Lecture 7: Intro to ML & MLlib

Professor Robert J. Brunner
Quinn Jarrell

Lab 4

- How was lab?

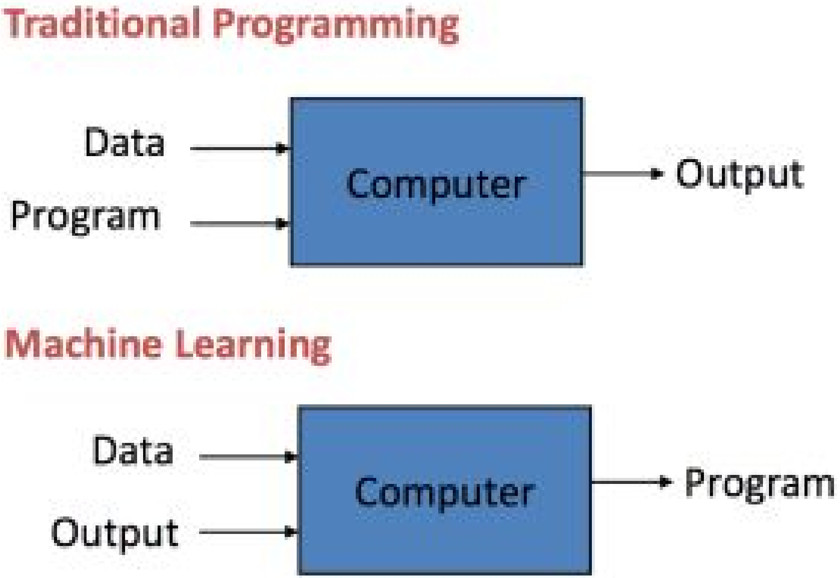
Announcement

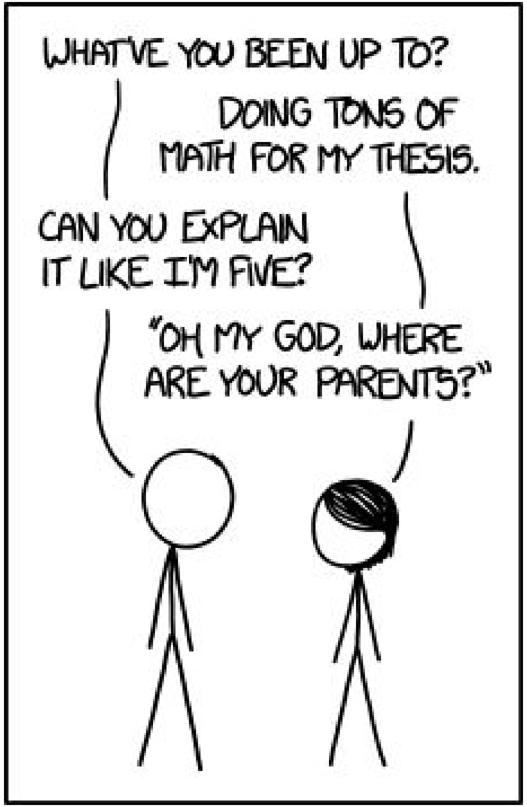
Our BlueWaters (National Petascale Computing Facility) Tour is on March 31st at 4pm

More logistics details coming soon.

Machine Learning

ML is about generalizing things :D





ML is actually really Math heavy!

From CS446 SP17 HW1:

The Learning Problem:[3] Let $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}$ represent m samples, where each sample $\vec{x_i} \in \mathbb{R}^n$ is an n-dimensional vector, and $\vec{y} \in \{-1, 1\}^m$ is an $m \times 1$ vector representing the respective labels of each of the m samples. Let $\vec{w} \in \mathbb{R}^n$ be an $n \times 1$ vector representing the weights of the linear discriminant function, and θ be the threshold value.

We *predict* $\vec{x_i}$ to be a *positive* example if $\vec{w}^T \vec{x_i} + \theta \geq 0$. On the other hand, we *predict* $\vec{x_i}$ to be a *negative* example if $\vec{w}^T \vec{x_i} + \theta < 0$.

