A Few Useful Things to Know About Machine Learning

Dr. Fayyaz ul Amir Afsar Minhas

PIEAS Biomedical Informatics Research Lab

Department of Computer and Information Sciences

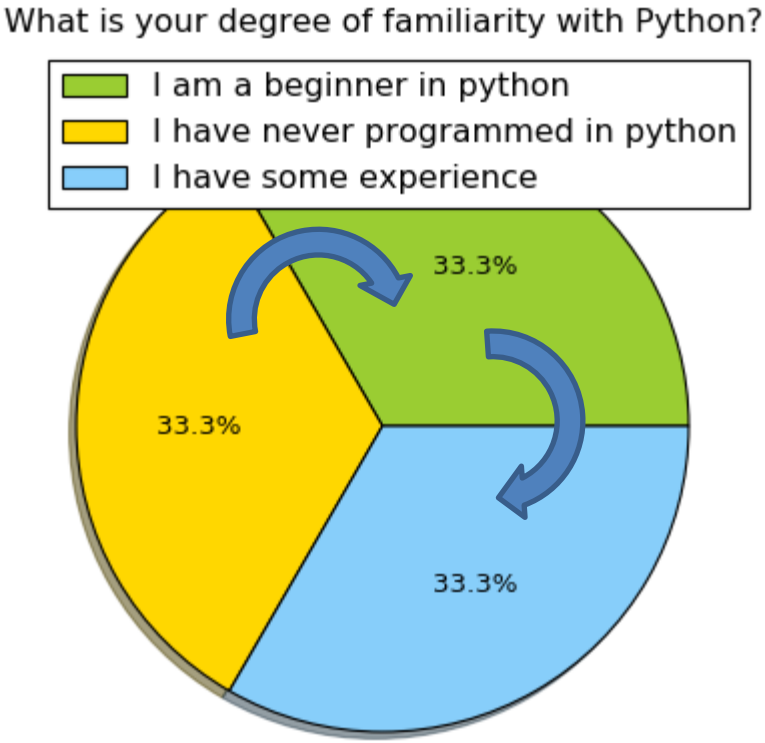
Pakistan Institute of Engineering & Applied Sciences

PO Nilore, Islamabad, Pakistan

http://faculty.pieas.edu.pk/fayyaz/

Survey Results

- Python Experience
- Topics
  - Will be covered in depth
    - SVM (Kernels, Structural Risk Minimization)
    - Regression
    - Recommender Systems
    - Experiment Design: Model Selection
    - Feature Selection
  - Not in depth (Self Reading and Assignments)
    - Perceptron and Neural Networks
    - Random Forests and Ensembles
    - ROC and PR curves
    - PCA and LDA
    - Curse of Dimensionality
  - Will cover if time allows
    - Reinforcement Learning
  - Not covered
    - Clustering
- Projects
  - If you have any ideas of your own, do let me know by Next Monday



Assignment 1: Part (B)

