An Overview of Classification Algorithms.

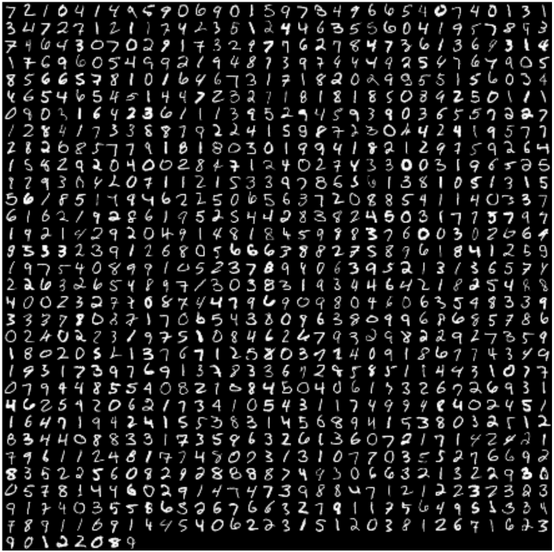
Colin Carroll

June 12, 2013

Goals

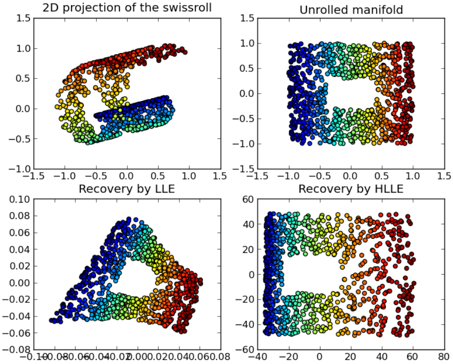
- Overview of what the problem is.
- High level view of what is available.
- *Some* intuition of how it works.

Goals



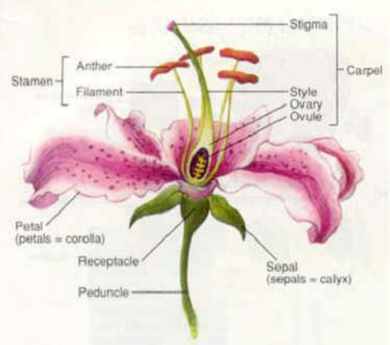
Not goals

- Feature detection.
- Dimensionality reduction.
- Tuning models.
- Pros and cons.



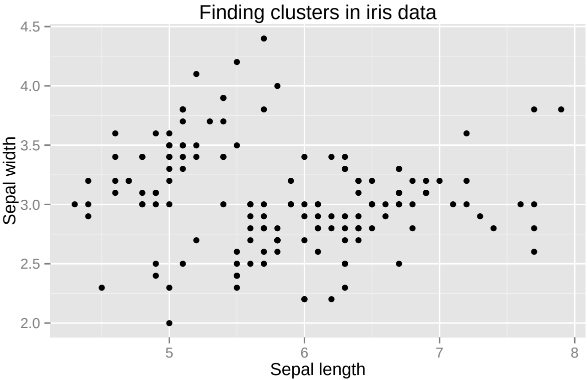
The iris dataset

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica



	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica

The iris dataset



Supervised vs. unsupervised learning

Unsupervised learning

- Should expect results to be worse.
- Not magic.
- Easiest if we assume all data is 2 dimensional.

K-means clustering

We assume there will be k clusters. Initialize by selecting k points $\mu_1, \ldots, \mu_k$. Proceed by

- Assigning each point to the μ_j nearest to it. Gives k sets $C_1, \ldots, C_k$.
- Choose new μ_j's by letting $\mu_j = \text{average}(C_j)$.