We hope that the learned linear function can separate the data set. That is, for the true labels we hope

$$y_i = \begin{cases} 1 & \text{if } \vec{w}^T \vec{x_i} + \theta \geq 0 \\ -1 & \text{if } \vec{w}^T \vec{x_i} + \theta < 0. \end{cases} \tag{1}$$

In order to find a good linear separator, we propose the following linear program:

$$\min_{\vec{w}, \theta, \delta} \quad \delta \tag{2}$$

$$\text{subject to} \quad y_i(\vec{w}^T \vec{x_i} + \theta) \geq 1 - \delta, \qquad \forall (\vec{x_i}, y_i) \in D \tag{3}$$

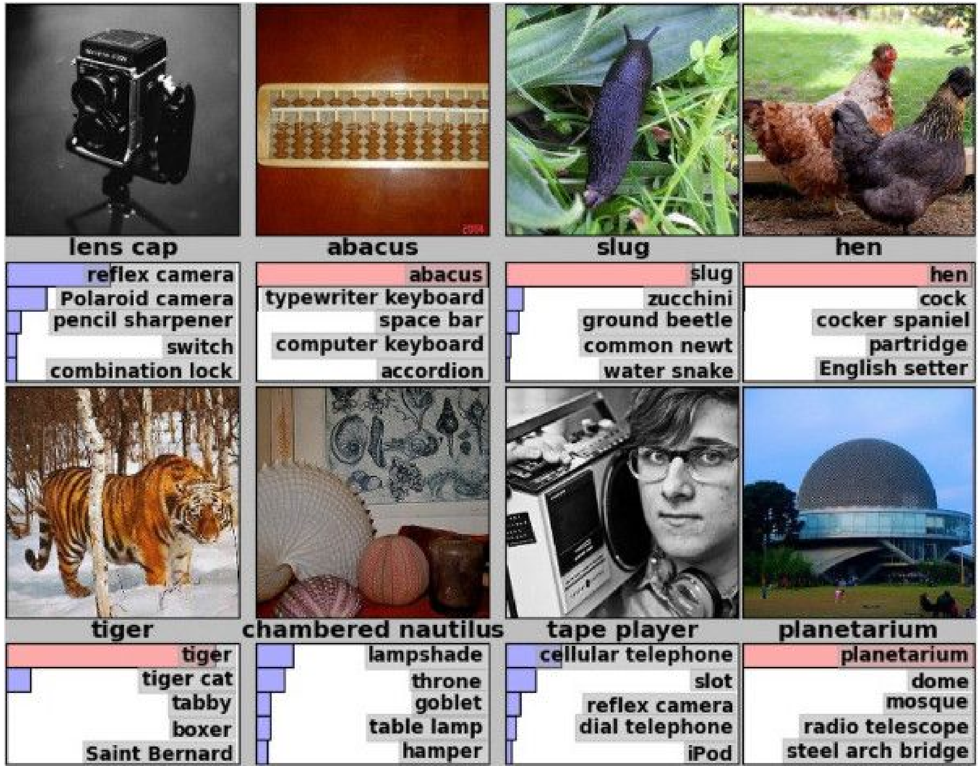
$$\delta \geq 0 \tag{4}$$

Types of Learning

- **Supervised**: Training data includes outputs
  - Classification (spam, etc.), Regression
- **Unsupervised**: Training data does not includes desired output
  - Clustering
- **Reinforcement**: Rewards from a sequence of actions.

