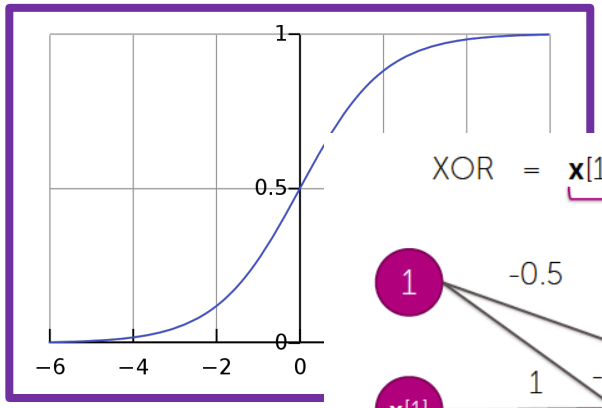
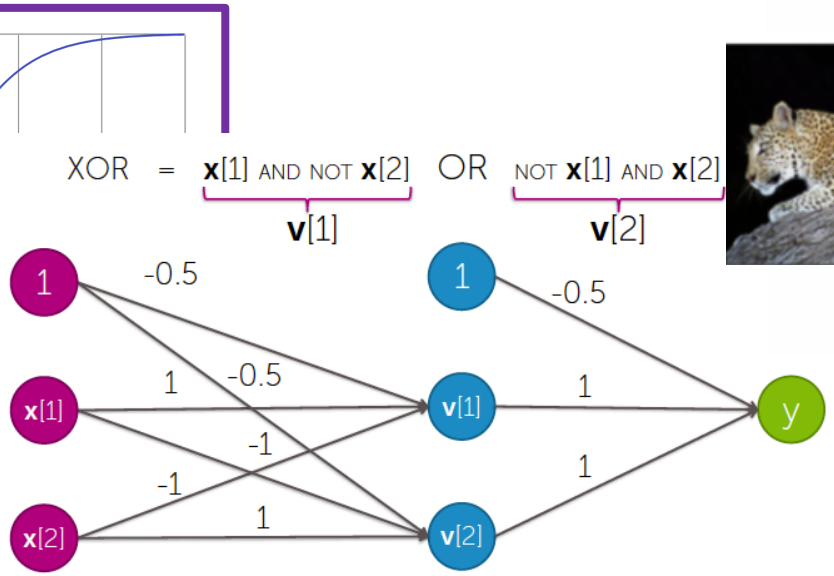
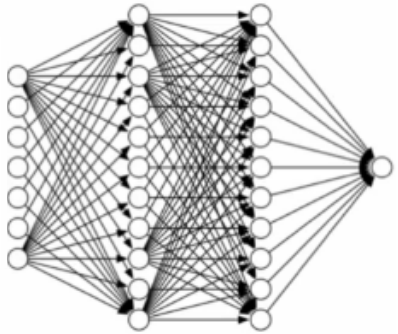
Tufts COMP 135: Introduction to Machine Learning

https://www.cs.tufts.edu/comp/135/2020f/

Final Review







Many slides attributable to:
Erik Sudderth (UCI), Emily Fox (UW),
Finale Doshi-Velez (Harvard)
James, Witten, Hastie, Tibshirani (ISL/ESL books)