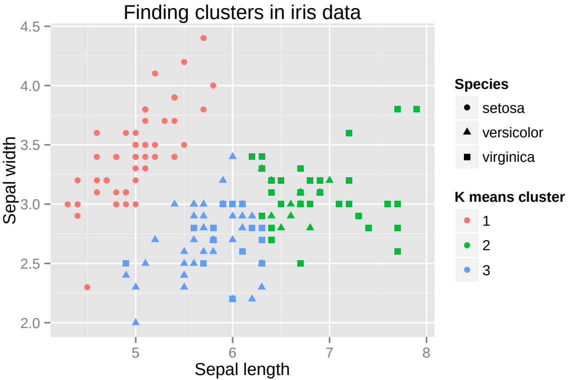
Stop when the C_j's stop changing.

K-means clustering

Given N observations $X = \{x_1, \ldots, x_N\}$, find a partition $S_1, \ldots, S_K$ so that

$$S_1, \ldots, S_K = \arg\min \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\|.$$

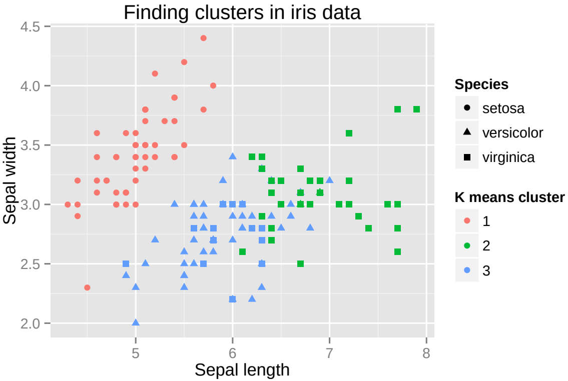
K-means clustering



K-means clustering

	1	2	3
setosa	50	0	0
versicolor	0	12	38
virginica	0	35	15

K-means clustering



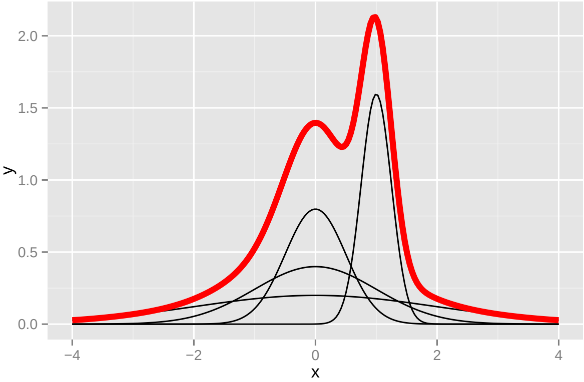
K-means clustering

	1	2	3
setosa	50	0	0
versicolor	0	2	48
virginica	0	36	14

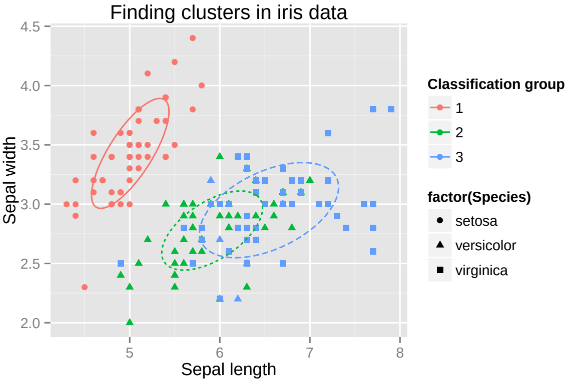
Mixture Modeling

We assume the members of each (unknown) cluster has been drawn from a different population.

Mixture Modeling



Mixture Modeling

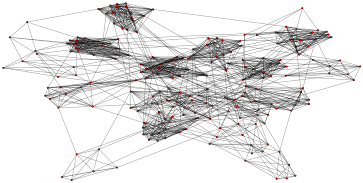


Mixture Modeling

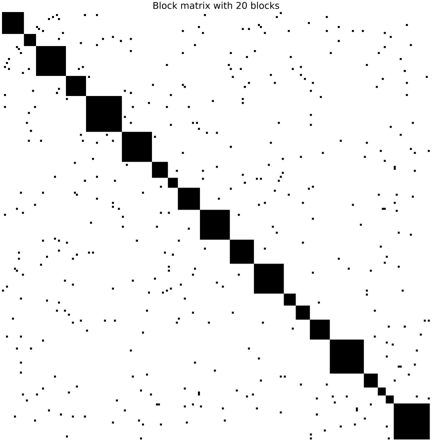
	1	2	3
setosa	50	0	0
versicolor	0	45	5
virginica	0	0	50