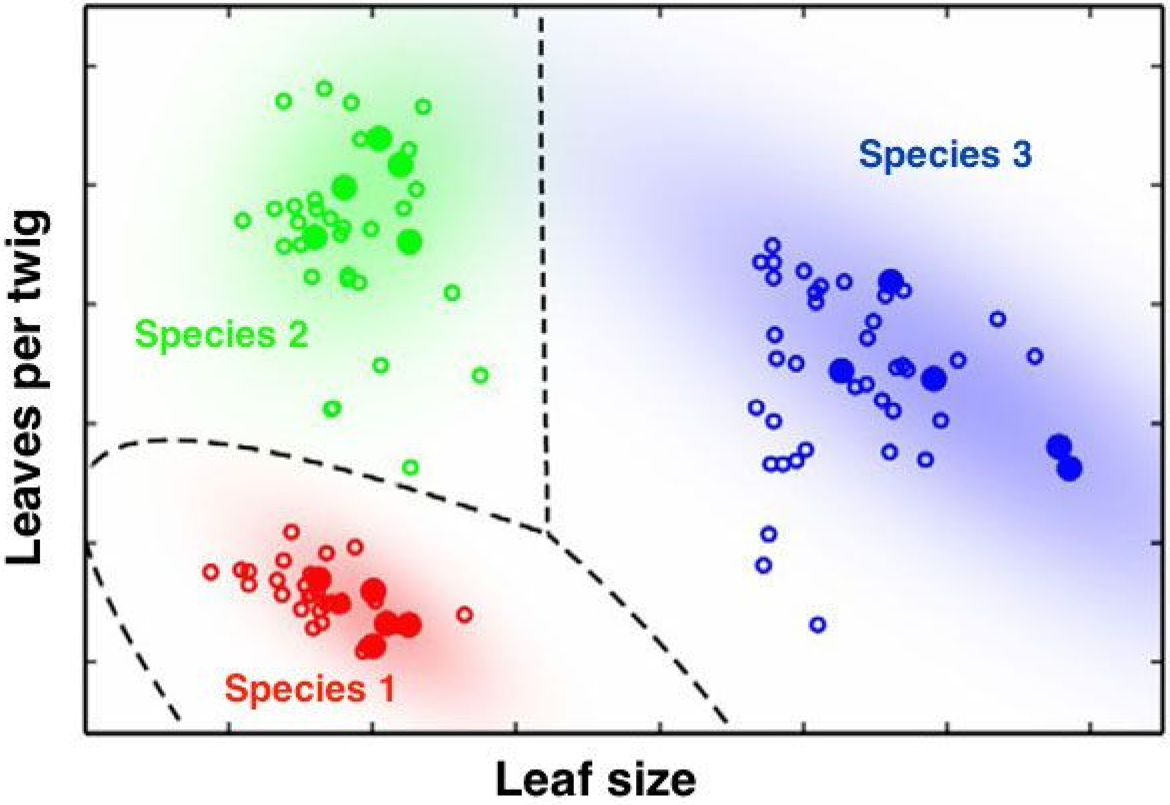
Classifiers

- Classifiers take in data and try to assign a label or labels to it



Classifiers

- Requires categories
- Requires labeled data

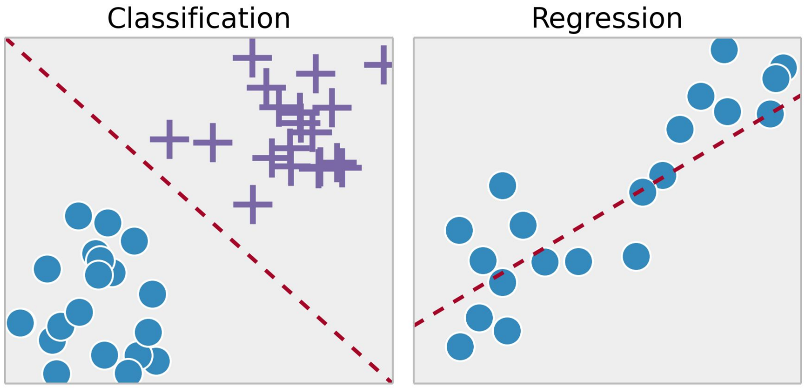


Regression

Regression VS Classification

- Regression derives a relationship between variables
    - Used to give a quantify for a new object
- Classification sorts things into categories
    - Works on discrete number of things, NOT continuous

