Neural Networks

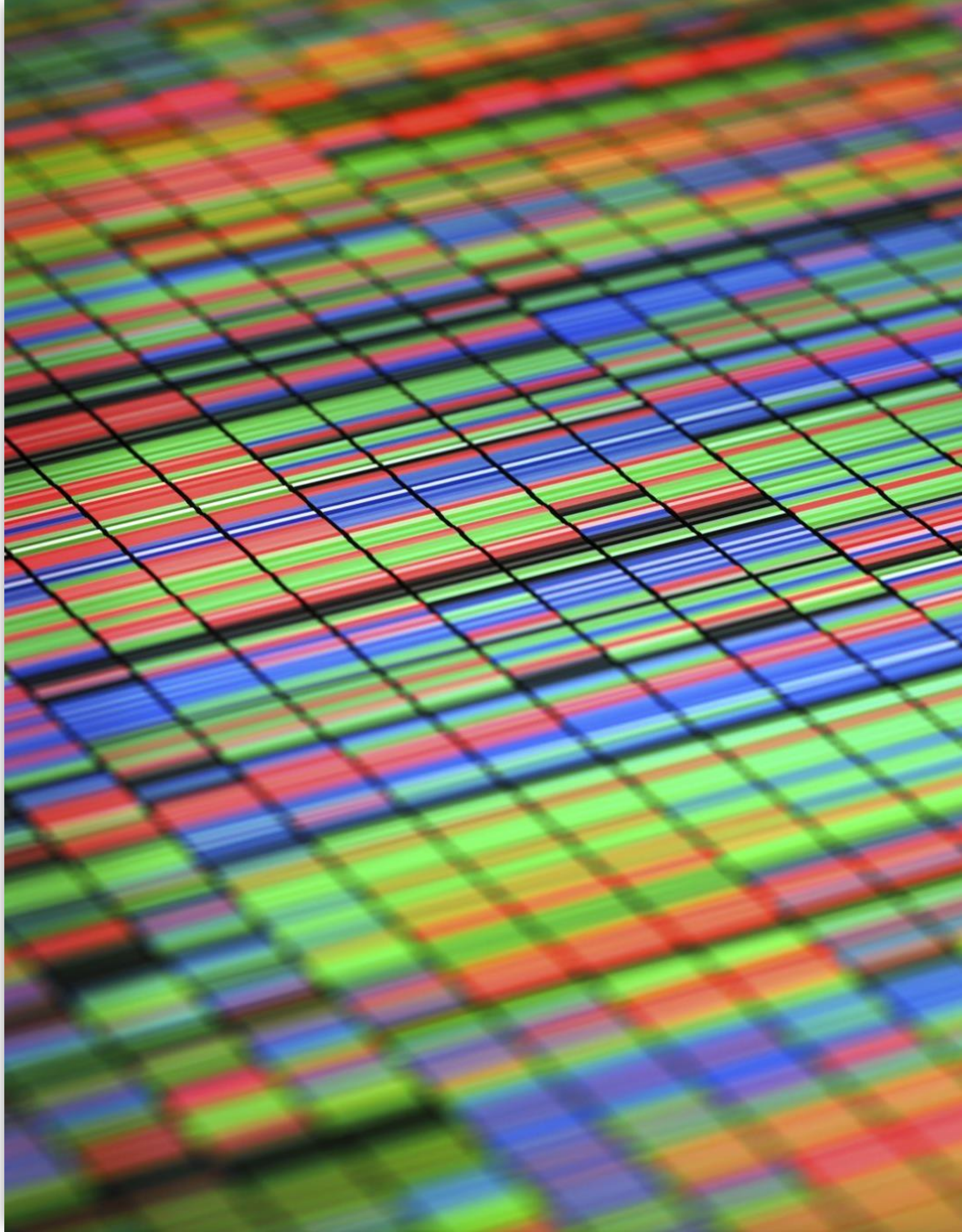
Data Visualization and PCA

CSCI 4850/5850

Pattern Recognition

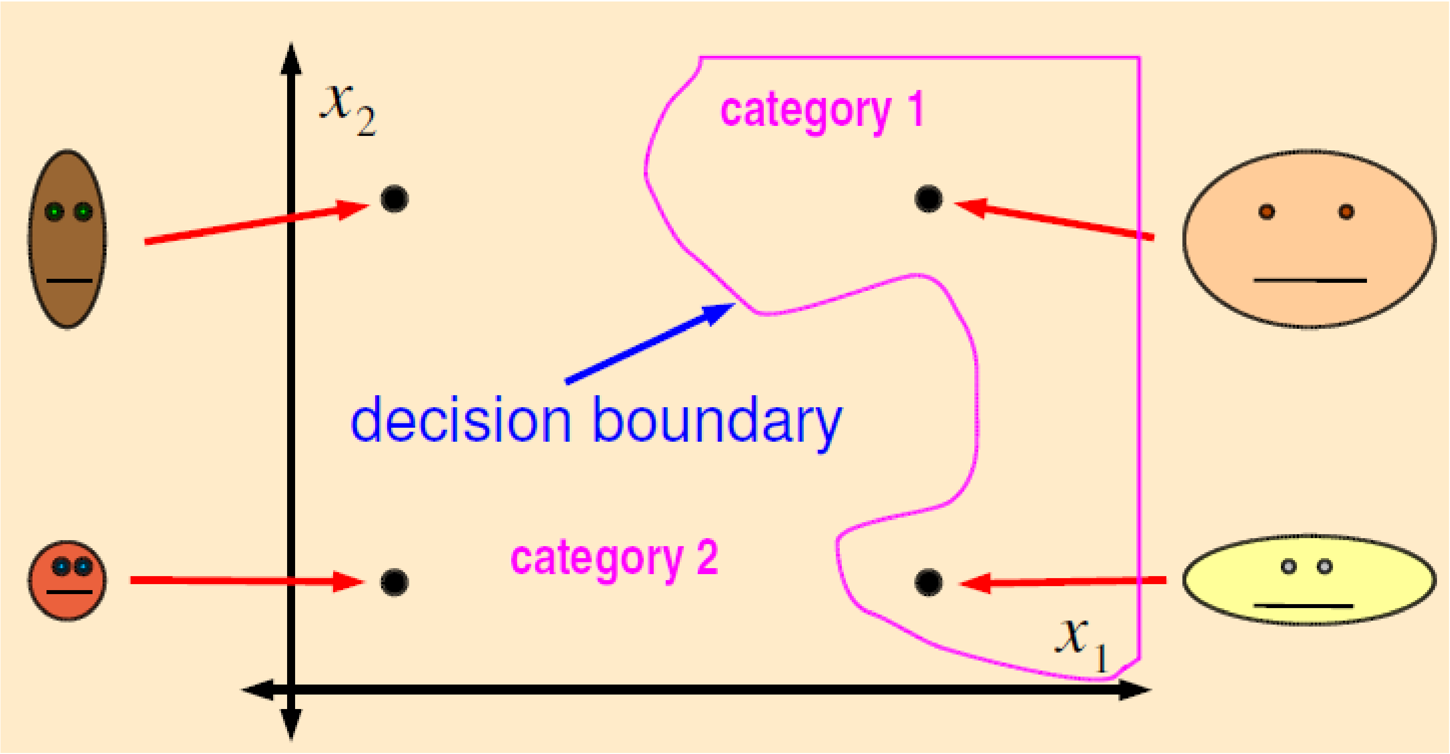
- The ***classification problem*** involves assigning one of a discrete set of **category labels** to an **object**, based on a collection of measurements (i.e **features**) of that object.

- The **statistical pattern recognition problem** involves finding a function which does a good job of classifying objects, based on a combination of prior knowledge and a **data set** of labeled objects (i.e. **feature vectors**).