Network Data

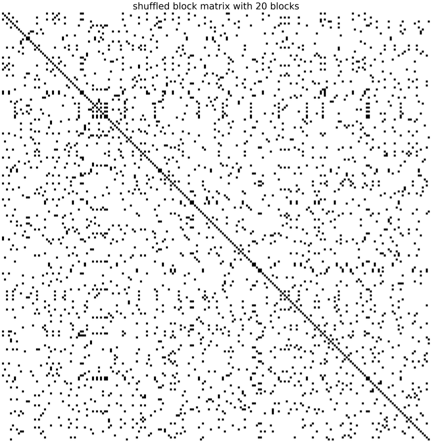
Too much to say for this, here are some pictures on clustering.



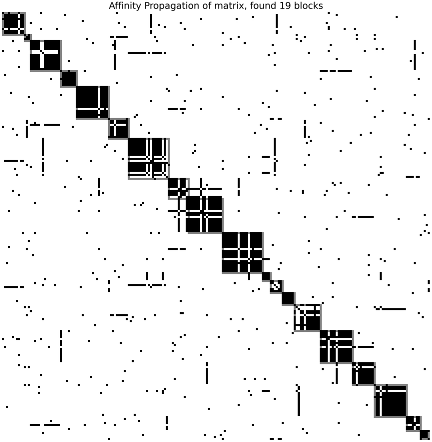
Network Data



Network Data



Network Data



Supervised learning

- Divide our data set into 120 training examples and 30 test examples.
- Not magic.
- Easiest if we assume all data is 2 dimensional.

Supervised learning

Broadly speaking, we will have a training set $\{x_j, \hat{y}_j\}$, where x_j are data and y_j are categories, and a loss function L that penalizes our model for being wrong. We use the training set to train a (hopefully simple) function f so that $\sum_j L(f(x_j), \hat{y}_j)$ is minimized.

Naive Bayes

$$p(\text{event}|\text{feature}) = \frac{p(\text{event})p(\text{feature}|\text{event})}{p(\text{feature})}$$

- $p(\text{tb}) = 0.005$
- $p(\text{positive}|\text{tb}) = 0.99$
- $p(\text{positive}|!\text{tb}) = 0.05$
- (so $p(\text{positive}) = 0.055$).

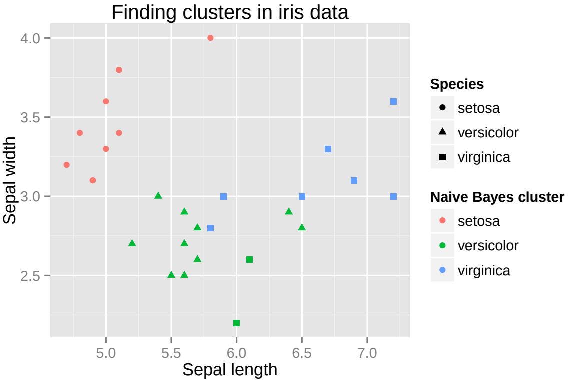
$$p(\text{tb}|\text{positive}) = \frac{0.005 * 0.99}{0.055} = 9\%.$$

Naive Bayes

1. Estimate the distribution of sepal lengths and widths for each iris species
2. Given some sepal lengths and widths, we'll be able to calculate the probability the event belongs to each category
3. Choose the category with the highest probability (*posterior*)