Prof. Mike Hughes

Semester In Review

Unit 1: Regression

Unit 2: Classifiers

Unit 3: Neural Nets

Unit 4: Trees & Ensembles

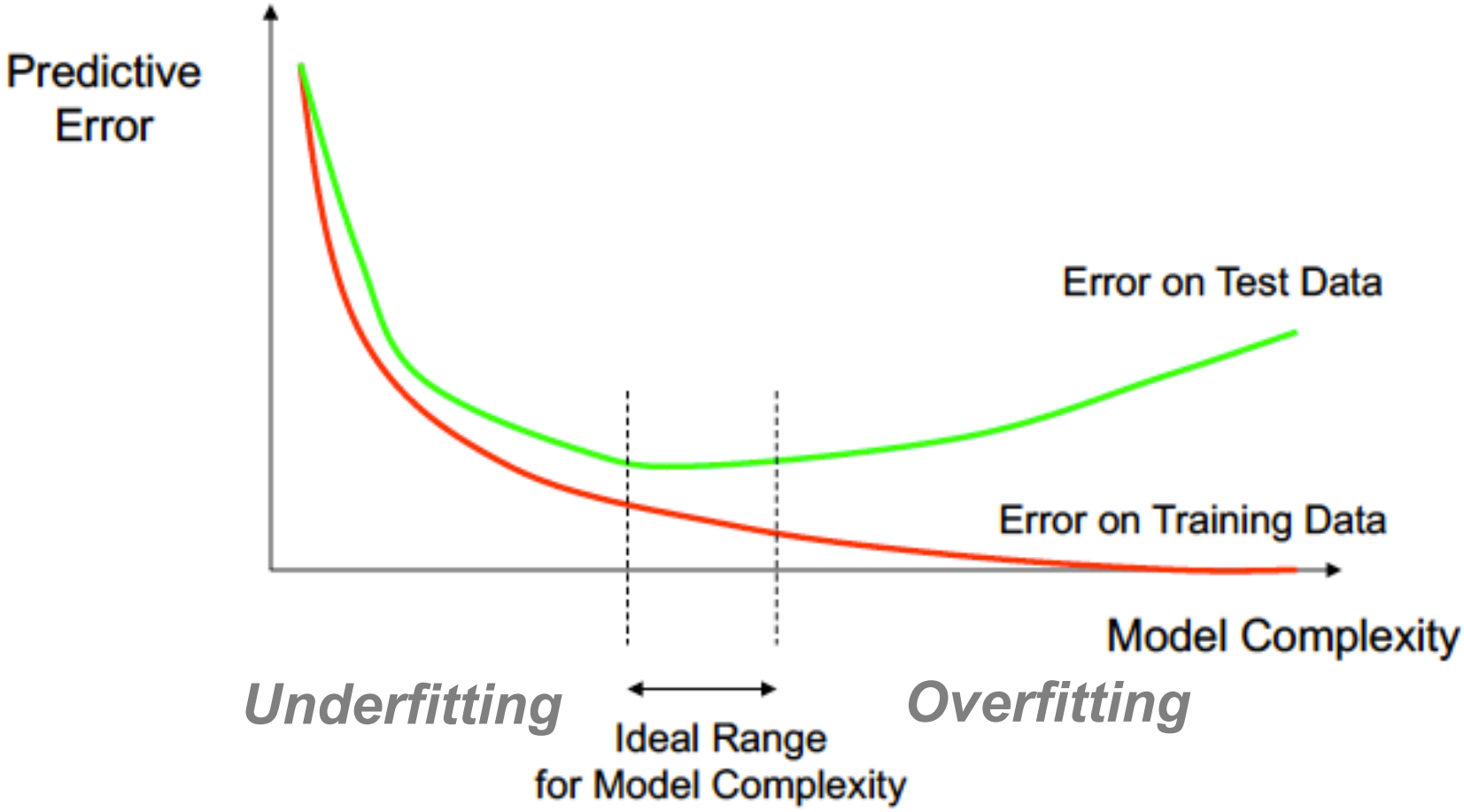
Unit 5: SVMs & Kernels

Unit 6: PCA & Recommendations

Supervised

Unsupervised

Model Complexity vs Error



Managing this tradeoff is the fundamental problem of machine learning

Supervised Learning: Methods

PARAMETRIC

- Linear Methods
  - Linear regression / Logistic regression
  - SVM (linear kernel)
- Neural Networks

NON-PARAMETRIC

- K-Nearest Neighbors
- Decision Trees
- Random Forests
- Kernels like RBF

What to know for each method

- Can it do regression? Binary classification? Multi-class?
- What are the parameters (learned from training)?
- What are hyperparameters (control complexity)?
  - What ranges of values are possible?
  - What limiting cases exist (e.g. K=training set size)?
- What is the prediction algorithm?
  - Runtime complexity (how depend on train size? feature size?)
- What are possible training algorithms?
  - Will it always give a useful answer?
  - Runtime complexity (roughly)
- How flexible are decision boundaries?
- What are possible input features?
  - Binary, Categorical, Numeric