  - Goodness is formalized in terms of a performance measure, typically a **loss/error function**.
  - The main goal is **generalization**: good classification performance on objects ***not*** used to find the classification function (i.e. "held-out" feature vectors).

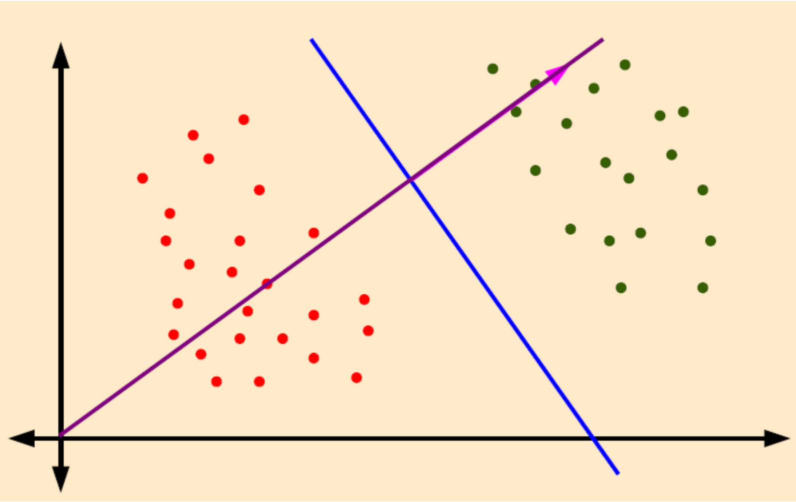


Features Reside in a Vector Space

- The vector space in which the features reside is often called the **input space** or object **feature space**. Finding a *classification function* can be thought of as assigning category labels to ***regions*** in this space.



