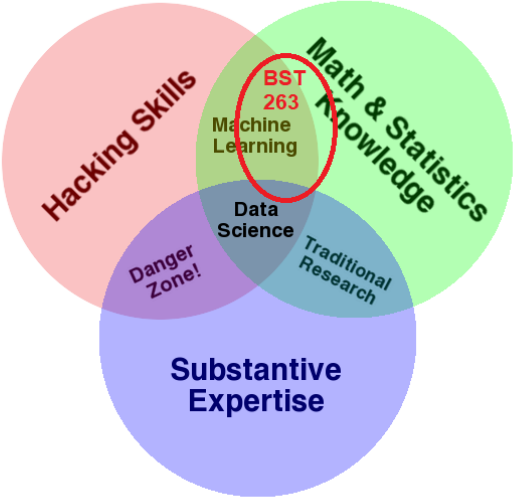
- Statisticians *tend* to focus more on:
  - uncertainty quantification
  - theoretical guarantees on performance
  - variations on well-established model classes
  - applications in science and medicine
- Machine learners *tend* to focus more on:
  - algorithms and computation
  - empirical performance on benchmark datasets
  - inventing complex new methods/models
  - applications in tech and industry

The scope of this course



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

The scope of this course



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

The scope of this course

- Cover the core statistical learning methods — how they work and how to use them.
  - Supervised learning (regression, classification)
  - Unsupervised learning (dimension reduction, clustering)

- Cover basic mathematical foundations of statistical learning.
  - Need math/stats to stay out of the "Danger Zone"!
  - There will be considerable mathematical content (including derivations/proofs) in lectures, homeworks, exams, etc.

- Coding experience for statistical learning in R language.
  - Labs and homeworks will involve considerable R coding.
  - You must be familiar with R (or learn it very quickly!)

What this course is not

- This is NOT a course on "hacking skills"...
  - We won't cover things like collecting data, data cleaning & wrangling, plotting, EDA, feature engineering, pipeline building, parallel computing, Hadoop/MapReduce, etc.
  - Take a different course if you want to learn these skills. Other courses in the HDS curriculum cover many of these things.
  - Hacking skills can more easily be learned on your own, whereas the math/stats is much harder to learn outside of a structured classroom environment.

- We will NOT cover neural networks or deep learning.
  - Deep learning is covered in BST 261: Data Science II.
  - BST 263 does not include deep learning, in order to avoid redundant content in the HDS curriculum.

- Do not expect to learn these things in this course, or you will be disappointed!

Outline

(Go over course website and syllabus)

Outline

Course overview

Course website and syllabus

Choosing among methods

No free lunch!

- No method dominates all others, across all problems.
- Roughly speaking, for any two methods, each will perform better on some problems compared to the other.
- Wolpert (1996) proves this in the "no free lunch theorem".
- That said, some methods seem to consistently perform better on the types of datasets that appear in practice.

Empirical comparison of methods on a variety of datasets

- Caruana and Niculescu-Mizil (2006) compared various supervised learning methods on a variety of benchmark datasets; see next slide.
- A common theme of top performing methods is the use of ensembling (such as bagging, boosting, stacking) to exploit "the wisdom of crowds."
- Interesting interview: "Using statistical algorithms for success in Kaggle's data science competitions"

