Exercises

1) Are labels required for supervised or unsupervised machine learning algorithms? supervised

2) Calculate the manhattan distance matrix for genes with the following observations (don't normalize):

Gene	S1	S2	S3
A	1	5	3
B	3	2	5
C	5	1	4
D	4	1	4

	A	B	C	D
A	0	7	9	7
B		0	4	3
C			0	1
D				0

d(A,B) = abs(1-3) + abs(5-2) + abs(3-5) = 7

Exercises

3) Calculate $d_{A,BD}$ and $d_{C,BD}$ using the distance matrix and a) complete linkage as well as b) single linkage

	A	B	C	D
A	0	10	19	10
B	10	0	11	4
C	19	11	0	13
D	10	4	13	0

	single	complete
d(A,BD)	min(10,10)	max(10,10)
d(C,BD)	min(11,13)	max(11,13)

Exercises

4) What is the hidden variable in K-means clustering? Labels or cluster IDs

5) Why is normalization used prior to clustering? To ensure equal weighting of samples when generating clusters.

6) Update the cluster assignments (blue) based on the distances, and calculate new centroids (green) using this data:

Centroid 1	1	1	3			
Centroid 2	2	2	1			
Centroid 3	3	2	2			
	Cluster	S1	S2	Distance1	Distance2	Distance3
A	2	1	2	9	1	2
B	3	3	3	8	8	5
C	1	6	7	29	61	52
D	3	2	2	10	2	1
E	1	2	5	1	17	16
F	1	1	5	0	16	17
G	3	7	2	45	37	26
H	2	1	2	9	1	2
Centroid 1	1	3	5.6			
Centroid 2	2	1	2			
Centroid 3	3	4	2.3			

Exercises

7) What metric does K-means use to assign rows to clusters? Sum of squared deviations

8) What cluster visualization methods would you use for the following tasks:

a) up-regulated genes in specific samples heatmap

b) Number of clusters as a function of distance between them dendrogram

d) Assign ancestry to an individual Fuzzy/mixture model

9) What type of machine learning would you use to predict benign/malignant (classification, regression)?

10) What algorithm/model can accurately handle two distributions (groups/labels) that overlap? Fuzzy clustering or Mixture model

11) What is the difference between soft and hard clustering? Soft one has membership to multiple groups

12) What is the advantage of soft over hard clustering? Soft can handle individuals that are mixtures of multiple groups, can handle overlapping distributions

13) Mixture models use [hard/soft] labels such that assignments to groups is not discrete.

Today’s objectives

Visualizing data

Principal Component Analysis, t-SNE

Supervised machine learning

- Linear discriminants

- Support vector machines

- Decision trees and Random forests

2-D vs N-dimensional clustering

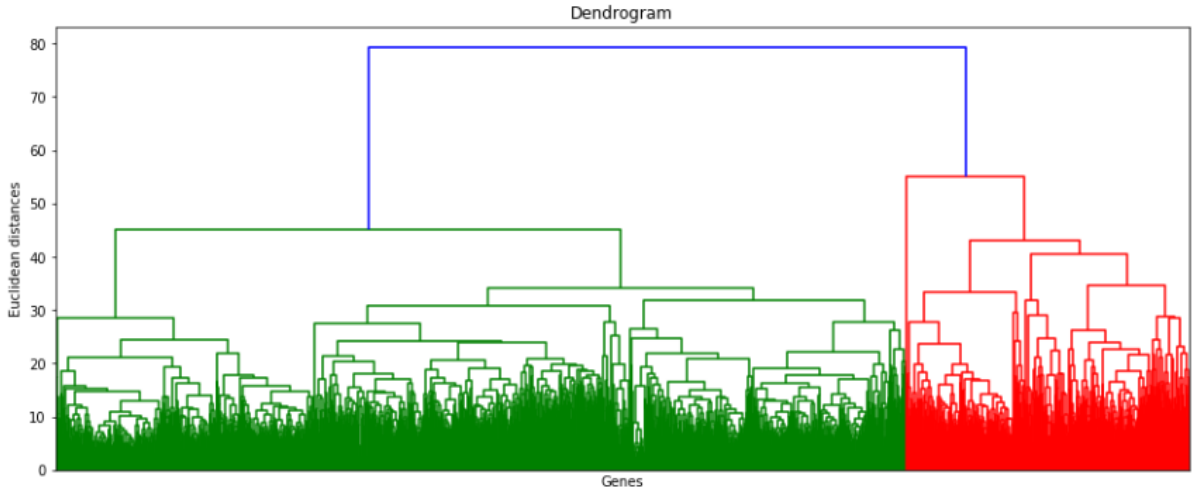
K-means clustering
S1 and S2
K = 4
Plot S1 vs S2
Color by cluster

K-means clustering
S1 … S173
K = 4
Plot S1 vs S2
Color by cluster

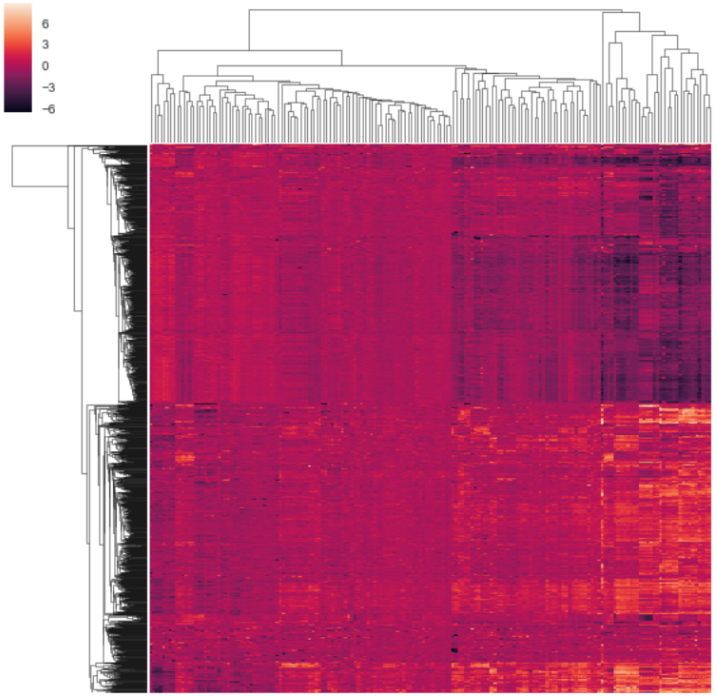
How do you visualize high dimensional clustering?

Cluster visualization

Dendrogram from hierarchical clustering



Heatmap from hierarchical clustering



PCA with K-means clustering

Dendrogram: tree
Heatmap: tree + data
PCA: dimension reduction

Principal Components Analysis

Principal component analysis (PCA)

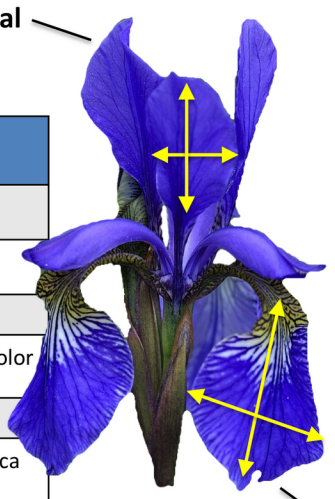
- A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

Many correlated variables-> 1 PC

- This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible).
- Each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set.

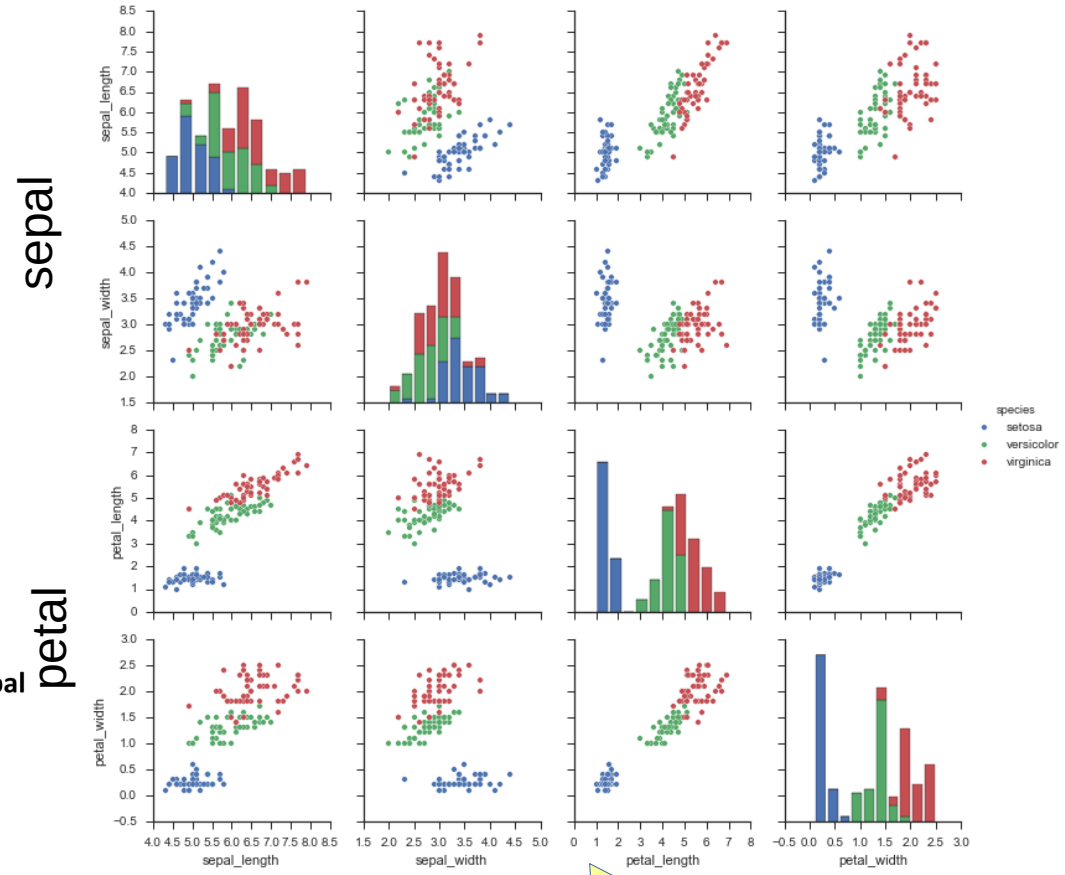
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

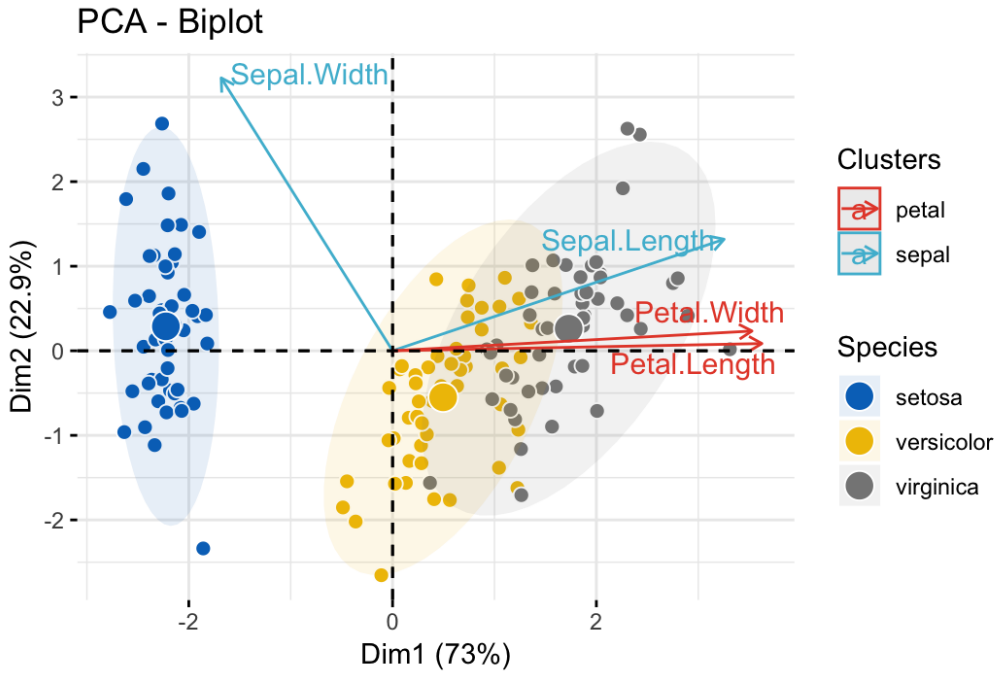


Features
(attributes, measurements, dimensions)

Class labels
(targets)



sepal correlated petal

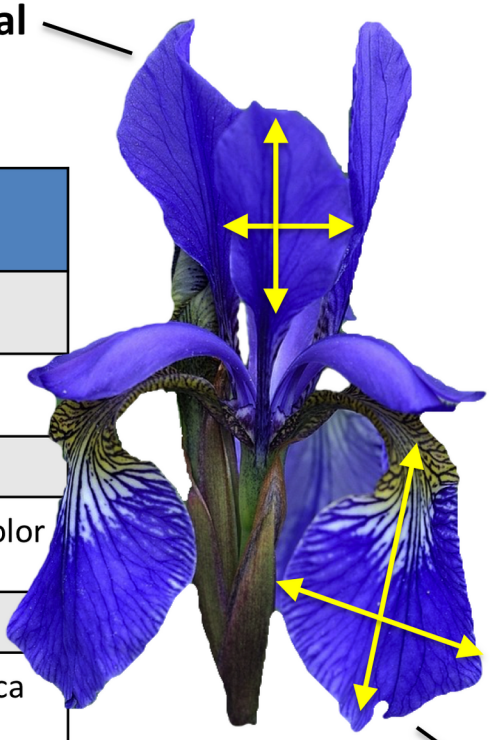


PCA reduces 4 dimensions to principal components that explain the most variation in the data
- species are not known (unsupervised)
- Up to 4 PCs are returned, but last explains little variance