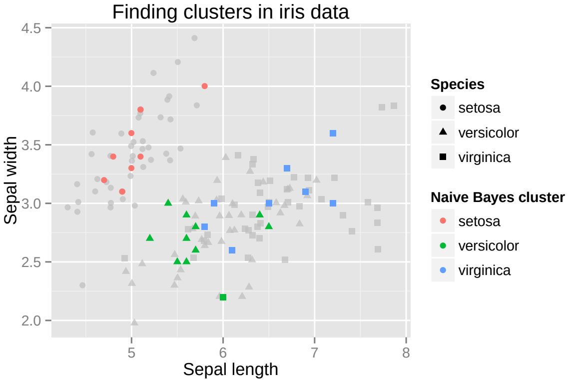
Called *naive* Bayes because we assume sepal length is independent from sepal width. Which isn't super realistic. Still does ok!

Naive Bayes


Finding clusters in iris data
4.0
3.5
3.0
2.5
Sepal width
5.0
5.5
6.0
6.5
7.0
Sepal length
Species
setosa
versicolor
virginica
Naive Bayes cluster
setosa
versicolor
virginica

K nearest neighbors

Ask your K nearest neighbors which group you belong in.



K nearest neighbors

Results with 2-dimensional training data:

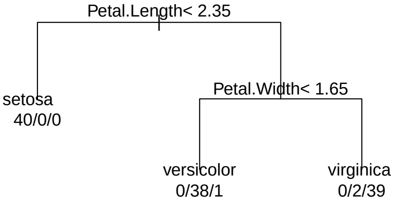
	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	8	2
virginica	0	4	6

Results with 4-dimensional training data:

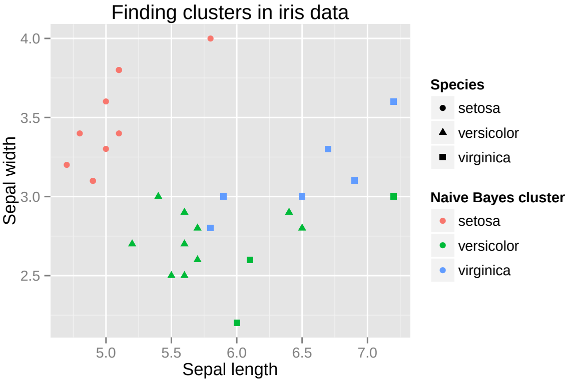
	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	0
virginica	0	1	9

Decision tree

Find which question divides your training set the most homogeneously, repeat.



Decision tree



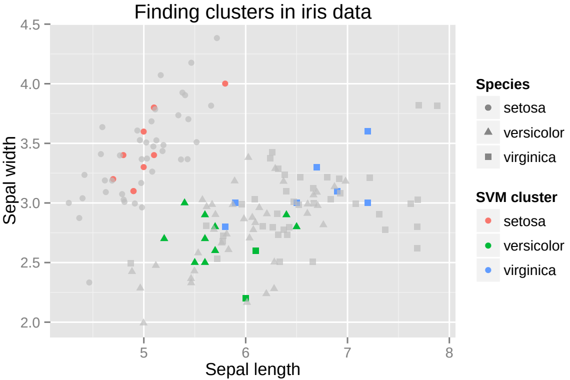
Decision Tree

Results with 4-dimensional training data:

	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	0
virginica	0	3	7

Support vector machines

Divide the space using a hyperplane.



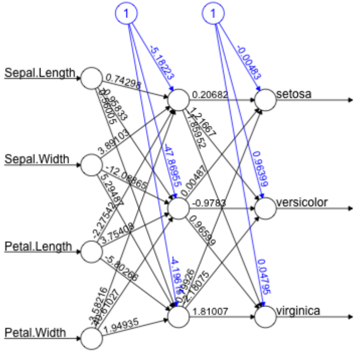
Support vector machines

	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	0
virginica	0	2	8

Neural nets

Compose some activation functions together.

$$\sum_j \frac{1}{1+e^{x_j \cdot w_j}}$$



Neural nets

	prediction		
	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	0
virginica	0	3	7