Empirical comparison of methods on a variety of datasets

MODEL	CAL	COVT	ADULT	LTR.P1	LTR.P2	MEDIS	SLAC	HS	MG	CALHOUS	COD	BACT	MEAN
BST-DT	PLT	**.938**	.857	**.959**	**.976**	.700	.869	**.933**	.855	**.974**	**.915**	.878*	**.896***
RF	PLT	.876	.930	.897	.941	**.810**	.907*	.884	.883	.937	.903*	.847	.892
BAG-DT	–	.878	.944*	.883	.911	.762	.898*	.856	**.898**	.948	.856	**.926**	.887*
BST-DT	ISO	.922*	.865	.901*	.969	.692*	.878	.927	.845	.965	.912*	.861	.885*
RF	–	.876	.946*	.883	.922	.785	.912*	.871	.891*	.941	.874	.824	.884
BAG-DT	PLT	.873	.931	.877	.920	.752	.885	.863	.884	.944	.865	.912*	.882
RF	ISO	.865	.934	.851	.935	.767*	**.920**	.877	.876	.933	.897*	.821	.880
BAG-DT	ISO	.867	.933	.840	.915	.749	.897	.856	.884	.940	.859	.907*	.877
SVM	PLT	.765	.886	.936	.962	.733	.866	.913*	.816	.897	.900*	.807	.862
ANN	–	.764	.884	.913	.901	.791*	.881	.932*	.859	.923	.667	.882	.854
SVM	ISO	.758	.882	.899	.954	.693*	.878	.907	.827	.897	.900*	.778	.852
ANN	PLT	.766	.872	.898	.894	.775	.871	.929*	.846	.919	.665	.871	.846
ANN	ISO	.767	.882	.821	.891	.785*	.895	.926*	.841	.915	.672	.862	.842
BST-DT	–	.874	.842	.875	.913	.523	.807	.860	.785	.933	.835	.858	.828
KNN	PLT	.819	.785	.920	.937	.626	.777	.803	.844	.827	.774	.855	.815
KNN	–	.807	.780	.912	.936	.598	.800	.801	.853	.827	.748	.852	.810
KNN	ISO	.814	.784	.879	.935	.633	.791	.794	.832	.824	.777	.833	.809
BST-STMP	PLT	.644	**.949**	.767	.688	.723	.806	.800	.862	.923	.622	.915*	.791
SVM	–	.696	.819	.731	.860	.600	.859	.788	.776	.833	.864	.763	.781
BST-STMP	ISO	.639	.941	.700	.681	.711	.807	.793	.862	.912	.632	.902*	.780
BST-STMP	–	.605	.865	.540	.615	.624	.779	.683	.799	.817	.581	.906*	.710
DT	ISO	.671	.869	.729	.760	.424	.777	.622	.815	.832	.415	.884	.709
DT	–	.652	.872	.723	.763	.449	.769	.609	.829	.831	.389	.899*	.708
DT	PLT	.661	.863	.734	.756	.416	.779	.607	.822	.826	.407	.890*	.706
LR	–	.625	.886	.195	.448	.777*	.852	.675	.849	.838	.647	.905*	.700
LR	ISO	.616	.881	.229	.440	.763*	.834	.659	.827	.833	.636	.889*	.692
LR	PLT	.610	.870	.185	.446	.738	.835	.667	.823	.832	.633	.895	.685
NB	ISO	.574	.904	.674	.557	.709	.724	.205	.687	.758	.633	.770	.654
NB	PLT	.572	.892	.648	.561	.694	.732	.213	.690	.755	.632	.756	.650
NB	–	.552	.843	.534	.556	.011	.714	-.654	.655	.759	.636	.688	.481

Caruana and Niculescu-Mizil (2006)

Considerations when choosing among methods

- Supervised or unsupervised task?
- Is the outcome continuous or discrete?
- What is your goal? (Prediction or insight?)
- How well does the model match the data generating process?
- Likelihood-based or algorithmic method?
- How big is n? How much flexibility is needed?

Supervised or unsupervised task?

- In a supervised learning task, we are given training data examples $(x_1, y_1), \ldots, (x_n, y_n)$, and we construct a function $\hat{f}(x)$ for predicting future values of y given x.
  - Regression
  - Classification

- In an unsupervised learning task, we are given training data examples $x_1, \ldots, x_n$, and we compute some summaries such as cluster assignments, a low-dimensional projection, or parameters of the probability distribution of the x's.
  - Dimension reduction (e.g., PCA, ICA, etc.)
  - Clustering

Is the outcome continuous or discrete?

- Regression (continuous outcomes): Linear regression, lasso, elastic net, smoothing splines, KNN, support vector regression, regression trees.
- Classification (discrete outcomes): Logistic regression, LDA, QDA, KNN, support vector machines, classification trees.
- GLMs such as Poisson regression and Negative Binomial regression can handle discrete outcomes $y \in \{0, 1, 2, \ldots\}$.