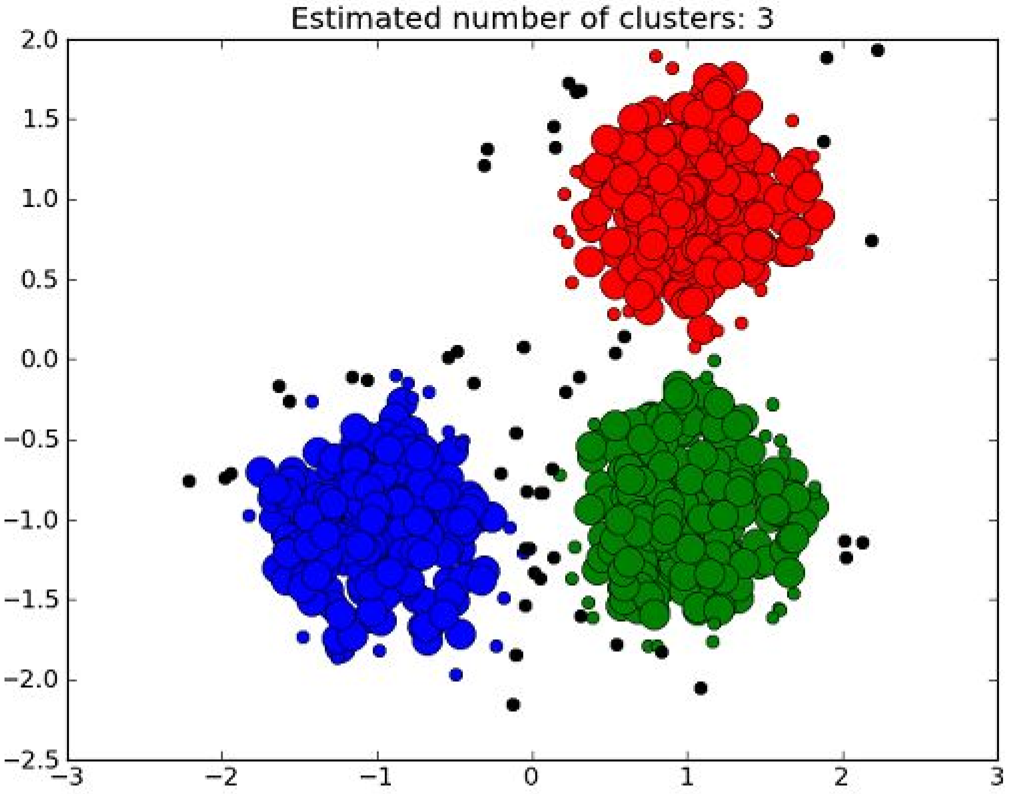
- Classification: When function/program being learned is **discrete**.
- Regression: When the function/program being learned is **continuous**.