Four species, with 4 flower measurements
Use PCA to visualize differences

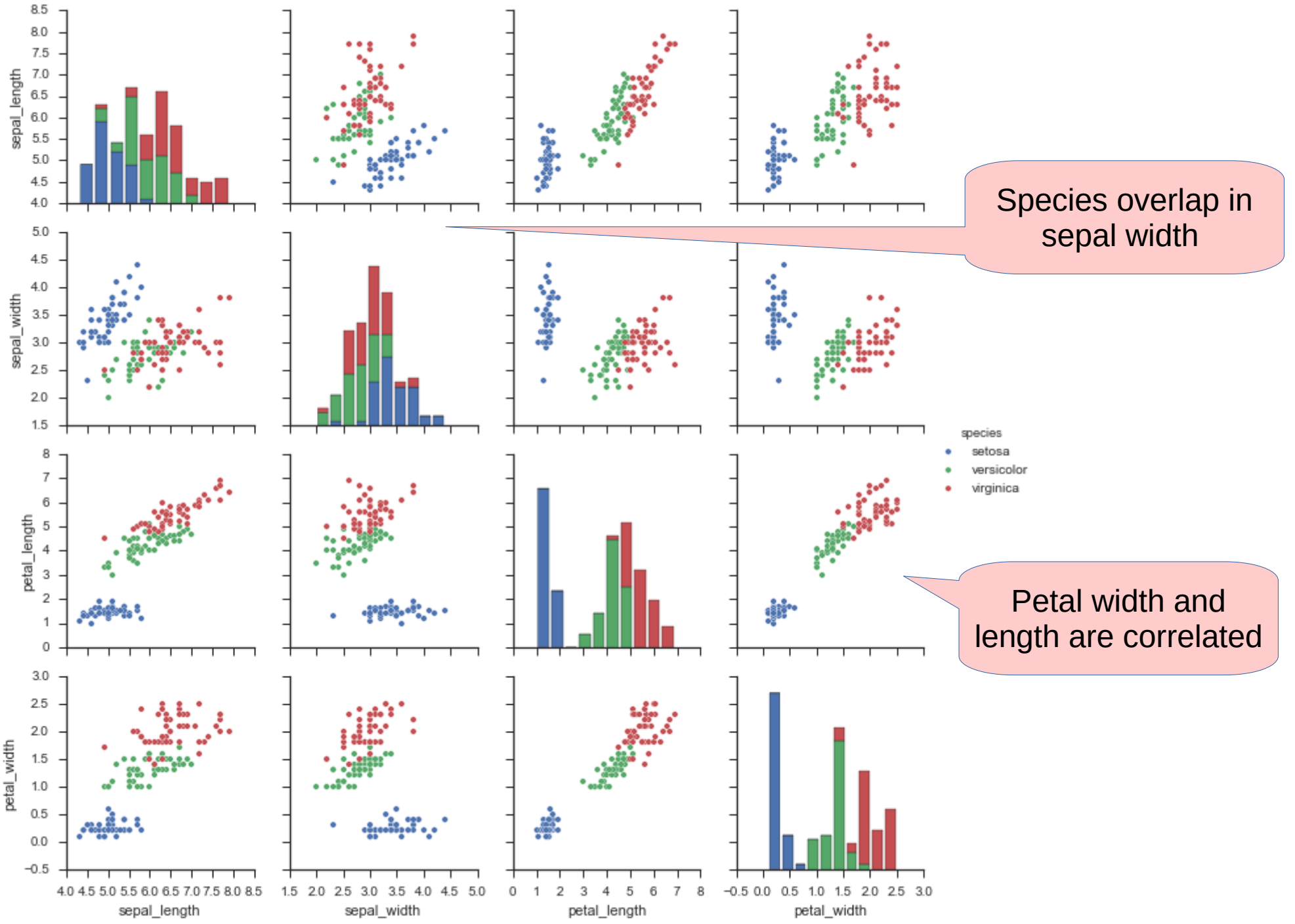
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

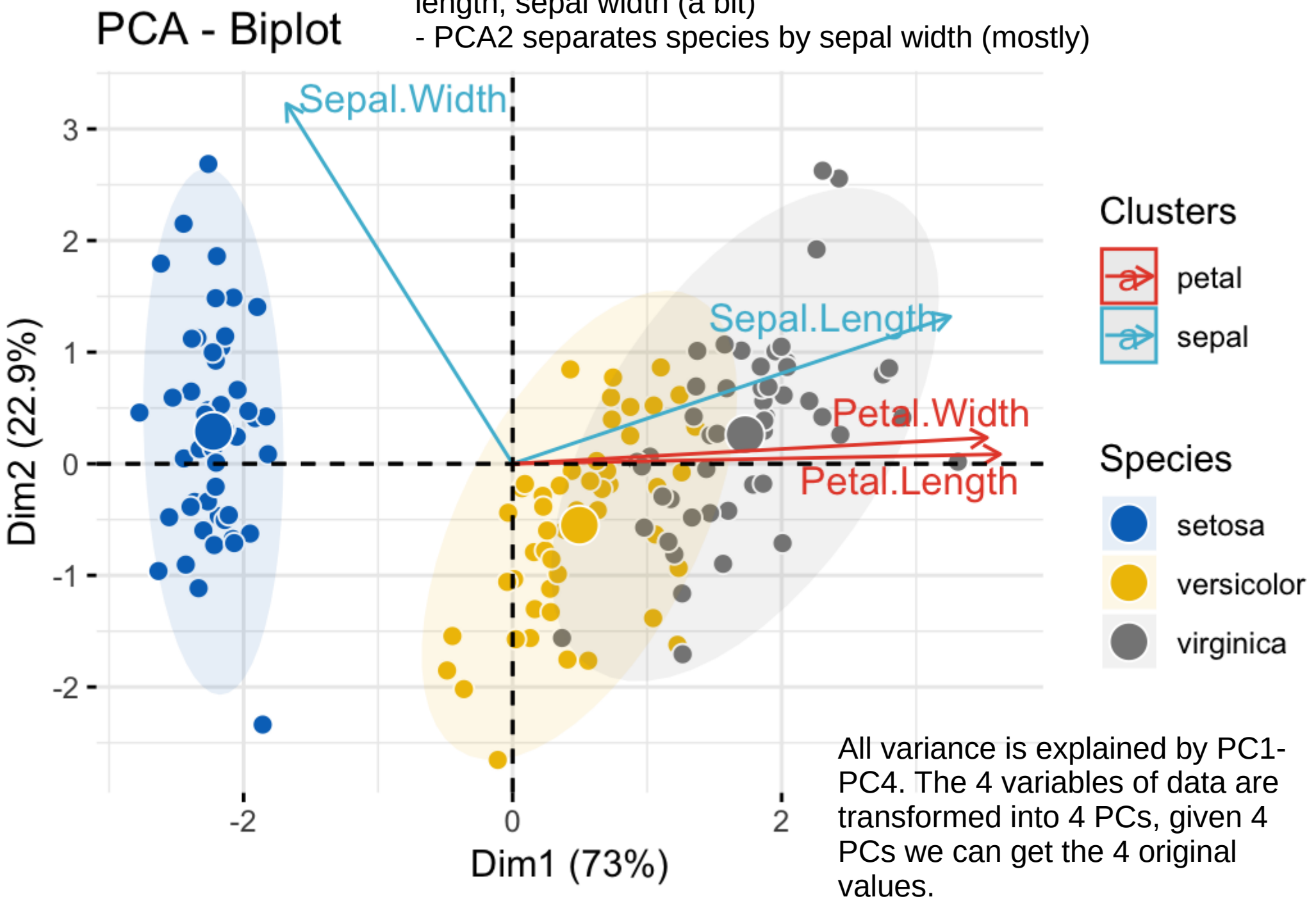


Features
(attributes, measurements, dimensions)

Class labels
(targets)


sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica
Species overlap in sepal width
Petal width and length are correlated

PCA loadings show how each character affects PC
- PCA1 separates species by petal width and length, sepal length, sepal width (a bit)
- PCA2 separates species by sepal width (mostly)



All variance is explained by PC1-PC4. The 4 variables of data are transformed into 4 PCs, given 4 PCs we can get the 4 original values.

How to find PCs

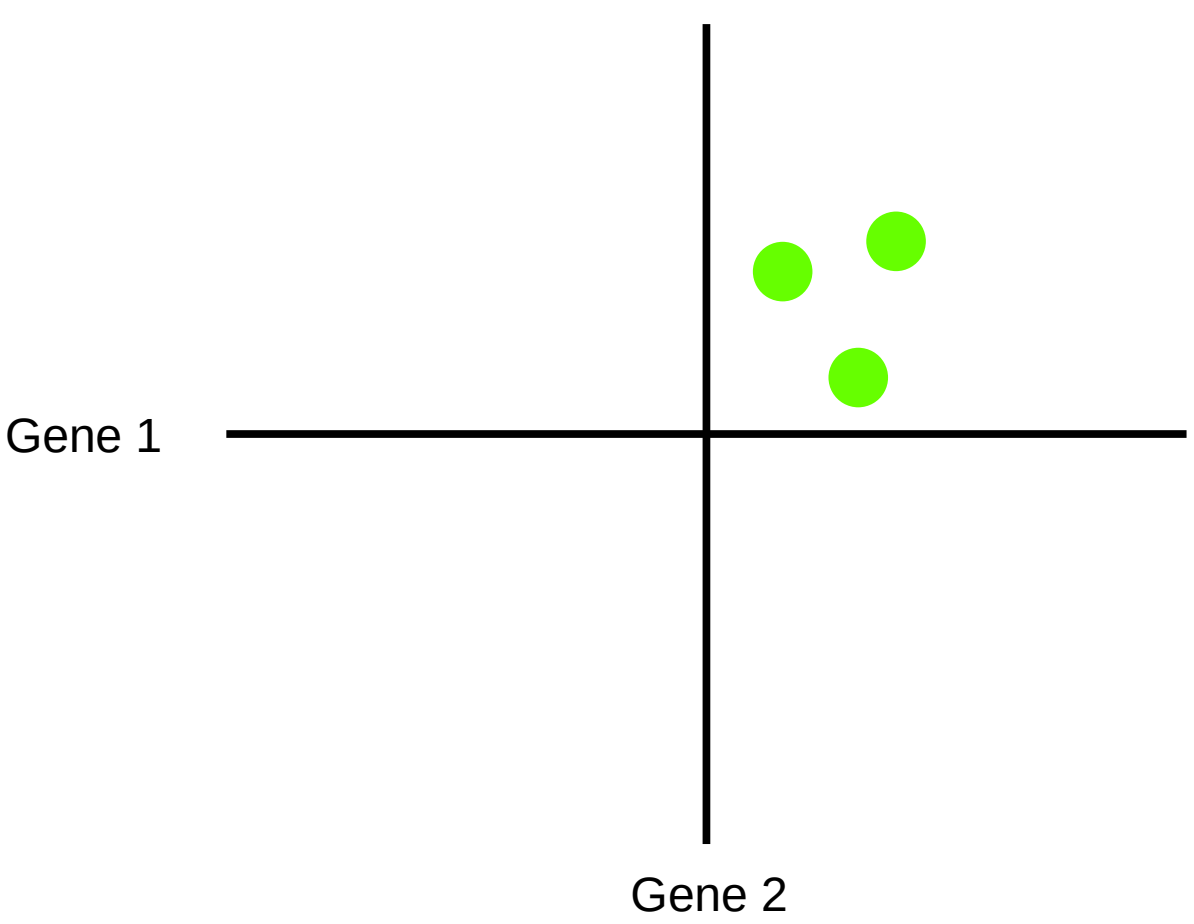
Step 1: Find a new coordinate system that captures the greatest variance of the data on the first coordinate (called the first principal component).

Step 2: Find a new coordinate system that captures the greatest remaining variance (after removing that capture by the first PC).

Step 3..n: Repeat until number of coordinates = number of variables.

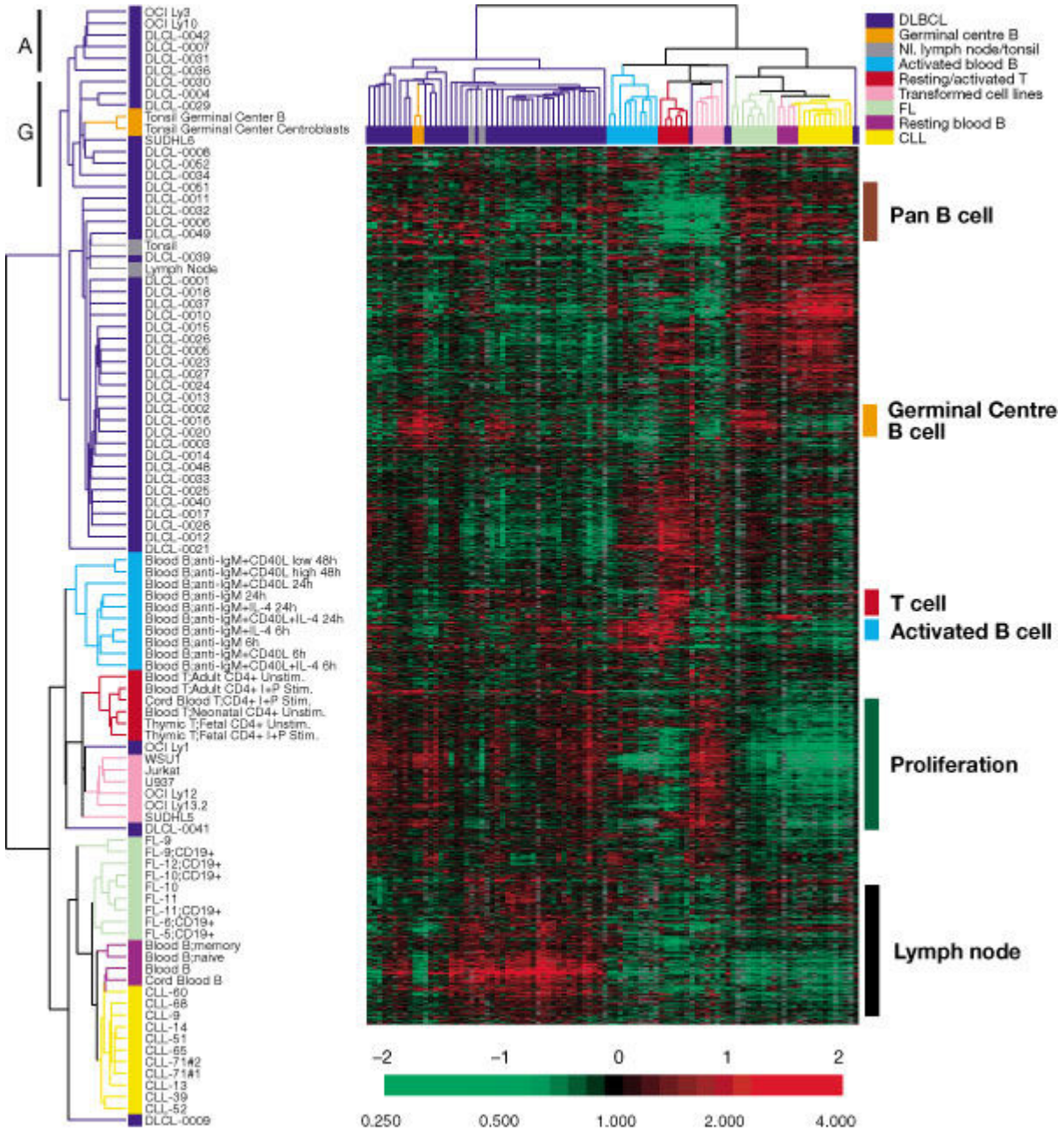
A simple 2D coordinate system

Step 1: Find a new coordinate system that captures the greatest variance of the data on the first coordinate (called the first principal component).