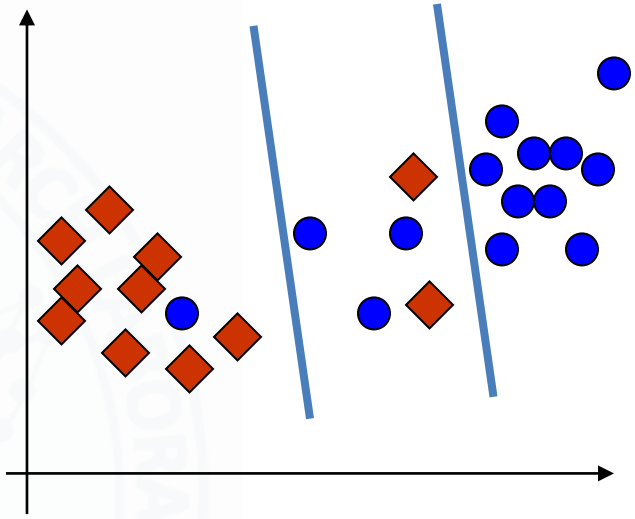
- Short Questions
  - What is the percentage error of the Nearest Neighbor classifier on **training** data? Why?
  - What is the percentage error of the Nearest Neighbor classifier on **test** data?
    - Can you develop a hypothesis and mathematical proof of why the error is such? [BONUS]
  - What is the **Space** Complexity of the Nearest Neighbor Classifier for testing a single example in terms of the number of training examples and the feature dimension?
  - What is the **Time** Complexity of the Nearest Neighbor Classifier for testing a single example in terms of the number of training examples and the feature dimension?
    - Plot the average time taken for testing 1000 examples with different number of training examples (0.1K, 1K, 10K, 100K…) and feature dimensions (2, 4, 8, 16, 32, …). Does your time complexity follow this curve?
  - What will be the effect of the number of dimensions on the error rate?
    - Plot the error rate vs. number of dimensions for d = (2,3,4…, 50). Does the plot agree with your intuition? Can you explain this behavior?
    - Can you develop a hypothesis that can explain this behavior? Also devise and conduct a computational experiment to prove this hypothesis.
- Make a linear classifier
  - $z = \text{sgn}(w_1x_1 + w_2x_2 + b)$, Implement a function which returns labels of the test data much like "NN"
  - Manually choose w_1 and w_2 using training data to minimize error over training data only
  - Report the test error and how you chose the line?
  - What is the **Space** Complexity of the Linear Classifier for testing a single example in terms of the number of training examples and the feature dimension?
  - What is the **Time** Complexity of the Linear Classifier for testing a single example in terms of the number of training examples and the feature dimension?
- Submission Deadline: Thursday, Feb. 4

Today's Topic

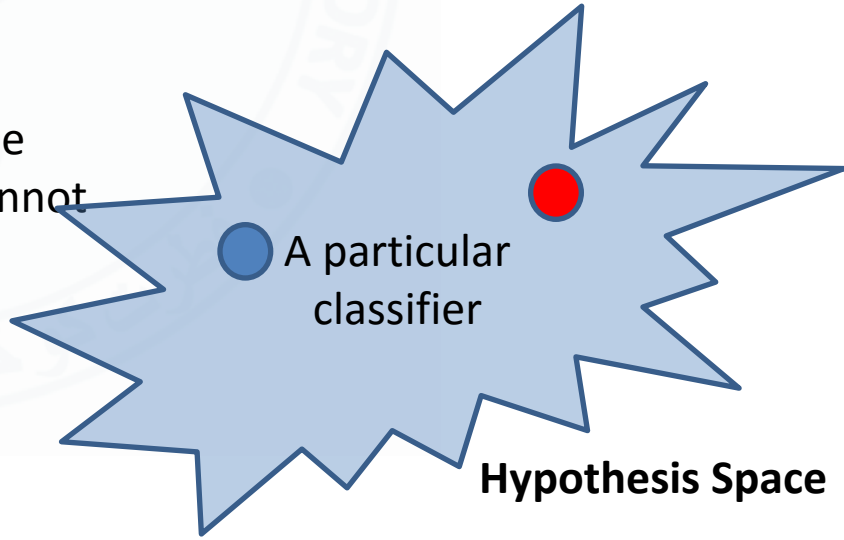
- Domingos, Pedro. "A Few Useful Things to Know About Machine Learning." *Commun. ACM* 55, no. 10 (October 2012): 78–87. doi:10.1145/2347736.2347755.

Learning

- Representation
  - How is the classifier represented?
    - What is the rule for Nearest Neighbor and Linear Classifiers?
  - What features are used?
  - If a classifier cannot represent the concept in a problem, it will not be able to solve it!
    - Hypothesis space
      - Space of possible classifiers
        - » Parameters of the classifier
    - If the solution to a problem lies outside the hypothesis space being considered, we cannot solve the problem!