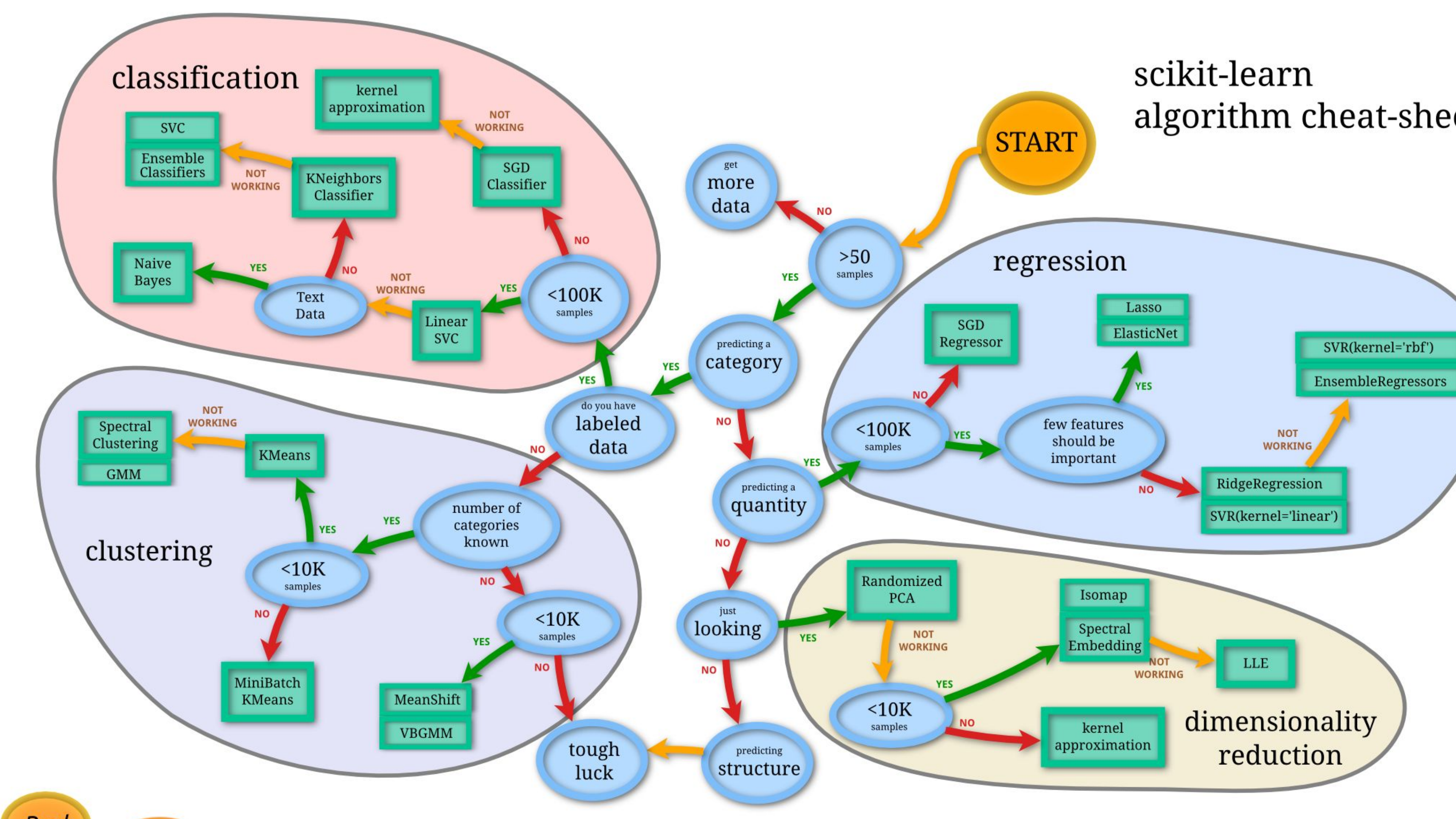


Clustering

- Don’t know the number of clusters or what the labels are

- Unsupervised





scikit-learn
algorithm cheat-shee
START
>50 samples
get more data
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
<100K samples
Linear SVC
Text Data
Naive Bayes
KNeighbors Classifier
SVC
Ensemble Classifiers
SGD Classifier
kernel approximation
clustering
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
<10K samples
MeanShift
VBGMM
regression
<100K samples
SGD Regressor
few features should be important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
<10K samples
Isomap
Spectral Embedding
LLE
kernel approximation
YES
NO
NOT WORKING

Training

- For all of these algorithms we have to give it data and let it train for one iteration
    - Can train for many iterations
    - We DO NOT train for infinity time
    - A lot of algorithms do not have a final state
- Each iteration, the algorithm learns more about the data

How to measure accuracy?

- If we are letting the algorithm learn about the data we pass to it, how do we measure the accuracy correctly?

How to measure accuracy?

- If we are letting the algorithm learn about the data we pass to it, how do we measure the accuracy correctly?
  - We cannot measure how many correct it guesses on the training set because it is biased
  - It has seen the data before and learned from it.
  - Showing it the same data would let it cheat

How to measure accuracy?

- If we are letting the algorithm learn about the data we pass to it, how do we measure the accuracy correctly?
    - We cannot measure how many correct it guesses on the training set because it is biased
    - It has seen the data before and learned from it.
    - Showing it the same data would let it cheat
- Instead we need to give it new data which it has not trained on
    - This requires more data

Training Set vs Test Set

- Instead of acquiring more data, let’s split the data we already have
- The training set will be images for the algorithm to train on
- The test set will only be for the algorithm to guess on

Have you ever seen a swan?

Have you ever seen a swan?

What about a black swan?

Garbage In Garbage Out

- If your data is not representative of the wider world, it will fail on real world data
- This can come from not having a wide enough data set
- Also can happen from biases in the data collection

On two occasions I have been asked, "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" ... I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.

— Charles Babbage, *Passages from the Life of a Philosopher*[2]

Too much training

What happens if we train for infinite rounds?