With two variables, find a new coordinate system that captures the greatest variance

Lets say we have many variables (genes) and we want to use PCA to show sample relationships.



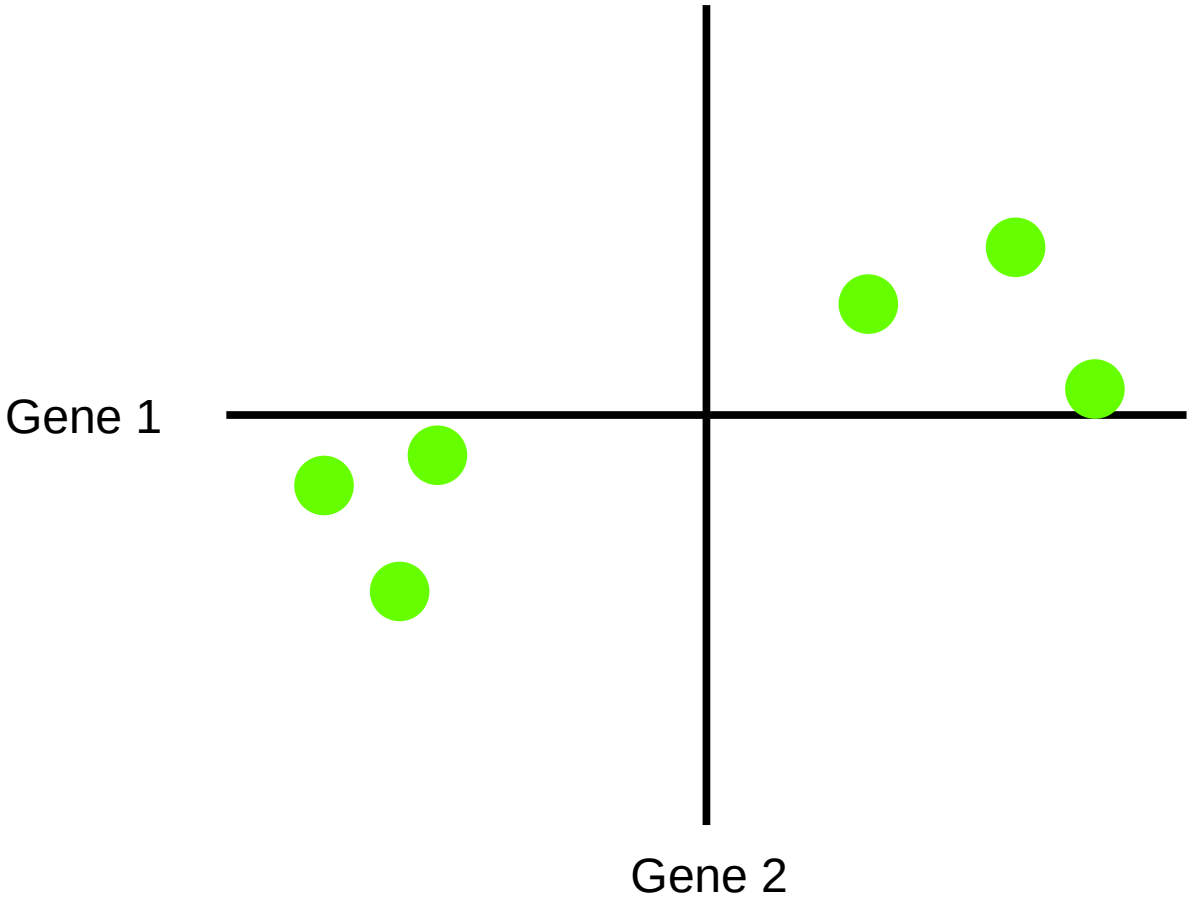
Genes are rows
Columns are samples

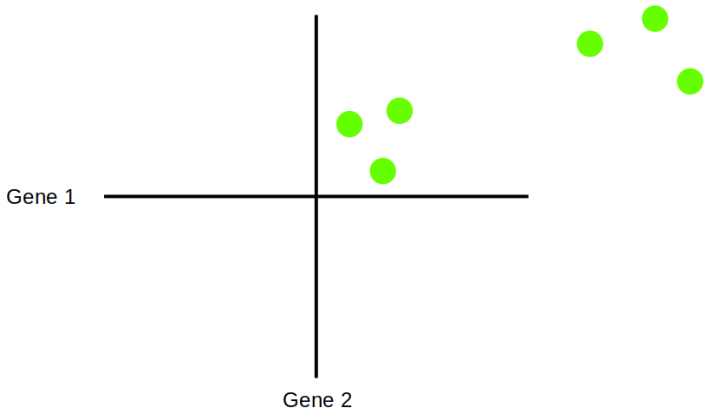
Dendrogram shows sample relationships as 'distance' across all genes

But how are the distances related to gene expression and which genes contribute?

Changing the coordinate system

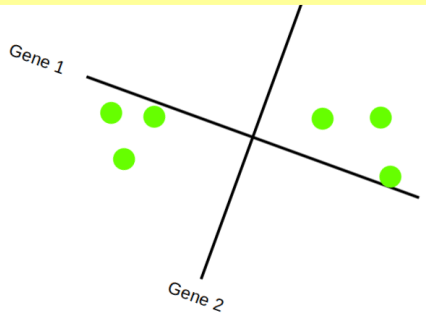
Step 1: Find a new coordinate system that captures the greatest variance of the data on the first coordinate (called the first principal component).





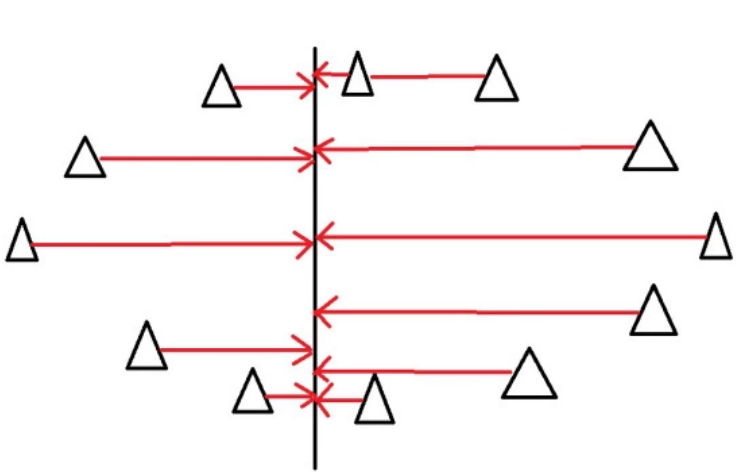
Shifting and rotating the values doesn't change their relationship.

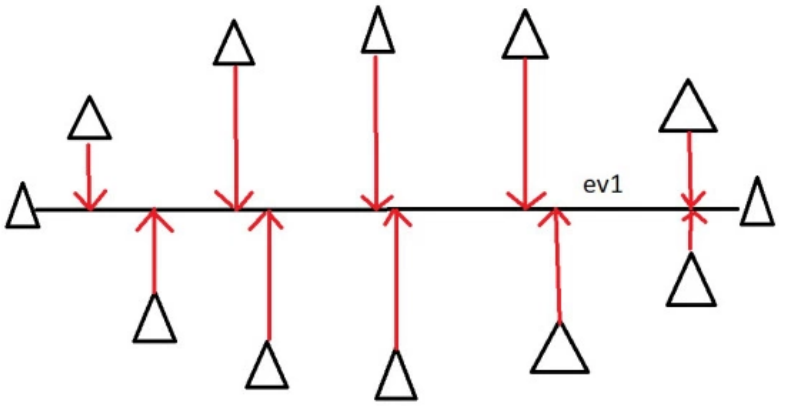
But the coordinate system changes



What vector has the most variance

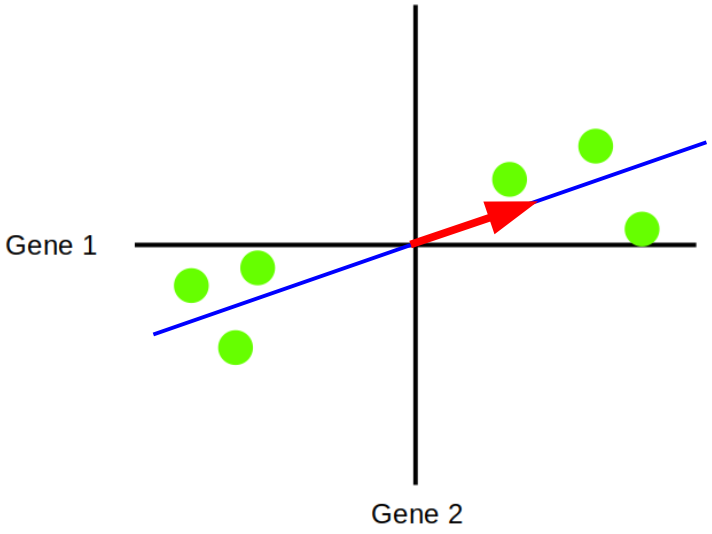
Find a direction (vector) where there is the most variance:

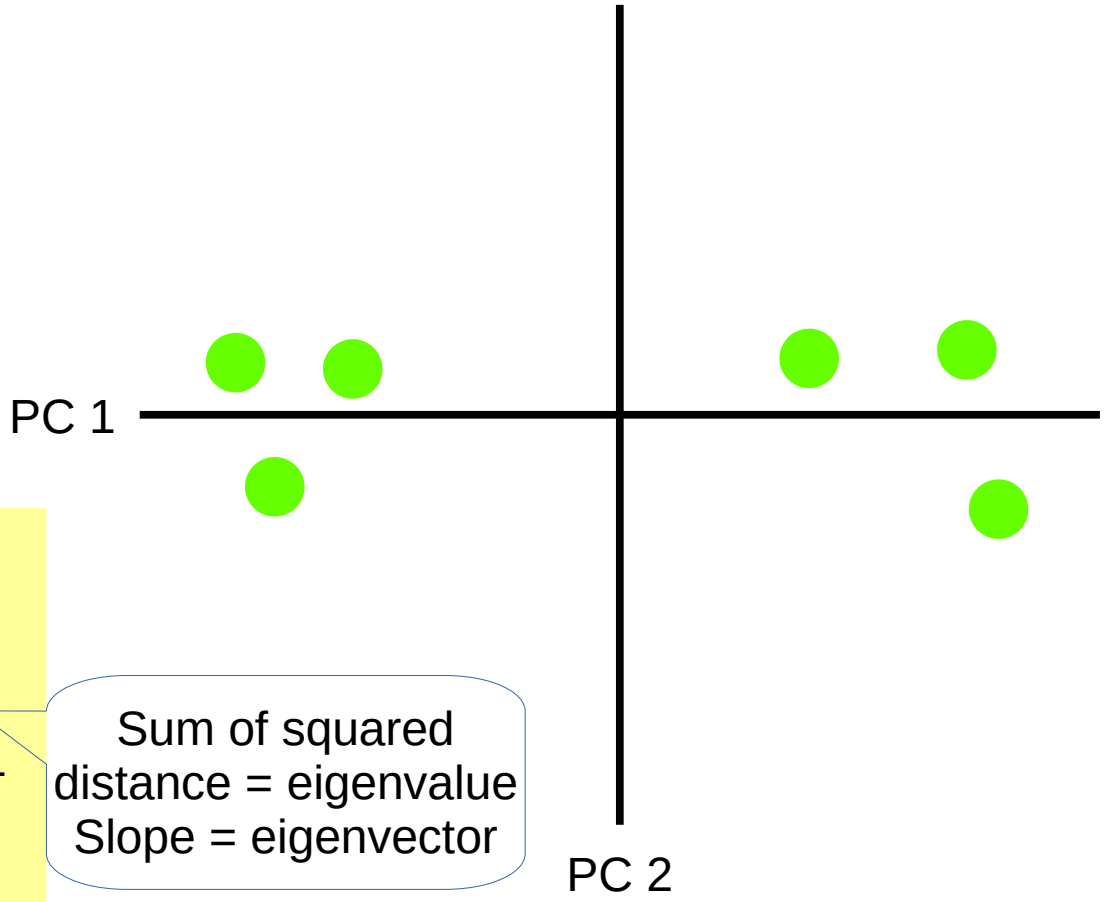




Minimizing the sum of the squared deviations from the vector = maximizing the sum of squared deviations to the perpendicular

Projection onto a new coordinate system





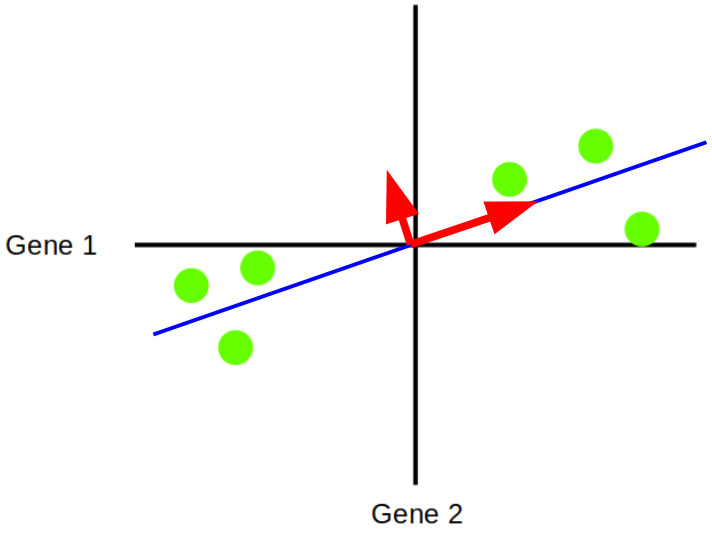
The first PC:

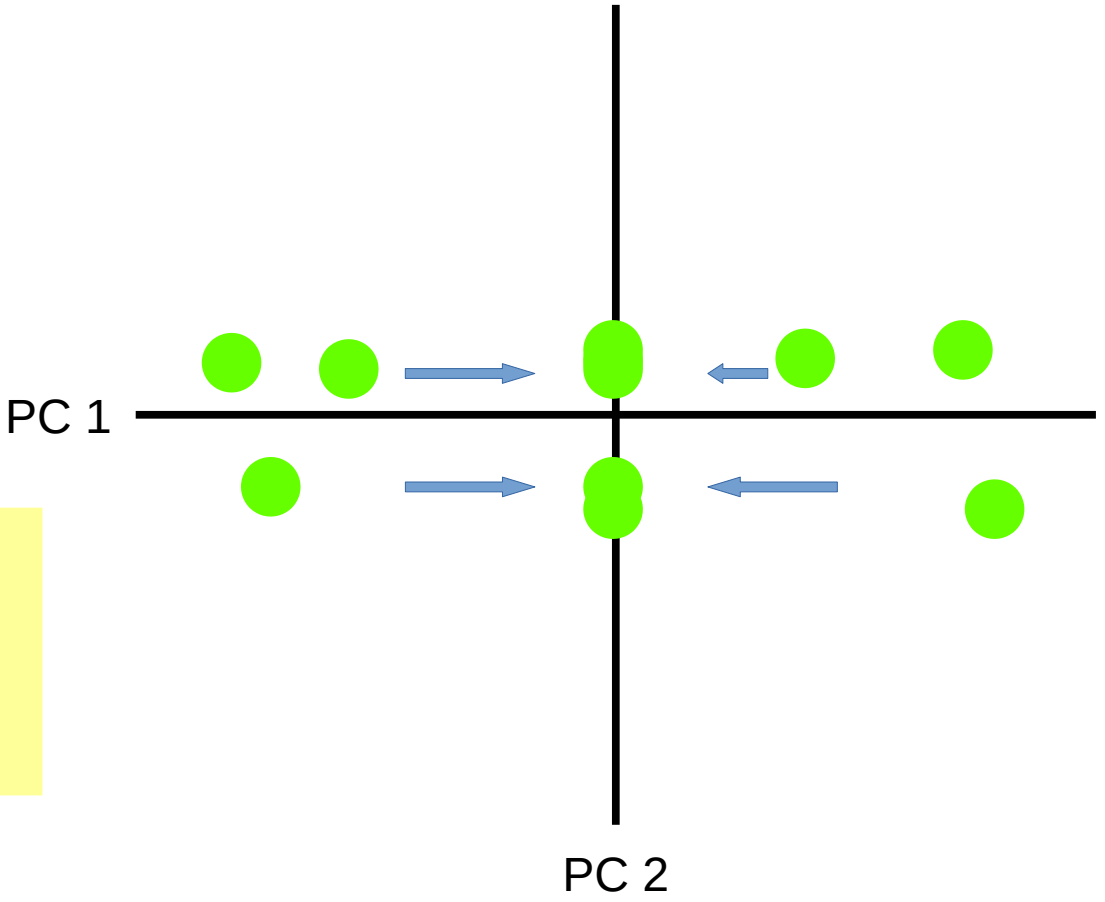
- Minimized deviations from line
- Maximizes distance from projected points to the origin (easier to compute)
- PC1 is a linear transformation of Gene 1 and Gene 2 (slope of the line = weights)
- Slope = 0.25, PC1 = 4 parts gene 1 to 1 part gene 2 (these are the loadings)

Sum of squared distance = eigenvalue
Slope = eigenvector

Finding PC2

Step 2: Find a new coordinate system that captures the greatest remaining variance (after removing that capture by the first PC).

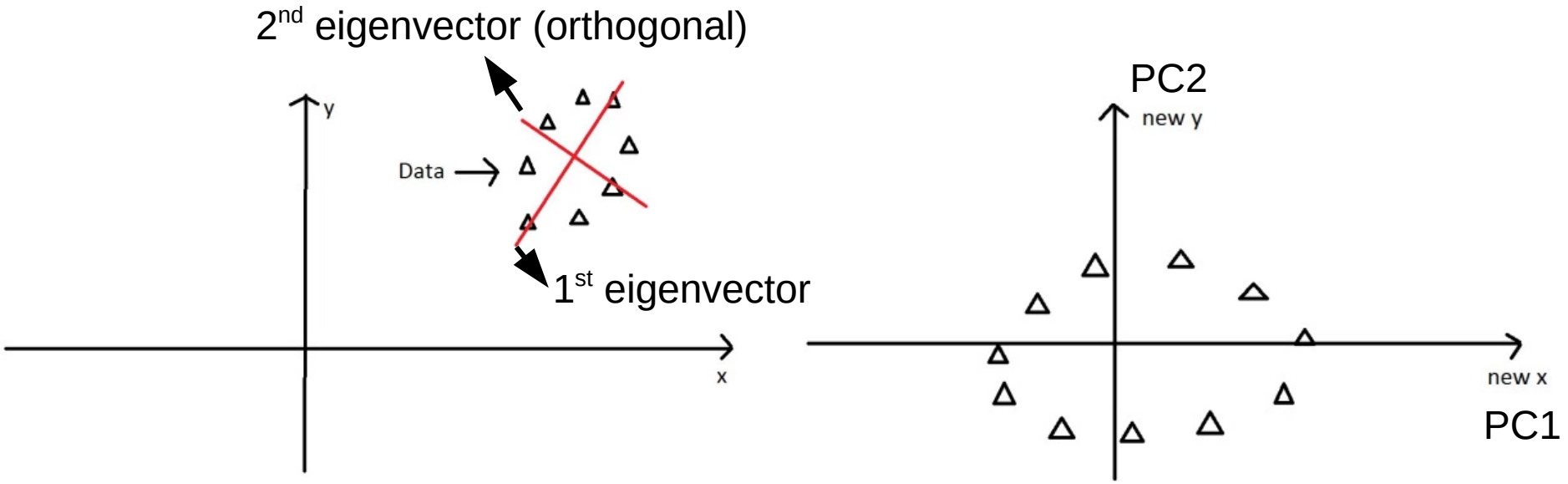




The second PC:

- Orthogonal (perpendicular) to PC1
- Subtract variation due to PC1
- PC2 captures all remaining variation if only two genes

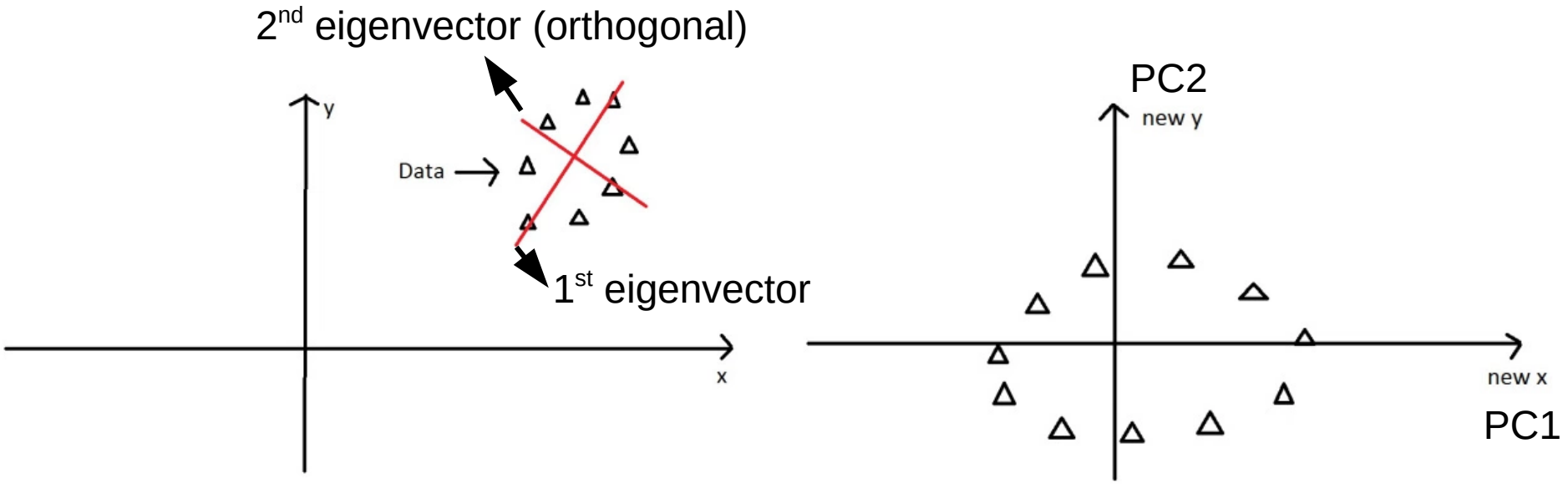
Transformation of coordinates 2D original to PC1 and PC2



What are the eigenvectors and eigenvalues for 2D-dimensional space?

- The eigenvector tells the new coordinate system with loadings (weights) of a linear combination of original variables
- The eigenvalue describe the variance whether the eigenvector is stretched or shrunk or reversed or left unchanged

Transformation of coordinates 2D original to PC1 and PC2



What are the eigenvectors and eigenvalues for 2D-dimensional space?

Eigenvectors and values exist in pairs: every eigenvector has a corresponding eigenvalue. An eigenvector is a direction, in the example above the eigenvector was the direction of the line (vertical, horizontal, 45 degrees etc.). An eigenvalue is a number, telling you how much variance there is in the data in that direction, in the example above the eigenvalue is a number telling us how spread out the data is on the line. **The eigenvector with the highest eigenvalue is a principal component.**

Orthogonal in 2-D and N-D

In 2-D space, two Euclidean vectors are orthogonal if they are perpendicular

In N-D Euclidean space, two vectors are orthogonal if and only if their dot product is zero

Dot product is an algebraic operation that takes two equal-length sequences of numbers (usually coordinate vectors) and returns a single number

The dot product of two vectors $a = [a_1, a_2, ..., a_n]$ and $b = [b_1, b_2, ..., b_n]$ is defined as:

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

The dot product can also be written using the transpose of a (a^T):

$$a \cdot b = a^T b$$

The transpose is defined as:

$$[A^T]_{ij} = [A]_{ji}$$

PCA

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component)

The full principal components decomposition of X_{np} (n rows, p columns) is: **T = XW**

X is the data matrix x_{ij} with n rows and p columns

W is the matrix of basis vectors, one vector per column, where each basis vector is one of the eigenvectors of the covariance matrix X^TX (loadings)

T is the matrix consisting of n row vectors, where each vector is the projection of the corresponding data vector from matrix X onto the basis vectors contained in the columns of matrix W, new coordinates

$$T_{n,L} = X_{n,p} W_{p,L} = \begin{bmatrix} x_{11} & \cdot & x_{1p} \\ \cdot & \cdot & \cdot \\ x_{n1} & \cdot & x_{np} \end{bmatrix} \begin{bmatrix} w_{11} & \cdot & w_{1L} \\ \cdot & \cdot & \cdot \\ w_{p1} & \cdot & w_{pL} \end{bmatrix}$$

Vector weights

The transformation is defined by a set of p-dimensional vectors of weights or loadings $w_{(k)}=(w_1,\ldots,w_p)_{(k)}$ that map each row vector $x_{(i)}$ of X to a new vector of principal component scores $t_{(i)}=(t_1,\ldots,t_m)_{(i)}$

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

The vector weights (w) are the eigenvectors.

With $w_{(1)}$ found, the first principal component of a data vector $x_{(i)}$ can then be given as a score $t_{1(i)} = x_{(i)} \cdot w_{(1)}$ in the transformed co-ordinates.

PCA example

```
Data
     0    1    2    3
0 -5.4  3.4  1.7  0.2
1 -5.5  2.4  3.8  1.1
2 -6.9  3.2  5.7  2.3
```

normalize

```
Standardized data
          0         1         2         3
0  0.778868  0.92582 -1.244639 -1.162476
1  0.632830 -1.38873  0.040808 -0.116248
2 -1.411698  0.46291  1.203831  1.278724
```

```
Covariance matrix
[[ 1.5        -0.40561396 -1.32151517 -1.39207624]
 [-0.40561396  1.5        -0.32585861 -0.1614366 ]
 [-1.32151517 -0.32585861  1.5         1.49074349]
 [-1.39207624 -0.1614366   1.49074349  1.5       ]]
```

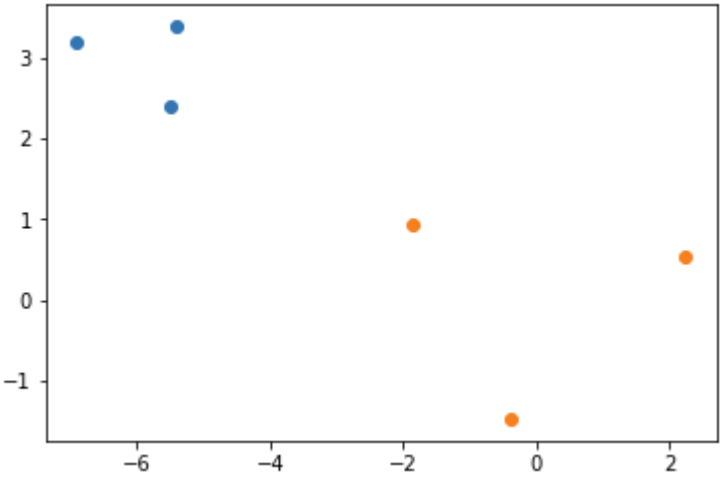
```
Correlation matrix
[[ 1.         -0.2704093  -0.88101011 -0.92805083]
 [-0.2704093   1.         -0.21723907 -0.1076244 ]
 [-0.88101011 -0.21723907  1.          0.99382899]
 [-0.92805083 -0.1076244   0.99382899  1.        ]]
```

```
Eigenvectors
[[-0.56256955 -0.28482127  0.2085529  -0.01158754]
 [-0.01992769  0.94022054  0.01238822 -0.0842887 ]
 [ 0.58017645 -0.17325102 -0.58923706 -0.71656212]
 [ 0.58865412 -0.06961505  0.78048184  0.69231487]]
```

```
Eigenvalues
[  4.30512951e+00   1.69487049e+00  -9.60124067e-17  -2.08427989e-17]
```