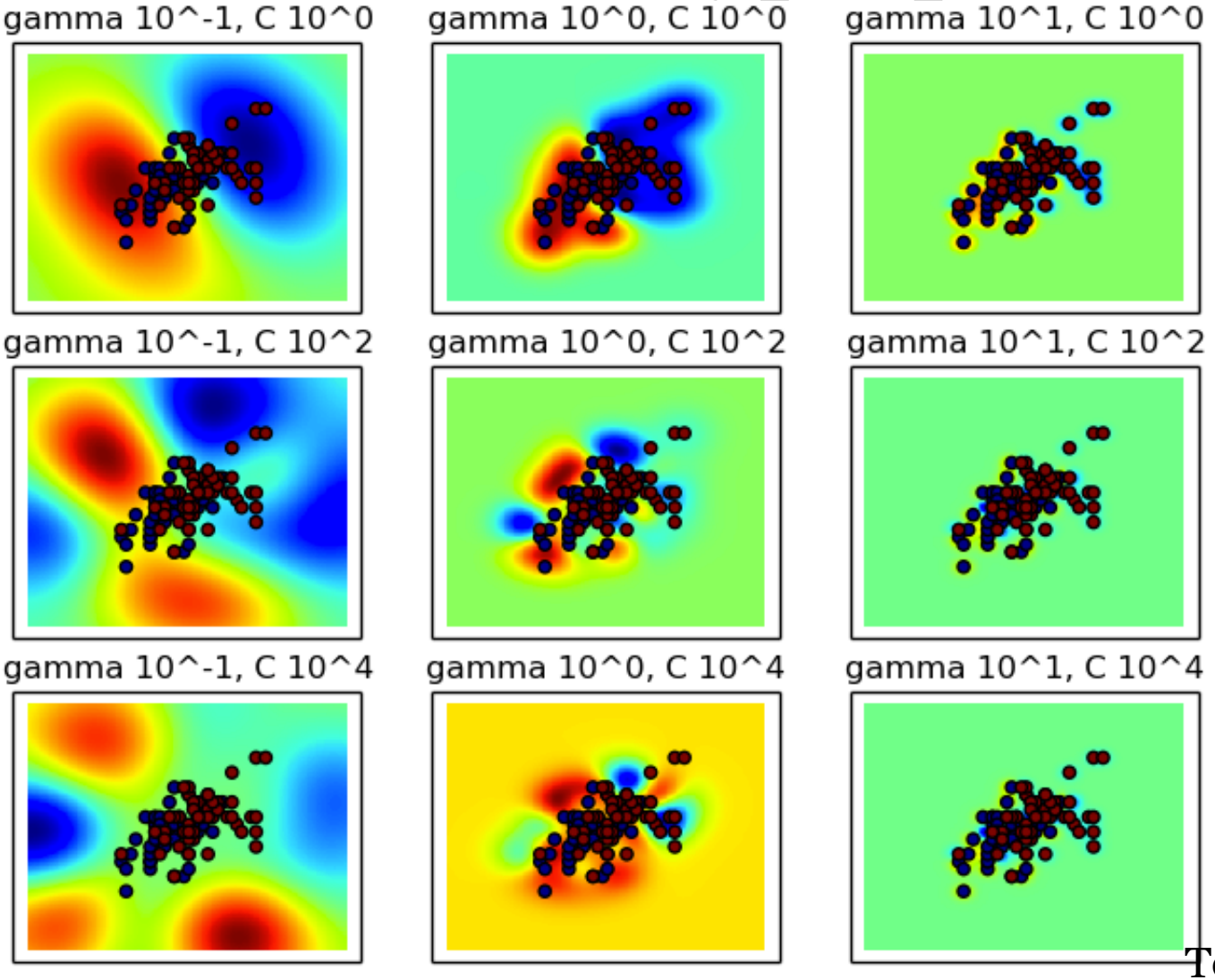
Summary of Methods

	Function class flexibility of decision boundary	Hyperparameters that control complexity	Other knobs to tune
Logistic Regression Classifier	Linear	Strength of L2 penalty on weights Strength of L1 penalty on weights	Include bias? Tolerance?
Decision Tree Classifier	Axis-aligned Piecewise constant	Max. depth Min. leaf size	Which cost function (gini vs. entropy)
K Nearest Neighbors Classifier	Piecewise constant	Number of Neighbors	Distance function?