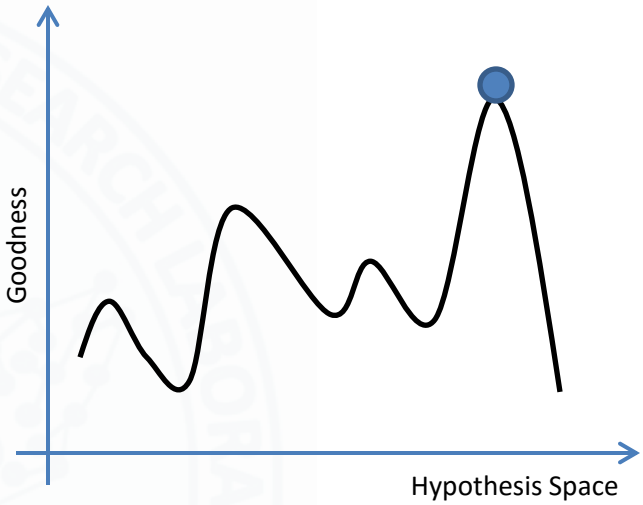


$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$



Learning

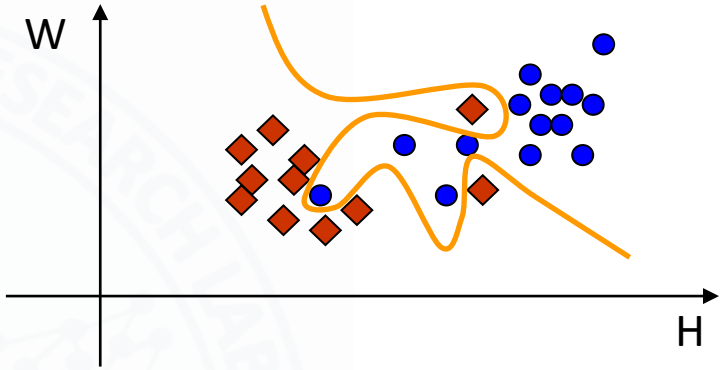
- Evaluation
  - Distinguishing good classifiers from bad ones
    - What is a good linear classifier?
    - How to define good and bad?
      - Training Error
- Optimization
  - Search for the good classifier
    - How do we find a good linear classifier?



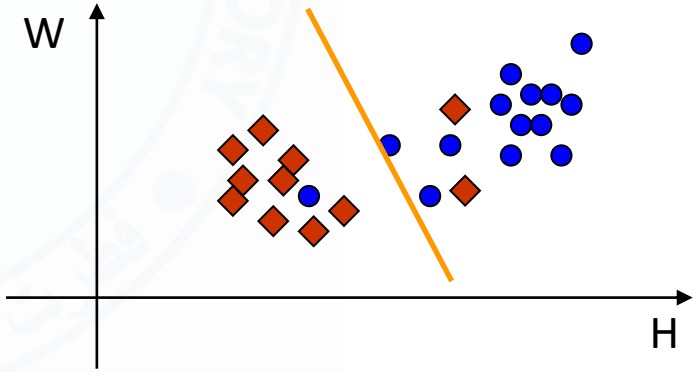
Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Generalization

- How can we make a classifier that does a 100% on training data?
  - Memorization
  - How will it work on unseen examples?
- How to generalize well?
  - Don't use the test data in training to tune the parameters and don't do a lot of tuning.
    - Example: Choosing a k in kNN that maximizes the score on known test examples
- Issue
  - We want to get good performance on test data but we don't have access to it so
    - Use Training Data
    - Bias!



Has great memorization but may generalize poorly



Has lesser memorization but may generalize better

Data Alone is Not Enough!

- Assume you have 100 binary variables and you are given a million training examples
  - What is the possible number of ways to label 100 binary variables?
    - 2^{100}
  - No rule can learn without embodying some knowledge or assumptions beyond the data it's given in order to generalize beyond it.
  - Example
    - The data is smooth
      - Similar examples have similar labels
    - Limited Complexity
      - The classification problem is not too hard
    - Data dimensions are redundant

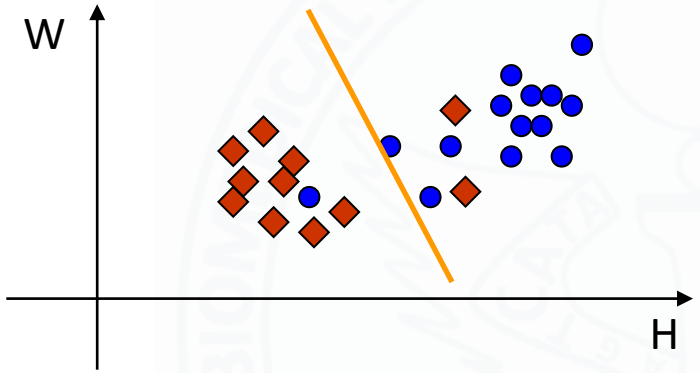
Data Alone is Not Enough!