PC1	PC2
-1.86302341	0.94519775
-0.37309067	-1.4849335
2.23611408	0.53973575

Data[:,0], Data[:,1]
PCA1, PCA2



Calculating PC1

	0	1	2	3	
Eigenvalues	2.87	1.13	0.00	0.00	
Eigenvector	-0.56	-0.02	0.58	0.59	Weights
	0.00	1.00	2.00	3.00	
0	0.78	0.93	-1.24	-1.16	
1	0.63	-1.39	0.04	-0.12	
2	-1.41	0.46	1.20	1.28	
A^TW					Sum
0	-0.44	-0.02	-0.72	-0.68	-1.86
1	-0.36	0.03	0.02	-0.07	-0.37
2	0.79	-0.01	0.70	0.75	2.24
	PC1				
0	-1.86				
1	-0.37				
2	2.24				

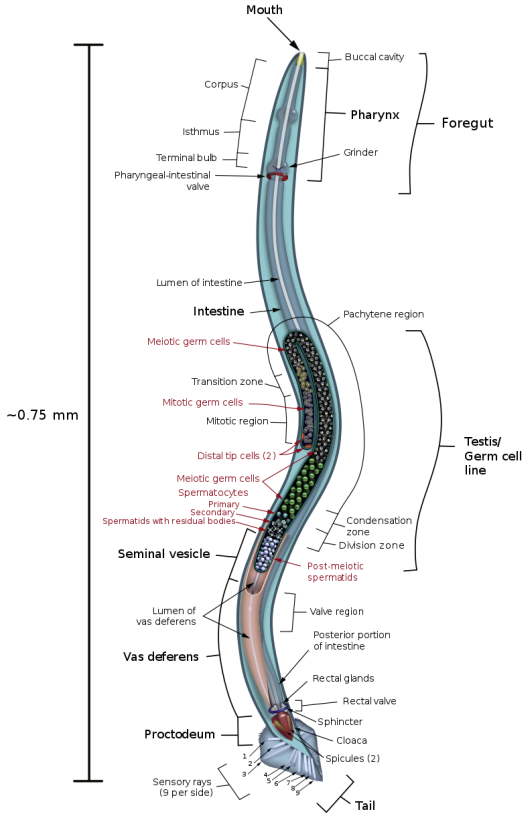
0.78 * -0.56 = -0.44
0.63 * -0.56 = -0.36
-1.41 * -0.56 = 0.79

0.93 * -0.02 = -0.02
-1.39 * -0.02 = 0.03
0.46 * -0.02 = -0.01

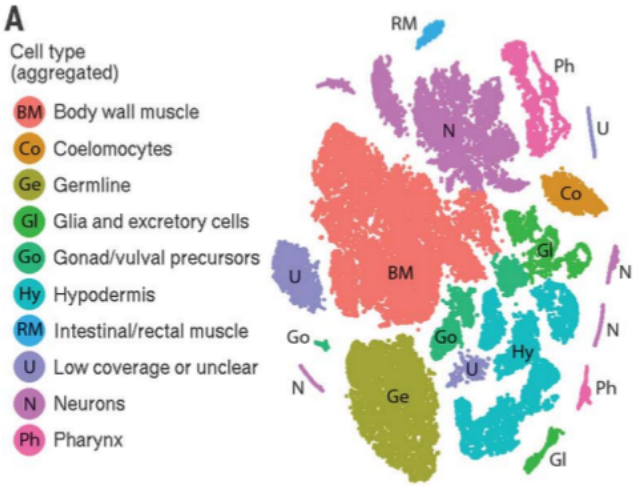
Single Cell Analysis and t-SNE

C. elegans

1031 cells (male)



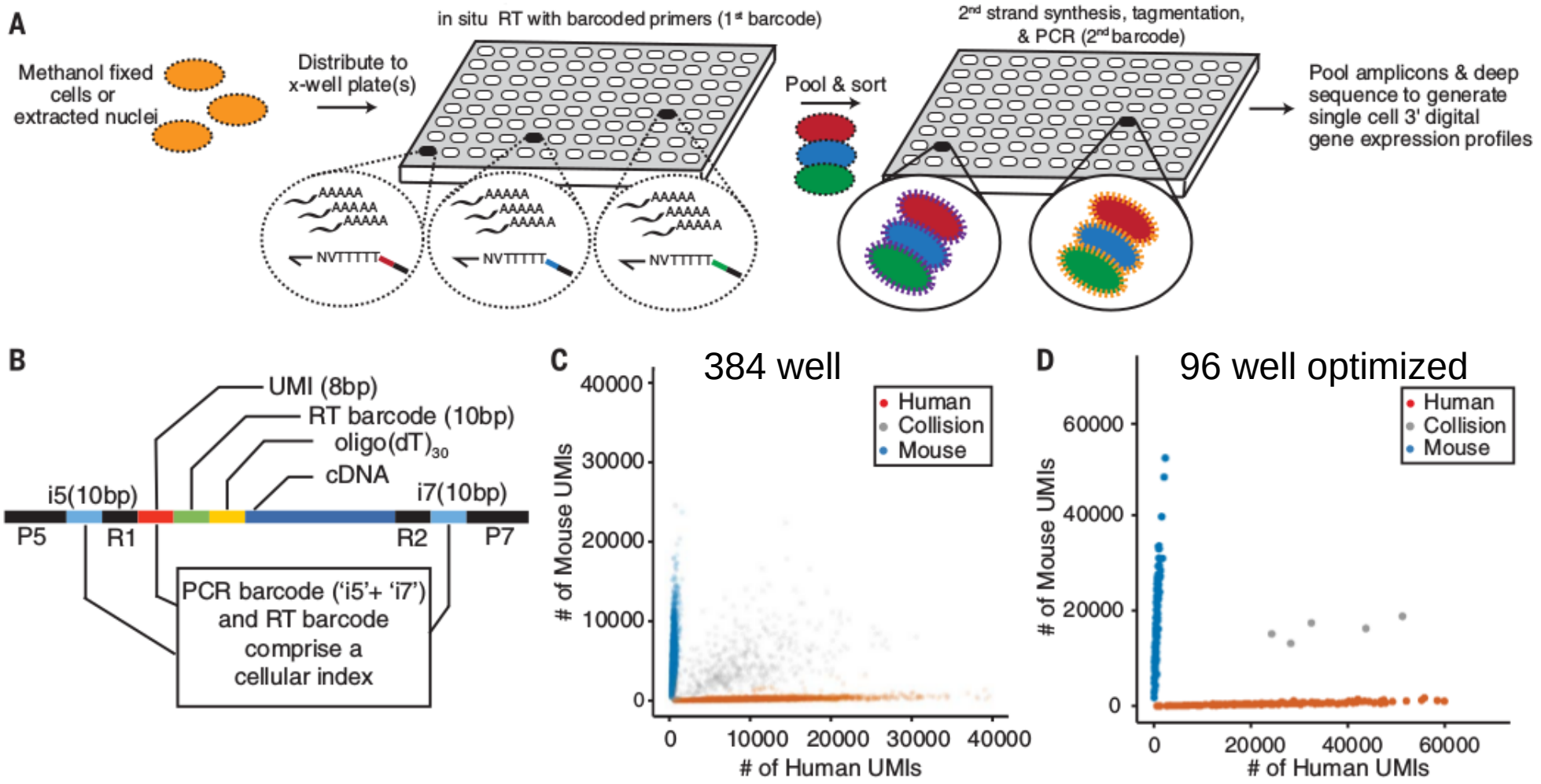
Single cell expression



Each cell has the same genome
Why are cell different?

- gene expression
- physical position (cell-to-cell communication)

Barcoding transcripts from single cells



C & D RNA-seq of a mixture of human mouse cells
UMI = unique molecular identifier
Collision = reads from both species

Single cell dataset

	Cell 1	Cell 2	Cell N
Gene 1			
Gene 2			
Gene N			

Cluster cells of similar type

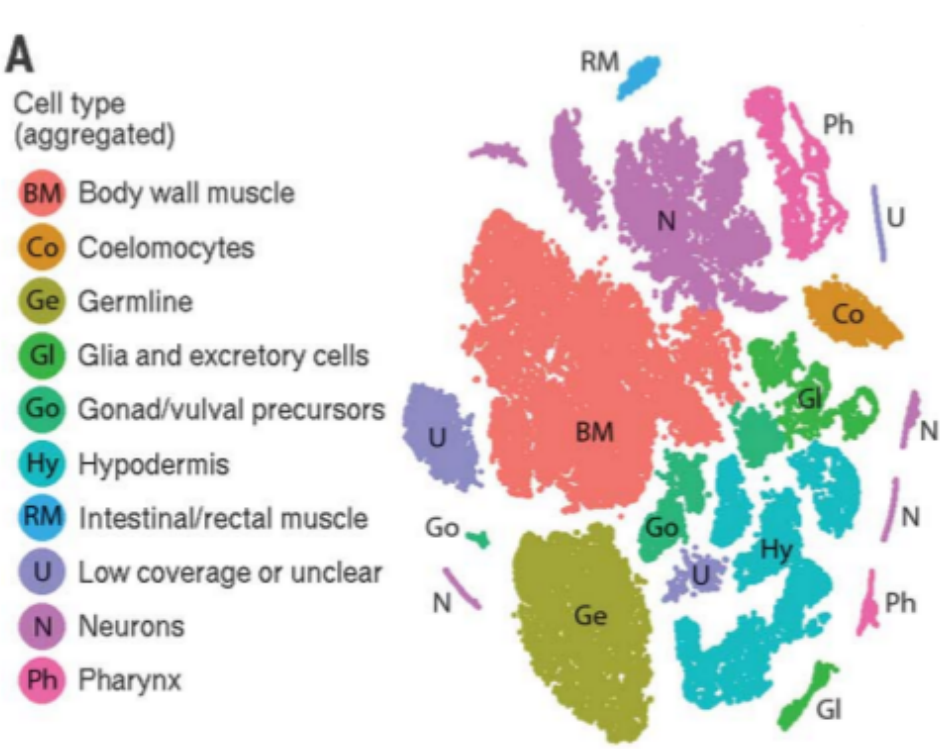
- How many cells of each type
- Expression profile of each cell type
- Labeling clusters (literature)

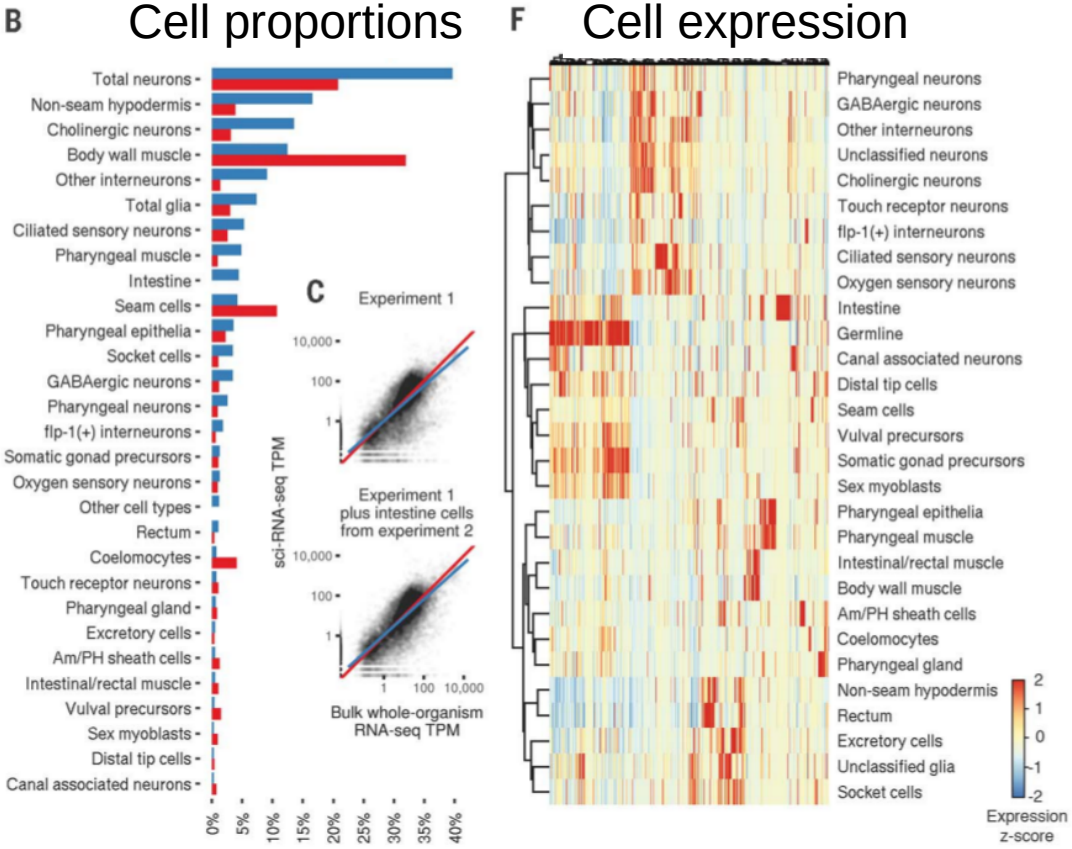
Visualizing single-cell transcript data

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for dimensionality reduction (as is PCA)

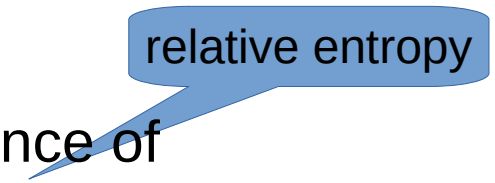
42,035 single cell transcriptomes to 26 cell types with t-SNE





t-SNE algorithm

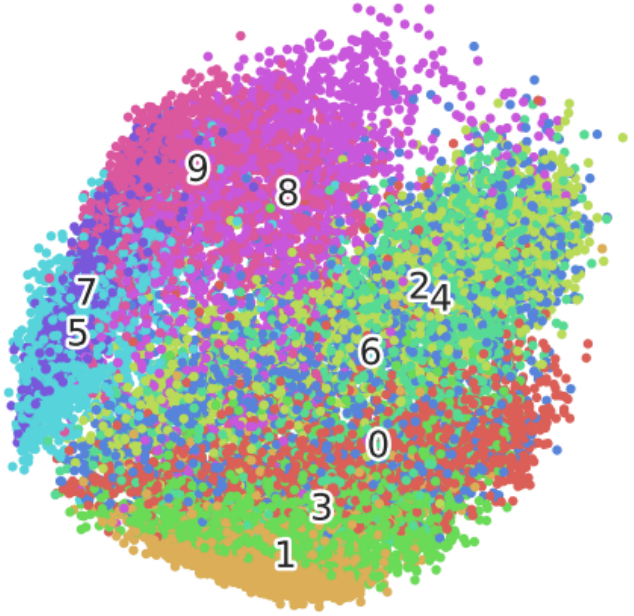
- Starts by calculating the probability of similarity of points in high-dimensional space and calculating the probability of similarity of points in the corresponding low-dimensional space.
- The similarity of points is calculated as the conditional probability that a point A would choose point B as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian (normal distribution) centered at A.
- Then, minimize the difference between these conditional probabilities (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space.
- To measure the minimization of the sum of difference of conditional probability, the sum of Kullback-Leibler divergence of overall data points from a gradient descent method.