What is your goal? Prediction versus insight

Prediction

- Sometimes, all we care about is making an accurate prediction of y given x.
- Examples: predicting disease risk, detecting disease, predicting survival.
- In this case, the prediction function $\hat{f}(x)$ can be treated as a "black box" that takes an input x and produces a prediction y, without giving any insight into why or how the prediction was made.
- Example methods: KNN, random forests, SVMs, smoothing splines, Gaussian processes, neural networks — generally speaking, flexible/nonparametric methods.
- More flexible methods tend to be less interpretable.

What is your goal? Prediction versus insight

Insight / understanding

- Sometimes, we are more interested in understanding the relationship between x and y. Typically, this involves inference for some parameters.
- Examples: causal inference, inferring biological mechanisms, genetic disease variants, finding biomarkers.
- In this case, interpretability is key. For example, which variables in x are important? What is the relationship between these variables and y?
- Example methods: Linear regression, logistic regression, GLMs, lasso, elastic net, Bayesian models — generally speaking, parametric or model-based methods.
- More interpretable methods tend to be less flexible.

How well does the model match the data generating process?

- Every method involves assumptions about the distribution of the data, a.k.a. the data generating process.

- *Likelihood-based methods* are based on a probabilistic model for the data.
  - Assumptions are explicit $\implies$ Tend to be more interpretable
- *Algorithmic methods* directly specify an algorithm or an objective function to optimize.
  - Assumptions are implicit $\implies$ Tend to be less interpretable

- Even the simplest method will perform optimally if its assumptions perfectly match the data generating process.
- But if little is known about the data generating process, then a more flexible method may be preferable.

Likelihood-based versus algorithmic method?

Likelihood-based methods

- Examples: linear regression, logistic regression, GLMs, Bayesian models, Probabilistic PCA, mixture models.
- Advantages:
  - Interpretability: model parameters and latent variables correspond directly to quantities of interest.
  - Complex dependency structures can easily be defined using hierarchical generative models.
  - Uncertainty quantification is usually straightforward.
  - Performance can be improved by exploiting domain knowledge when building the model.
  - Correctness and optimality guarantees hold under general conditions, provided that the model is correct.
- Disadvantages:
  - More complex probabilistic models tend to be more computationally intensive.
  - Simpler probabilistic models tend to be less flexible.

Likelihood-based versus algorithmic method?

Algorithmic methods

- Examples: CART, random forests, neural networks, SVMs, ensembles, hierarchical clustering.
- Advantages:
  - Computationally fast, in many cases.
  - Simpler to implement, usually, relative to comparable likelihood-based methods.
  - Certain algorithms exhibit excellent performance in practice.
- Disadvantages:
  - Less interpretable. Post hoc analysis is often required to get insight into what the black box is doing.
  - Establishing correctness and optimality properties requires greater theoretical effort.
  - Uncertainty quantification is often difficult, requiring bootstrapping or similar techniques.

Likelihood-based versus algorithmic method?

Methods in both camps

- Some methods such as lasso and elastic net are kind of in-between.
- These procedures are defined algorithmically by optimizing an objective function.
- However, the objective function can also be viewed as arising from a particular probabilistic model.

How big is n? How much flexibility is needed?

- Statistical and computational concerns both depend on:
  - the sample size n, and
  - the flexibility of the model (e.g., number of parameters).

- Computational concerns
  - Obviously, computation time will grow with n.
  - Computational complexity (how fast it grows) is important.

- Statistical concerns
  - Overfitting or underfitting can occur if the flexibility of the method is not matched appropriately to the dataset.
  - Most methods have knobs that control flexibility.

    e.g., number of predictor variables to use, regularization parameter, number of neighbors in KNN, tree depth, Bayesian prior, number of clusters.
  - How to set these knobs? Stay tuned!

References

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 161–168. ACM, 2006.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer Series in Statistics, New York, 2009.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.