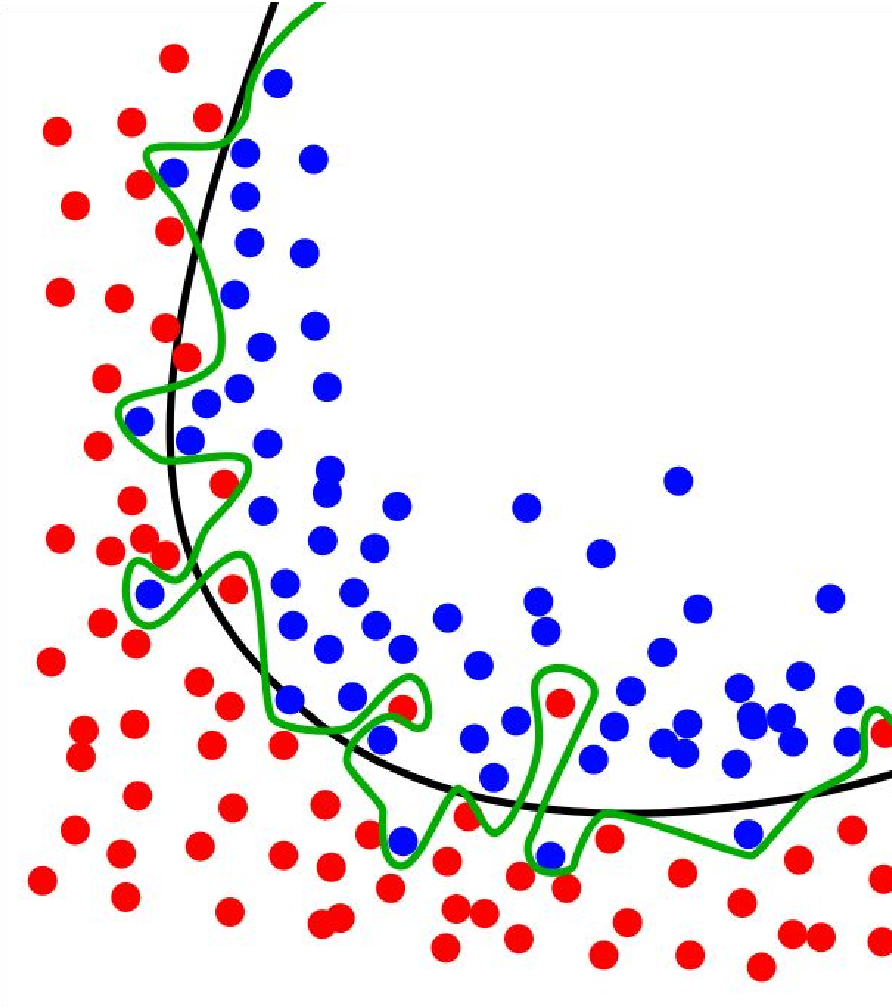
Too much training

What happens if we train for infinite rounds?

- Data has random variation
- Training data will have the same random variation for every time it is trained
- So algorithms will try to learn the random variation

Overfitting

- If the data is not varied enough or too similar we it may adapt to patterns that only exist within your data set, not the real world
- It will adjust to the random noise within the data set



How can we fix overfitting?

- We will always have random noise
- We cannot get rid of it

How can we fix overfitting?

- We will always have random noise
- We cannot get rid of it

We can reduce it by

- Adding random noise to the data set
- Using the accuracy on the test set to select when to stop training

When you should use MLib

- Obviously when you have extremely large data
- When it is not GPU intensive
    - A lot of machine learning benefits from a single GPU than 100 CPUs

Lab 5 Advice

- Again, don’t wait until Thursday
  - (seriously, this week’s lab will take some time)
- There are really good programming guides online for spark

Projects

Start thinking of ideas

Start building groups

Start identifying what technologies you want to use

- We’ll teach you new ones if we haven’t covered the one you want

Project Examples

- These are mainly about exploring a technology and explaining what you ran into
- The TAs and the rest of iDSI are currently working on
  - Setting up Ambari on Openstack
  - Running HDFS on Openstack
  - Graph Anomaly Detection using GraphX
  - Using Fabric to manage a cluster on Openstack
  - Loading Custom Binary Formats (aka Bitcoin) on Hadoop
  - Optimizing PySpark
  - Setting up Spark

Project Idea

- We’re trying to build a graph of things people need to use Hadoop/Spark/Others on OpenStack
- We need reports on things like
  - Cleaning datasets using Spark
  - Exploring when to switch to using MLib from Sk-learn
  - Dealing with Ambari failures
  - Loading large datasets into OpenStack
  - Validating data using Spark
  - Confidence Levels on Data Quality
  - Etc…

Class is going to change

- For the next 3ish weeks things will stay the same
- After that though, this class will is going to become more about peer review and advice on pursuing your technical papers
- We plan on having around 2-3 technical papers per group
  - These do not have to be extremely long (2-3 pages)
  - If you have a very deep topic, you can publish only one
- There will be presentations on your technical paper
  - No need for slides, you'll have your paper projected and you'll have 3 minutes to talk about it
  - We are still ironing out details about the presentation, we'll have more on it next week