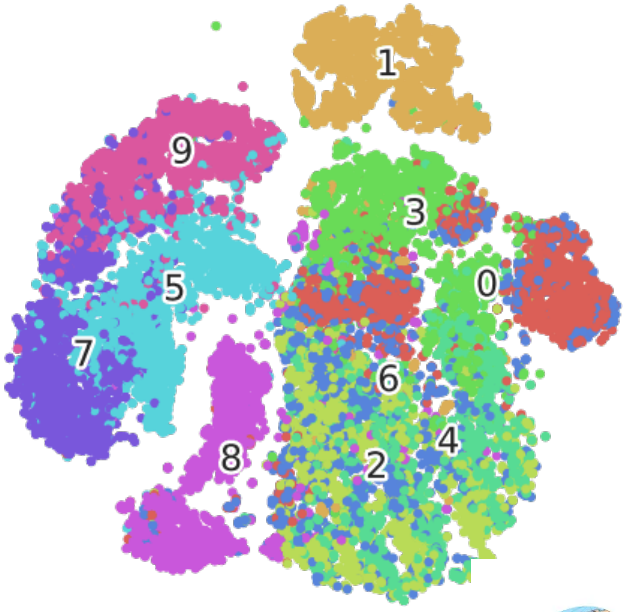
Example

t-SNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.

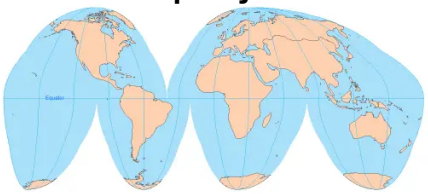
PCA



t-SNE



World map projection



Doesn't PCA account for all variation?
Yes, but we only plot first two PCs

https://www.datacamp.com/community/tutorials/introduction-t-sne

t-SNE vs PCA

- Time: t-SNE is computationally intensive (hours), PCA is quick (minutes)
- Model: t-SNE is probabilistic (stochastic), PCA is mathematical (deterministic)
- Objective: PCA places dissimilar data points far apart in a lower dimension representation. t-SNE also places similar points close together in the lower dimension.
- Functions: PCA uses linear functions, t-SNE uses polynomial
- Output: PCs can be used for inference (PCs are linear combinations of variables), but for t-SNE the input features are no longer identifiable, and you cannot make any inference based only on the output of t-SNE (visualize only)

Supervised machine learning

- Decision trees
- Random forests
- Linear discriminant analysis
- Support vector machine

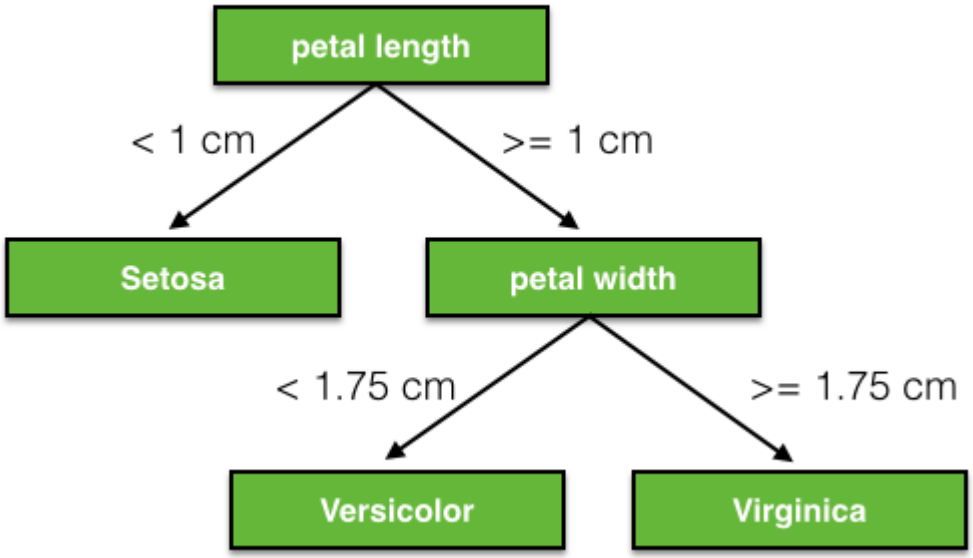
Decision trees (supervised)

Decision trees (categorical predictions)
Regression trees (continuous predictions)

Each interior node corresponds to one of the input variables
Edges to children for each of the possible output values
* There can be multiple children/value and multiple nodes/input

Fisher's *Iris* Data [hide]

Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.3	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*



Decision tree algorithm

Top down recursion (greedy algorithm)

Variable selection at each node: Gini impurity, Information gain

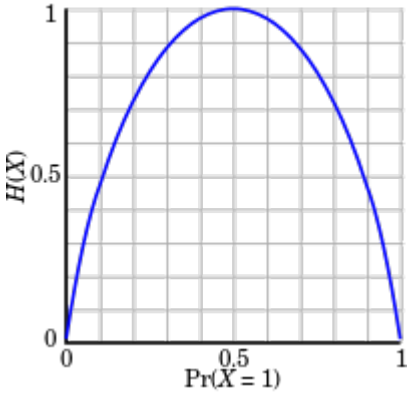
Entropy = uncertainty

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_a p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

Essentially choose the split that results in the purest (low entropy) daughter nodes

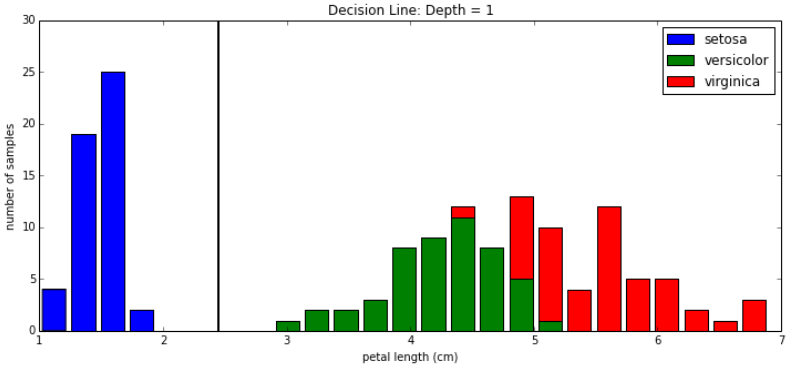
Entropy (H) is a measure uncertainty



Bernoulli process

Max H when p = 0.5

- Higher H = more information gain by knowing outcome
- H = expected value of information content
- IG tells us how much information we gain about T by knowing a



Knowing petal length (a) tells us a lot about species (T)

Entropy example

Given 7 samples; 4 with cancer (C) and 3 with no cancer (NC)
Which mutation tells us the most about whether a sample is cancer or not?
Or - which mutation most information in the first step of a decision tree

Samples	Mutation 1	Mutation 2	Mutation 3	Mutation 4
C1	1	1	1	0
C2	1	1	0	1
C3	1	0	1	1
C4	0	1	1	0
NC1	0	0	0	0
NC2	0	1	0	0
NC3	1	1	0	0

Entropy gain for mutation 1:
IG(T,a) = H(T) - H(T|a) = 0.98-0.857
IG(T,a) = 0.128

Mutation 1: 0.128
Mutation 2: 0.006
Mutation 3: 0.522 (best)
Mutation 4: 0.292

Total entropy of sample:
$H(T) = - \text{sum}(p(x) \log_2(px))$
$H(T) = -(4/7 \times \log2(4/7) + 3/7 \times \log2(3/7)) = 0.985$

Entropy of children (a = mutation 1):
$H(T \mid a=1) =$
$-(3/4 \times \log2(3/4) + 1/4 \times \log2(1/4)) = 0.811$

$H(T \mid a=0) =$
$-(1/3 \times \log2(1/3) + 2/3 \times \log2(2/3)) = 0.918$

Average of children:
$(4/7 \times 0.811) + (3/7 \times 0.918) = 0.857$

Decision tree algorithm

Top down recursion (greedy algorithm)

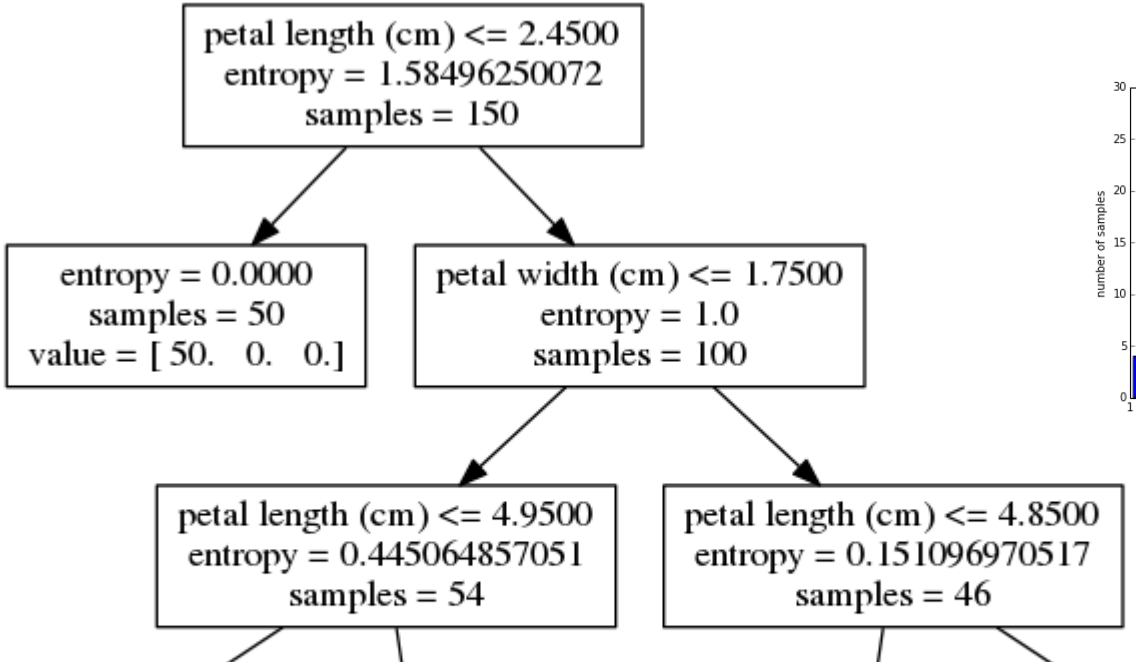
Variable selection at each node: Gini impurity, Information gain

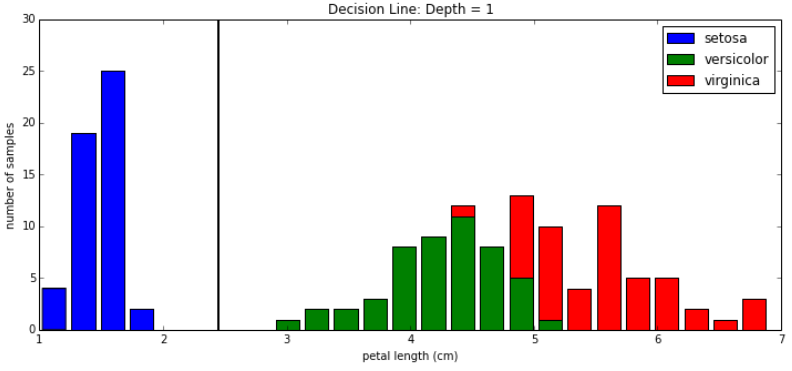
Entropy = uncertainty

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{a} p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

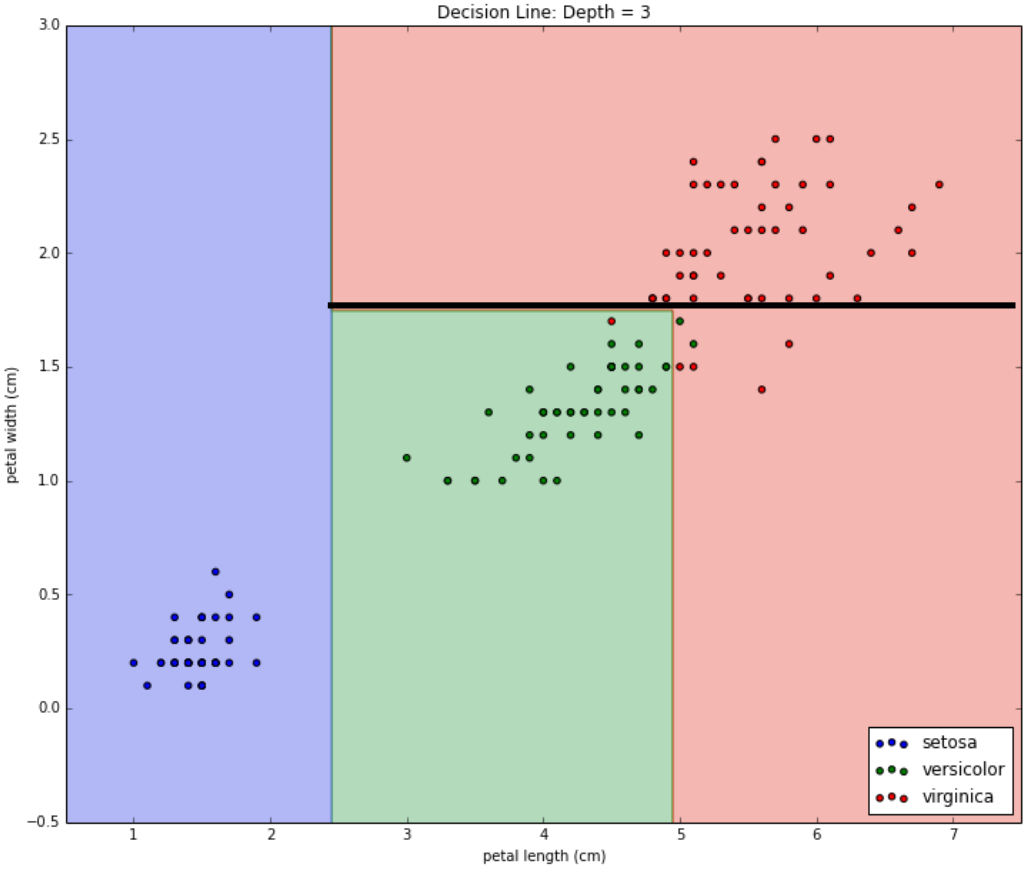
Essentially choose the split that results in the purest (low entropy) daughter nodes