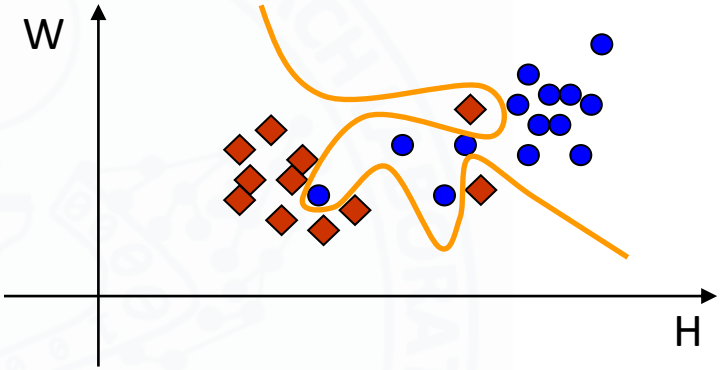
- No Free Lunch
  - No learner can beat random guessing over all possible functions to be learned

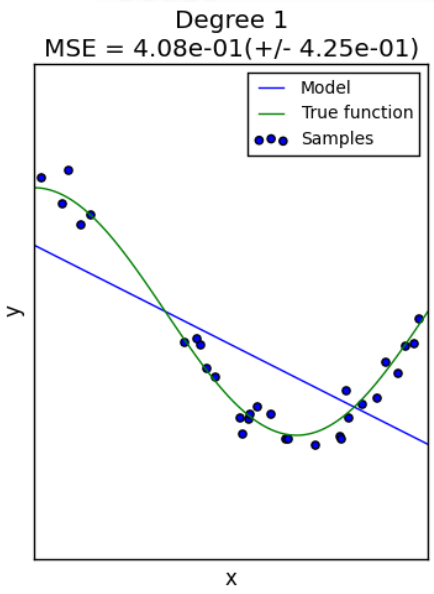
- Choice of the classifier is dependent upon the data and the assumptions we should make
  - If we know what makes our examples similar, we can use kNN
  - If we know what kinds of preconditions are required by each class, then we can me "IF… THEN …" rules
- A good classifier allows changing its assumptions based on the data so that it can have a big hypothesis space