Summary of Methods

	Function class flexibility of decision boundary	Hyperparameters that control complexity	Other knobs to tune
Random Forest Classifier	Average of many decision trees (average of axis-aligned piecewise constants)	• Min leaf size • Max depth • Number of trees (cannot overfit this!)	How to randomly sample features or examples
Linear SVM Classifier	Linear	C : scalar weight on error term	Tolerance?
RBF kernel SVM Classifier	Most flexible	C : scalar weight on error term Gamma : lengthscale	Pick another kernel?

SVM with RBF hyperparameters



https://scikit-learn.org/0.15/auto_examples/svm/plot_rbf_parameters.html

Toward zero training error

Limitations of ML

Claim:
Deep Learning is better for everything

Status

- DL is good for high-dimensional data with complex patterns (images, waveforms, text, etc.) where learned representation is key
- May not see benefits on small "spreadsheet"-like datasets with monotonic relationships between features and labels (e.g. age and mortality).

Further reading

Journal of Clinical Epidemiology 110 (2019) 12–22

Journal of Clinical Epidemiology

REVIEW

A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models

Evangelia Christodoulou[a], Jie Ma[b], Gary S. Collins[b,c], Ewout W. Steyerberg[d], Jan Y. Verbakel[a,e,f], Ben Van Calster[a,d,*]

Claim:
ML/DL is “black box”, not explainable

Status

- Explainable AI is active research area (thanks to DARPA & others)
- Important to define the “interpretability” needs for your specific task
- Current methods best for tabular data; less work on time series

Remedies

- **Surrogate models** mimic decisions of fancier ML (often in confined region of input space) with rule lists / decision trees
- **White box models** (see SLIM by Ustun & Rudin, graphical models)

Further reading

Zach Lipton. “The Mythos of Model Interpretability”. 2016. https://arxiv.org/pdf/1606.03490.pdf

Claim:
ML is correlation not causality

Status

- Yes, most off-the-shelf ML/DL methods are associational, will be "fooled" by correlations (increase ice cream sales predict increased drownings in public pools)
- Can be overcome if right formal model & assumptions are known and used from the beginning (but this is hard!)

Active Research

- Many efforts in online advertising: integrate ML with randomized experiments (A/B tests)
- Many efforts in healthcare: use ML to combine randomized clinical trials and observational health records datasets

Further Reading

Judea Pearl. The Seven Tools of Causal Inference, with Reflections on Machine Learning. Communications of the ACM (2019).

Claim:
ML/DL is not “fair”

Status

- Many deployed systems do not meet various fairness definitions
- Not possible for one system to meet all fairness definitions
- Important to define what “fairness” means for your specific task

Remedies

- **Involve stakeholders**
- **Do not just blindly “exclude race”**
- **Require careful audits before high-stakes deployment**

Takeaways

Goals of this course

Our goal is to prepare students to effectively apply machine learning methods to problems that might arise in "the real world" -- in industry, medicine, government, education, and beyond.

Gain **skills** and ***understanding*** for a future as:

- Developer using ML “out-of-the-box”
- ML methods researcher

After taking this course, you will be able to:

- *Think systematically and ethically*
  - Compare/contrast each method's strengths & limitations
  - "Can ML solve this problem?"
  - "***Should*** ML solve this problem?"

- *Deploy and debug rapidly on real problems*
  - Hands-on experience with open-source libraries
  - Address issues in "real-world" data analysis
    - Numerical issues, convergence issues, class imbalance, missing values, etc.

- *Evaluate carefully and honestly*
  - Design experiments to assess generalization to never-before-seen data
  - Select task-appropriate performance metrics
  - Report confidence or uncertainty in performance numbers

- *Communicate insightfully and reproducibly*
  - Surface key insights via figures, tables, and text in a written report
  - Provide details for a peer to repeat your analysis and draw same conclusions

Takeaway 1:

Never train on the test dataset

Takeaway 2

Spend **more time thinking about data preparation and evaluation** than you do fitting models / tuning hyperparameters

- Split data as you will in deployment
  - Split by subject
  - Split by time

  Split by spatial region/site

- Choose the right performance metric
  - Accuracy rarely the right metric in real applications
  - Use the target metric to train and to select hyperparameters

Takeaway 3

Make plots to gain insight

- Train vs validation errors as algorithm learns…

- Inspect false positive vs false negatives

Open Q & A

TOPIC 1: Ideas in ML

TOPIC 2: How to keep learning / get a job