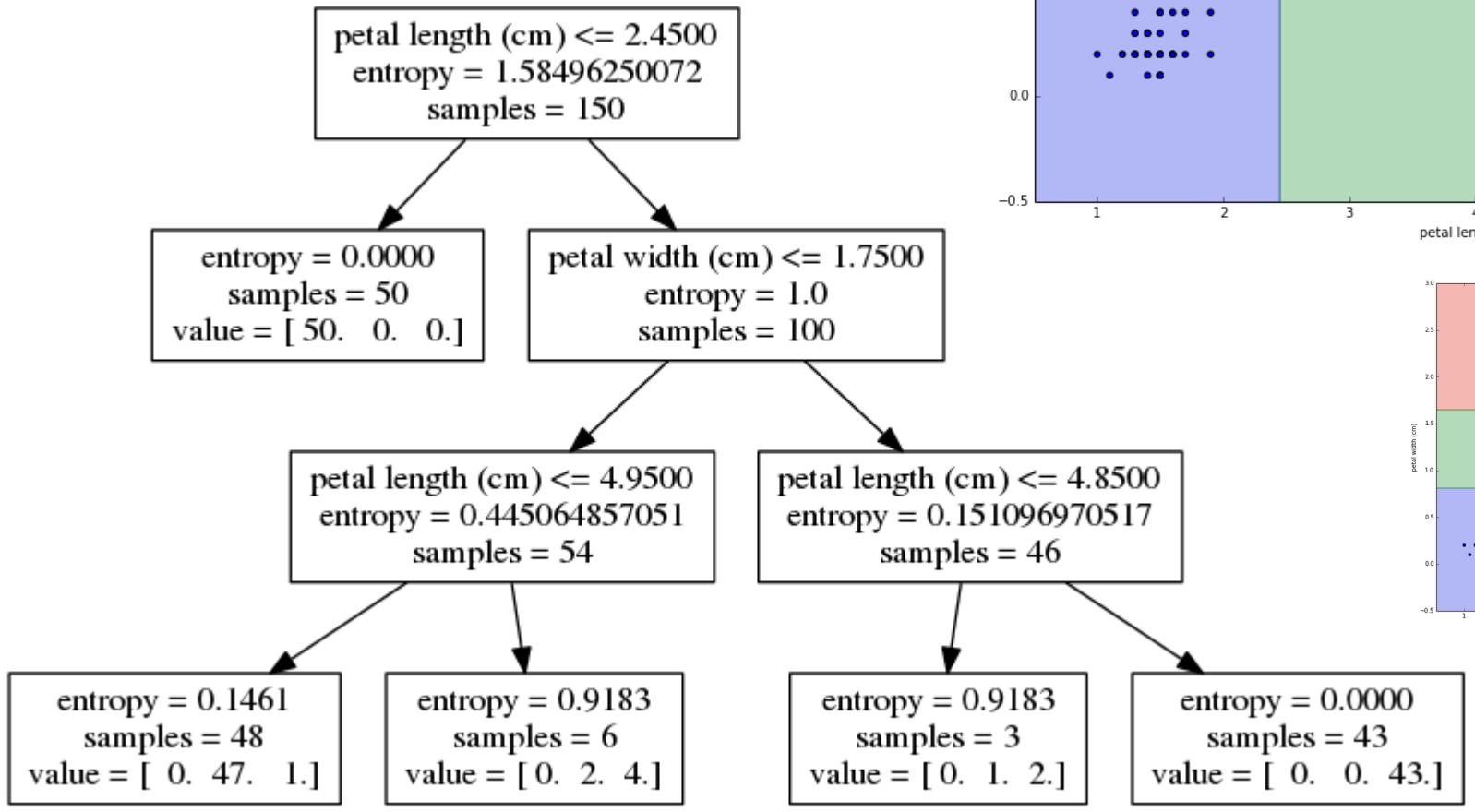


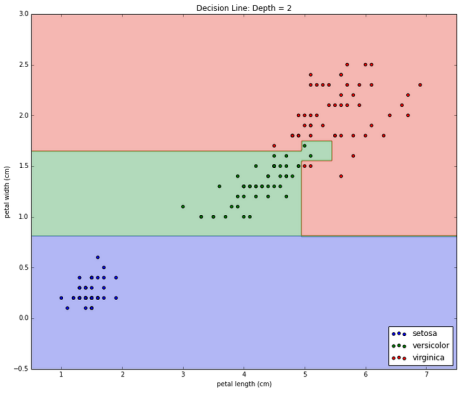


Decision tree

- Advantage: Simple to understand with both categorical and continuous data
- Disadvantage: not robust, sensitive to training data, overfitting (perfect classification is possible)

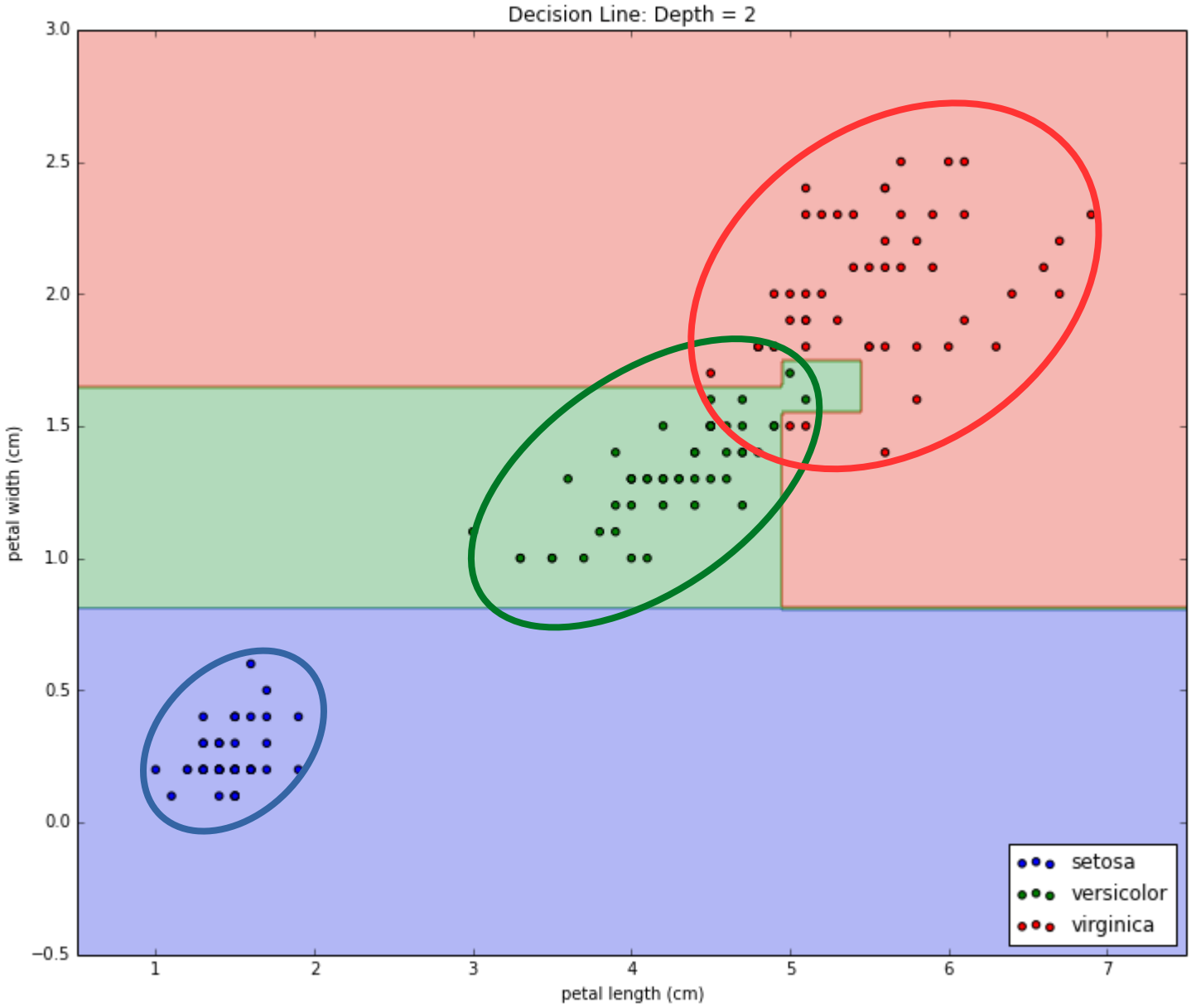






Over-fitting

Circles = truth

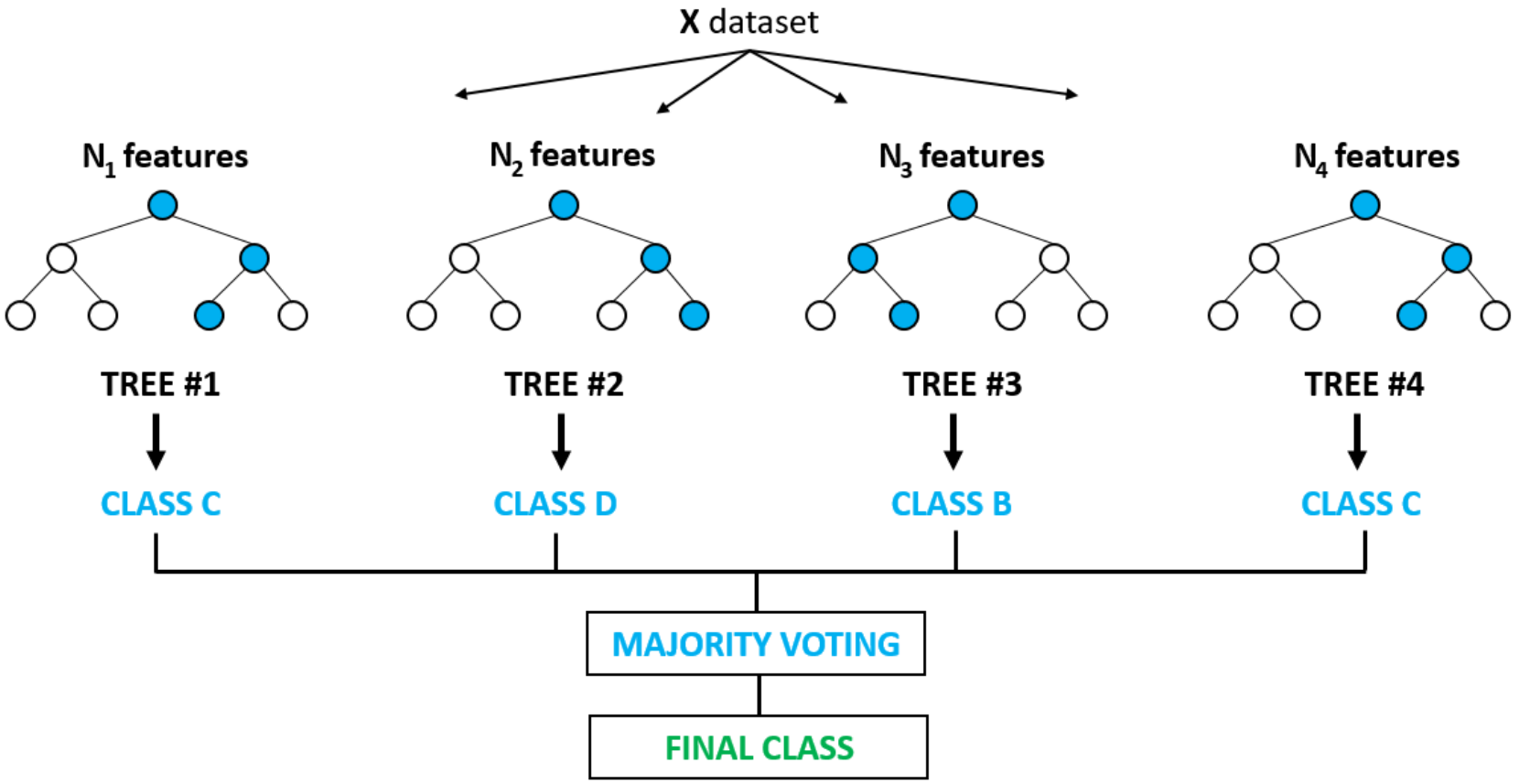


Random Forest (supervised)

An ensemble learning method for classification or regression that operates by constructing a multitude of decision trees.

- Output is the mode (class) or mean (regression) of the individual trees.
- Avoids overfitting and sensitivity to training data.

- randomized node optimization
- Uses out-of-bag error as an estimate of the generalization error.
- Measures variable importance through permutation.



- Select random sample of **features** with replacement, fit trees
- Average predictions or take **majority rule**
- **Feature bagging** decreases variance without causing bias
- A new sample is evaluated by ensemble of trees (majority or posterior probability)
- Disadvantage: no single easily interpreted tree
- To measure the **importance** of the j-th feature after training, the values of the j-th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set to estimate its importance.
  NB: closely related to the k-nearest neighbor algorithm (k-NN)

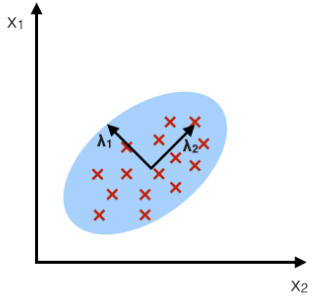
Linear discriminant analysis (LDA)

LDA finds a linear combination of features that characterizes or separates two or more classes

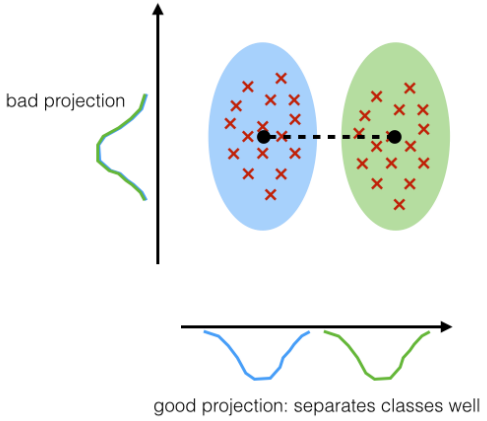
LDA is closely related to logistic regression in that it predicts a label based on continuous data.