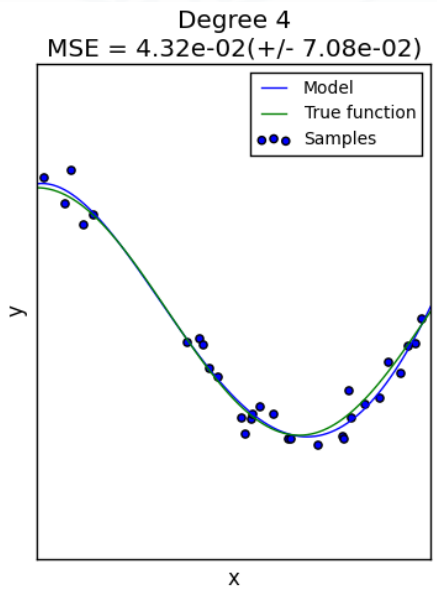
Over-fitting

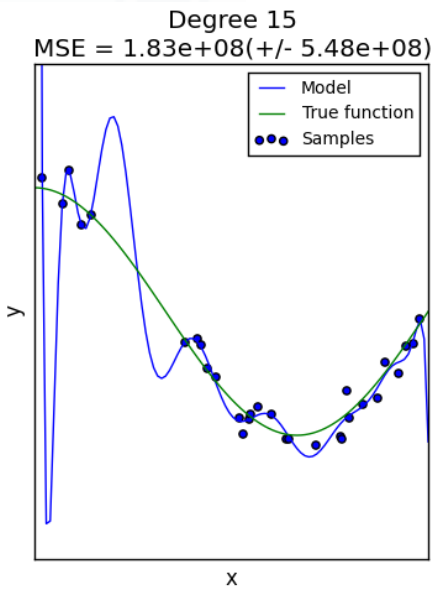
- Over-fitting is to be avoided



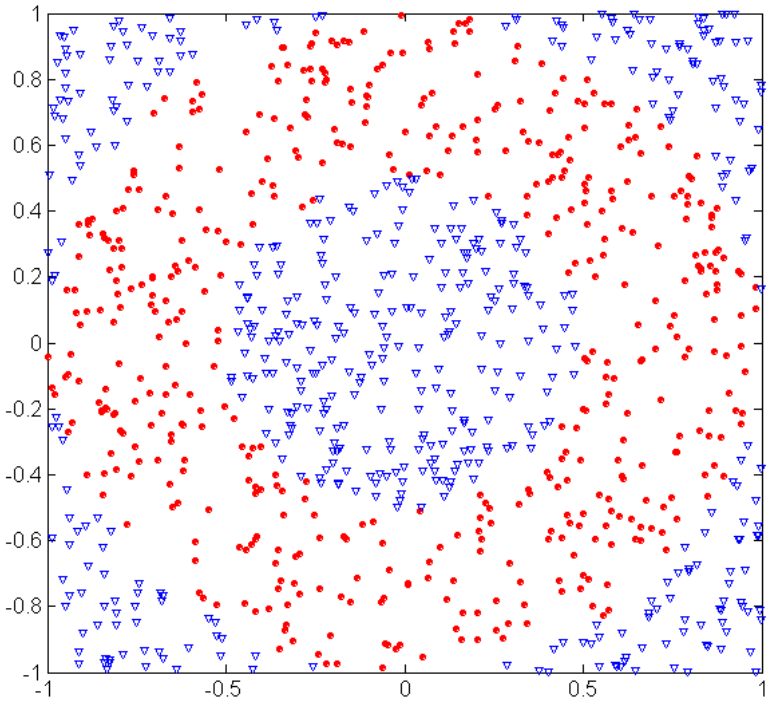


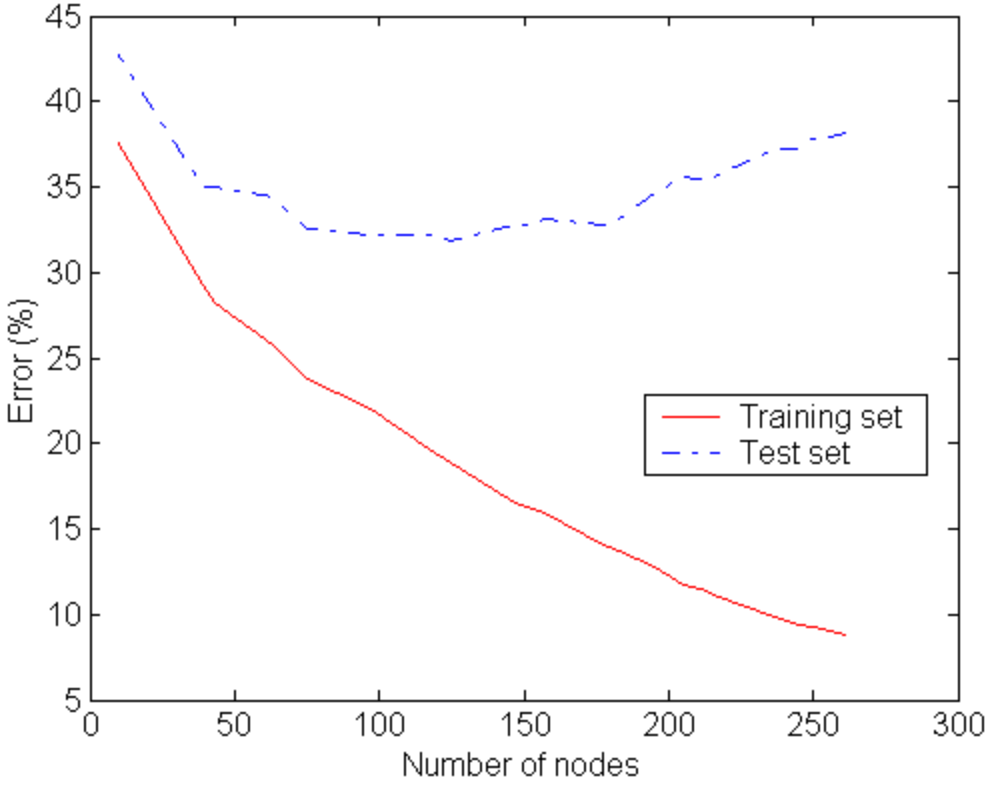






Overfitting in Tree Classifiers





Causes of Over-fitting

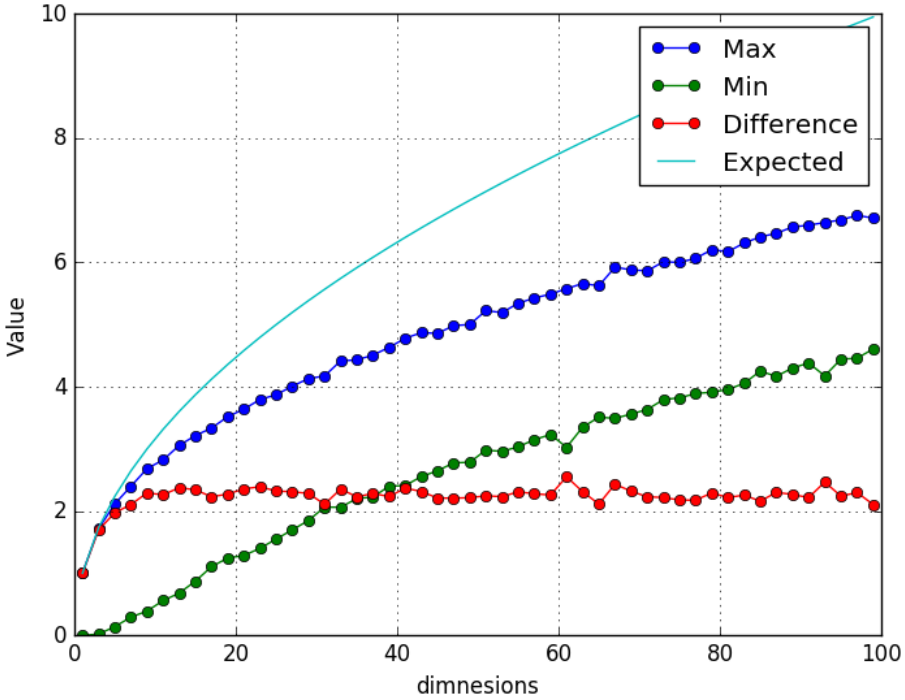
- Noise
- Insufficient Number of Examples
- Overly Complex Learner
- Bad evaluation protocols
- Over-tuning
- Limiting the Capacity of the classifier
  - SVMs can have an infinite capacity without overfitting!
- Use Cross-Validation Strategies

Intuition Fails in High Dimensions

- Let's assume N d-dimensional points generated uniformly randomly in the range $[0,1]^d$
- Let's take their distance from a fixed arbitrary query point (say the origin)
- Let's use the Euclidean Distance (l = 2)
  - $d(\boldsymbol{x}, \mathbf{0}) = \left[\sum_{i=1}^{d}(x_i - 0)^l\ \right]^{\frac{1}{l}}$
- Let's find the minimum and maximum distance across the whole set
  - The minimum distance should be:
    - Zero
  - The maximum distance should be:
    - $\left[\sum_{i=1}^{d}(x_i - 0)^l\ \right]^{\frac{1}{l}} = \left[\sum_{i=1}^{d}(1 - 0)^l\ \right]^{\frac{1}{l}} = d^{\frac{1}{l}}$
  - Their difference is called the "contrast"
- Let's compute it