LDA is also closely related to PCA in that they both look for linear combinations of variables which best explain the data. LDA uses labels to find the best combination of variables.

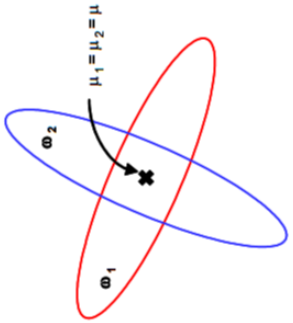
PCA:
component axes that maximize the variance



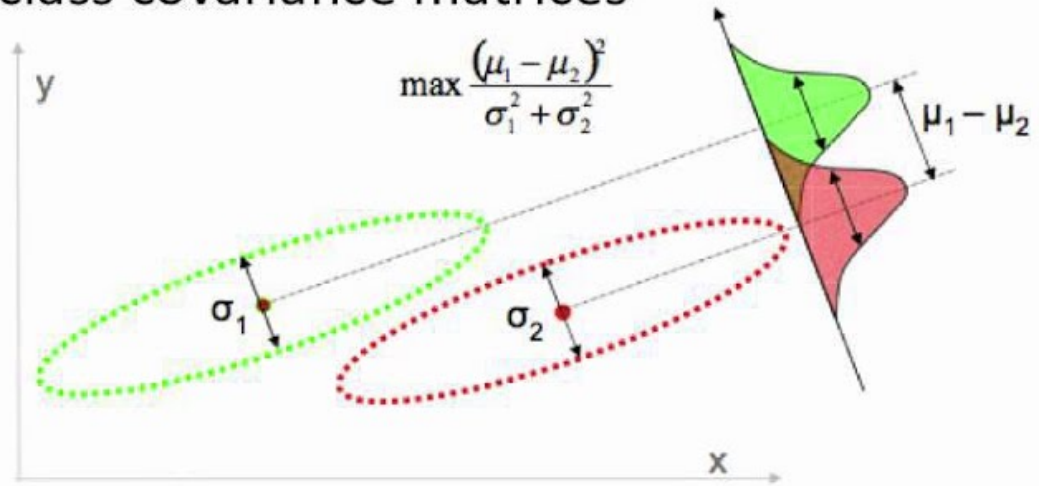
LDA:
maximizing the component axes for class-separation

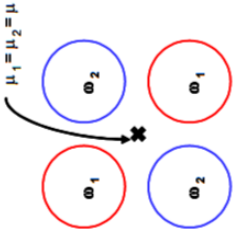


Linear Discriminant Analysis



- LDA: pick a new dimension that gives:
  - maximum separation between means of projected classes
  - minimum variance within each projected class
- Solution: eigenvectors based on between-class and within-class covariance matrices





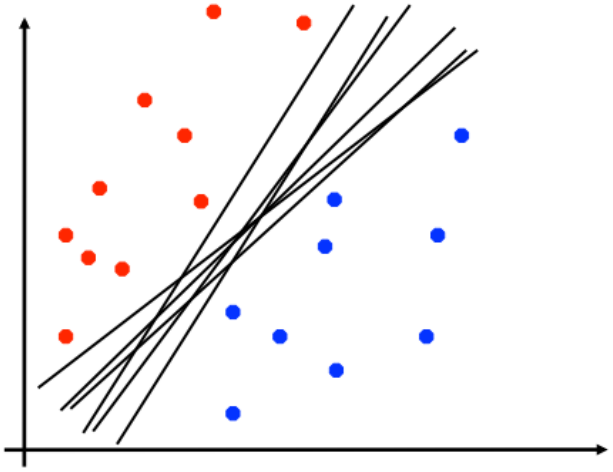
- Discriminant analysis works by creating one or more linear combinations of predictors, creating a new latent variable for each function.
- The first function created maximizes the differences between groups on that function, then the second function also does this, but can't be correlated with the first. Disadvantage: overfitting.

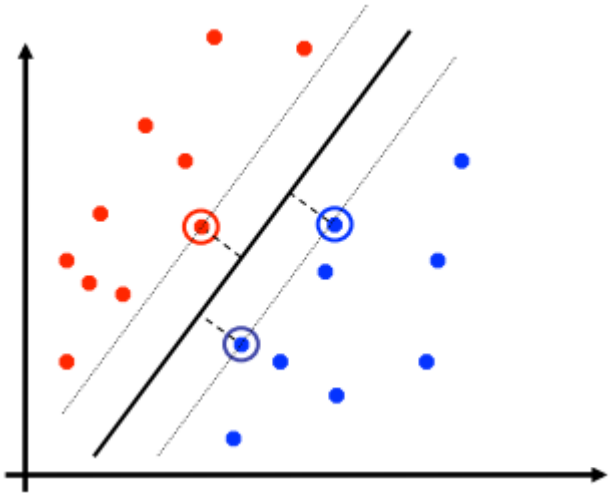
Support Vector Machines (SVMs)

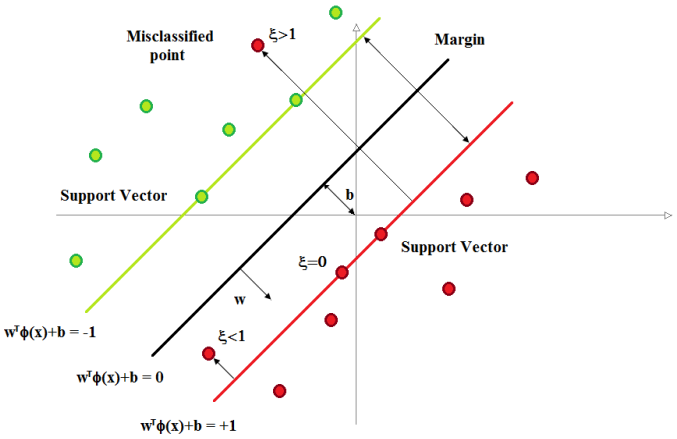
SVMs are supervised learning models with associated learning algorithms used for classification and regression analysis.

The algorithm builds a non-probabilistic binary linear classifier.

How to optimize linear classifier?



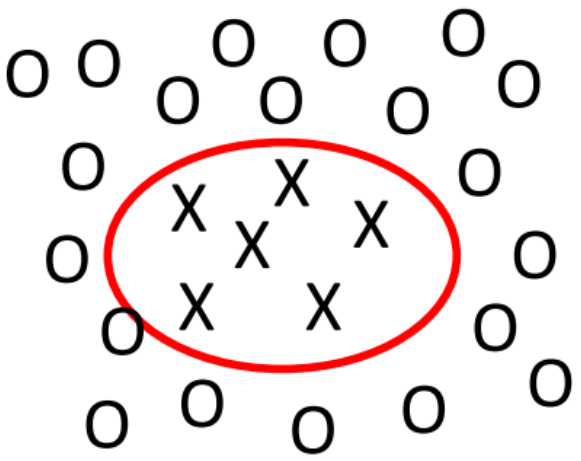




- Maximize the margin: separate categories are divided by a clear gap that is as wide as possible.
- Minimize number of misclassifications
- This can be done with quadratic programming

Problem: Non-linear surfaces
Solution: kernel methods

SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.



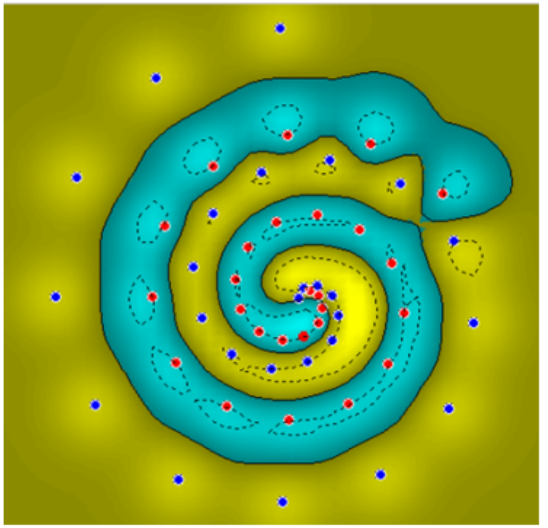
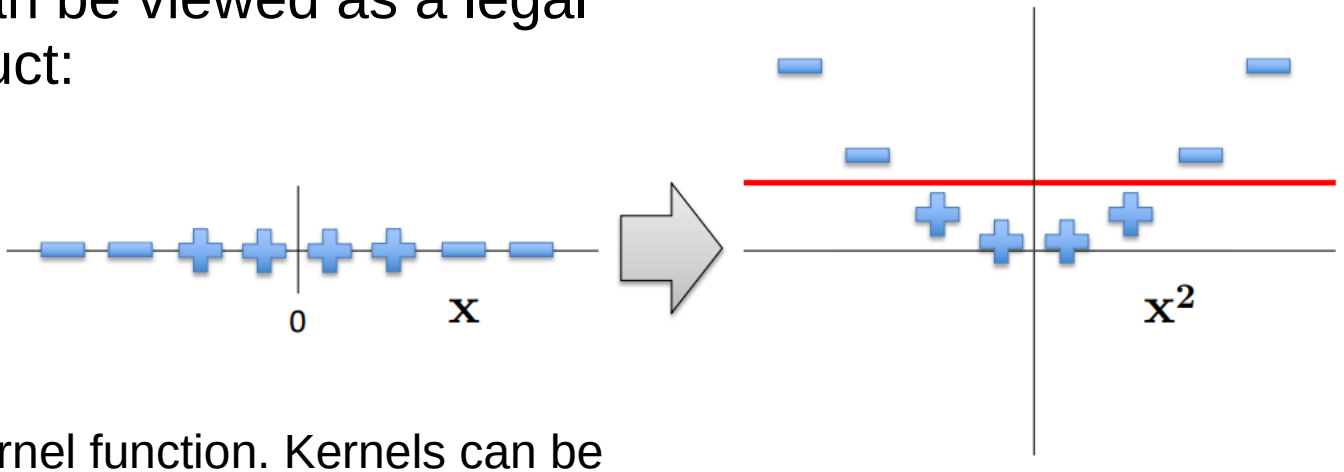


Image from http://www.atrandomresearch.com/iclass/

K(x,z) is a kernel if it can be viewed as a legal definition of inner product:
K(x,z) = Φ(x) · Φ(z)

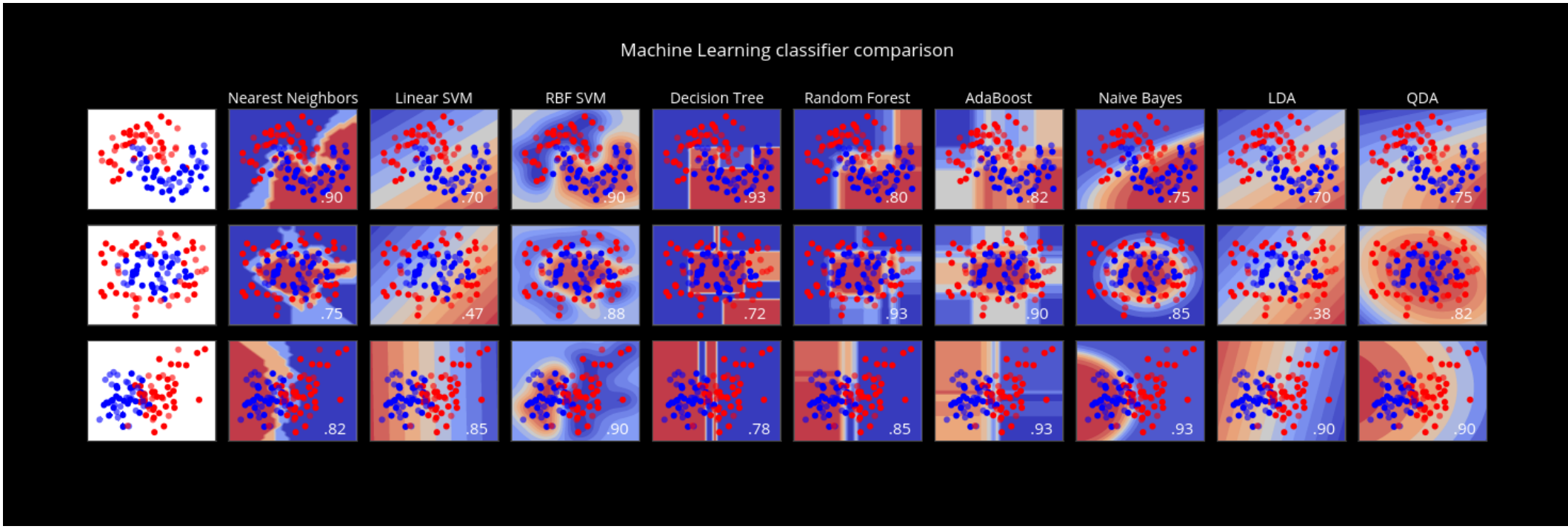
Apply SVM to Kernel



Dot products replaced by kernel function. Kernels can be linear, quadratic, Gaussian, etc, and can be used beyond SVM.

$$k(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j})^d$$

Comparing classifiers



Exercises

1) What does PCA maximize?

2) Which machine learning methods could be used to classify single cell transcriptome data? Which could be used to visualize? [kmeans, t-SNE, PCA, hierarchical clustering]

3) The output of t-SNE can be used for statistical inferrence [T/F]
4) What is the advantage of random forest over decision trees?

5) ___maximizes variance explained whereas ___maximizes separation between classes [LDA/PCA]
6) Which of these are supervised machine learning methods: K-means, PCA, LDA, SVM, Decision tree, Random Forest, t-SNE
7) What are kernels used to accomplish in support vector machines?

Exercises

1) Calculate information gain for each variable, which variable provides the most information gain?

Variable 1	Mutation	No mutation
Cancer	23	10
No Cancer	3	15

Variable 2	High expression	Low expression
Cancer	5	28
No Cancer	13	5