Intuition Fails in High Dimensions

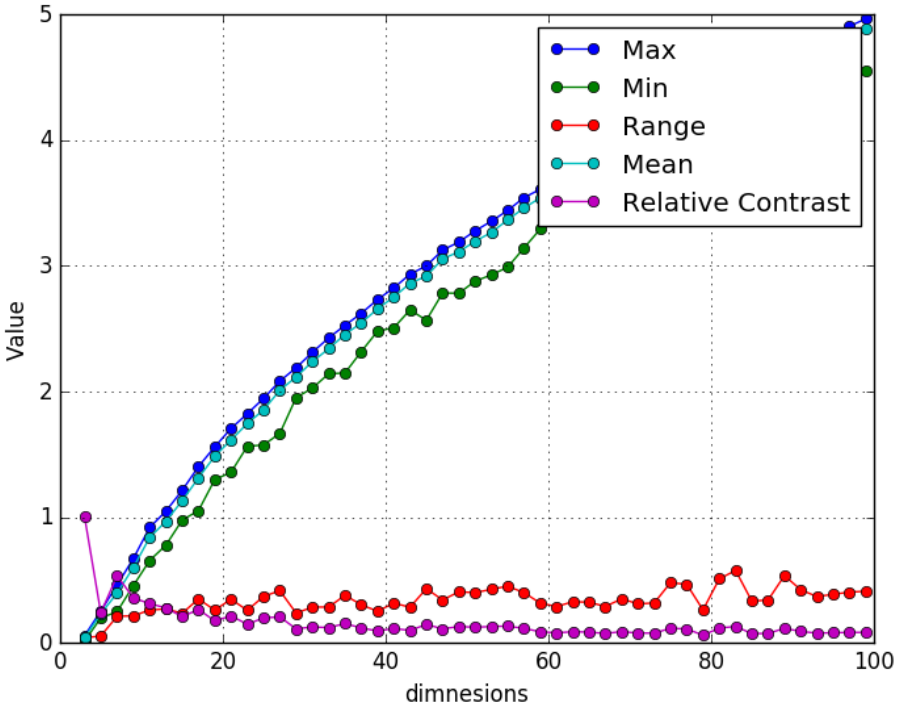
- This shows that the contrast does not increase as expected
- Why do we need contrast?
  - So we can see which points are closer to a given point and which are farther away
  - But this reduction in contrast means that we may NOT be able to classify correctly in higher dimensions using the Euclidean distance!



Aggarwal, Charu C., Alexander Hinneburg, and Daniel A. Keim. “On the Surprising Behavior of Distance Metrics in High Dimensional Space.” In *Database Theory — ICDT 2001*, edited by Jan Van den Bussche and Victor Vianu, 420–34. Lecture Notes in Computer Science 1973. Springer Berlin Heidelberg, 2001. http://link.springer.com/chapter/10.1007/3-540-44503-X_27.

Intuition Fails in High Dimensions

- Take the distance of a point (say, the origin) from all N points and see the values of the distances for the nearest k = 2d neighbors.
  - Relatively speaking, the closest one is as far away as the farthest one!



Feature Engineering is the key

- Little time is spent on machine learning
- Most time
  - Gathering Data
  - Pre-processing it
  - Feature Extraction

- Good classifiers allow you to reflect your understanding of the data in them
- Automation of feature engineering
  - Generate large number of features, then select
  - Deep Learning
- No replacement for the smarts you put into feature engineering

https://en.wikipedia.org/wiki/Feature_learning

https://en.wikipedia.org/wiki/Feature_engineering

More data beats a cleverer algorithm

- *"We saw dramatic improvement when moving from ten thousand to two million images."*
- Hays, James, and Alexei A. Efros. "Scene Completion Using Millions of Photographs." In *ACM SIGGRAPH 2007 Papers*. SIGGRAPH '07. New York, NY, USA: ACM, 2007. doi:10.1145/1275808.1276382.
- Halevy, Alon, Peter Norvig, and Fernando Pereira. "The Unreasonable Effectiveness of Data." *IEEE Intelligent Systems*, 2009.

Scene Completion Using Millions of Photographs

James Hays Alexei A. Efros
Carnegie Mellon University

Figure 1: Given an input image with a missing region, we use matching scenes from a large collection of photographs to complete the image.

- Try a simple model first before moving on to more complicated ones
- Complex classifiers are seductive

http://data-informed.com/why-more-data-and-simple-algorithms-beat-complex-analytics-models/

https://www.quora.com/In-machine-learning-is-more-data-always-better-than-better-algorithms

Other Ideas

- Use Ensembles
- Simplicity may not (always) imply accuracy
  - There is no necessary connection between the number of parameters of a model and its tendency to overfit.
  - Occam's Razo
    - KISS Principle
- Representable Does Not Imply Learnable
  - Multilayer Perceptrons and Decision Trees can, in theory, approximate any funtion!
  - However that does not mean it can solve any problem through learning!
- Correlation does not imply causation!

TO DO

- Complete the assignment
- Read the paper

- Quiz Next Lecture

End of Lecture-3

We want to make a machine that will be proud of us.

- Danny Hillis