Classification is Related to Projection

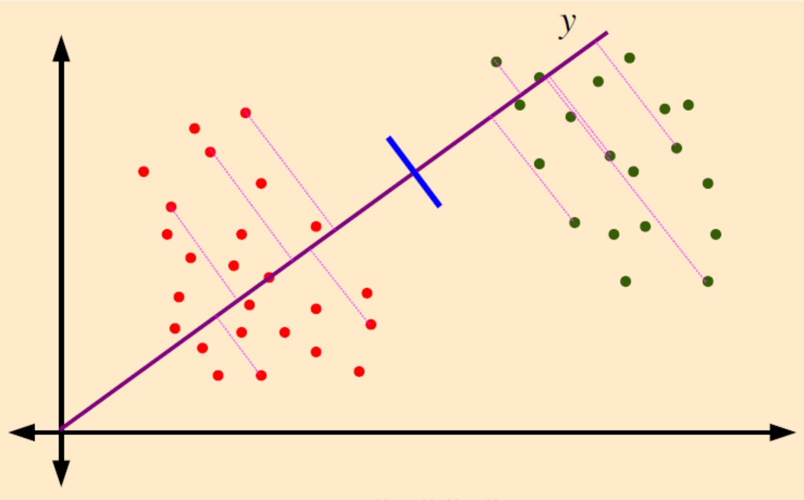


Separating regions involves finding the decision boundary between data points with different class labels

Individual pattern details are lost

Similarly, a regression can result in a projection along a vector where a decision boundary is plain to see

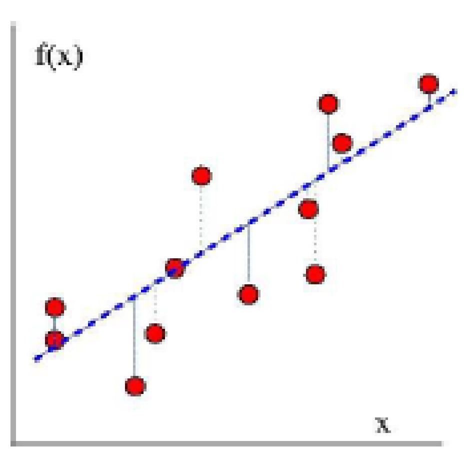
Individual pattern details are lost



Function Approximation

- If we would like to use the same techniques to study functions with continuous outputs (unlike the discrete outputs of the classification problem), then we are solving the **function approximation** or **regression** problem.

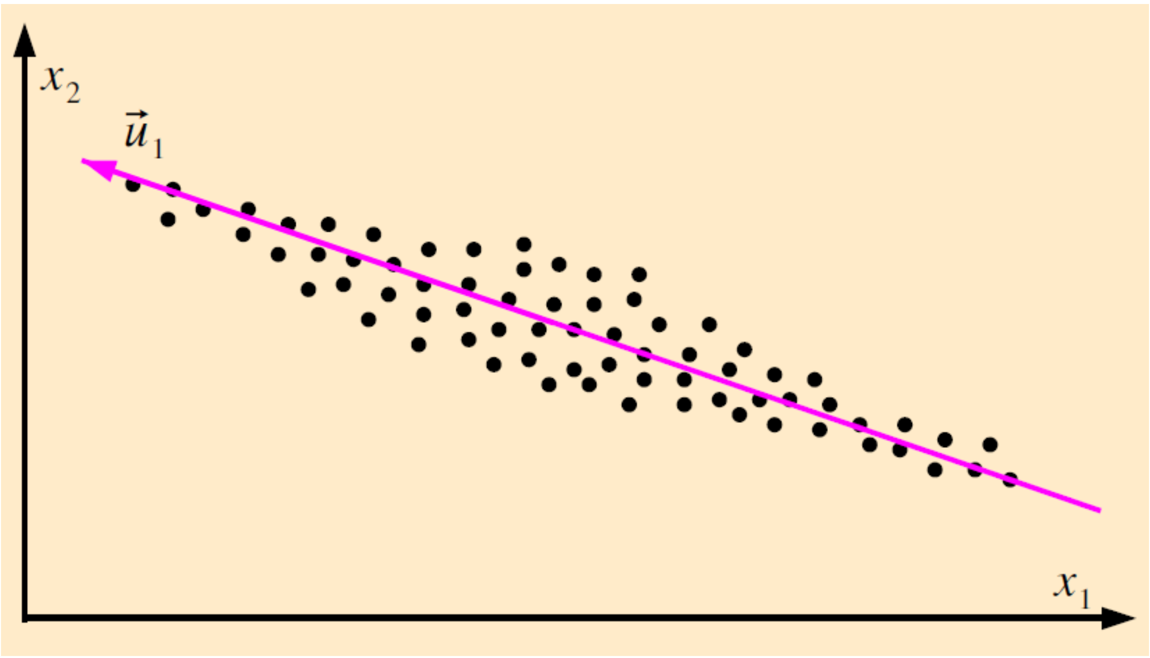
$$\begin{aligned} \vec{x} &\rightarrow \vec{y} \\ \Re^d &\rightarrow \Re^c \\ y_k &= f_k(\vec{x}) \end{aligned}$$

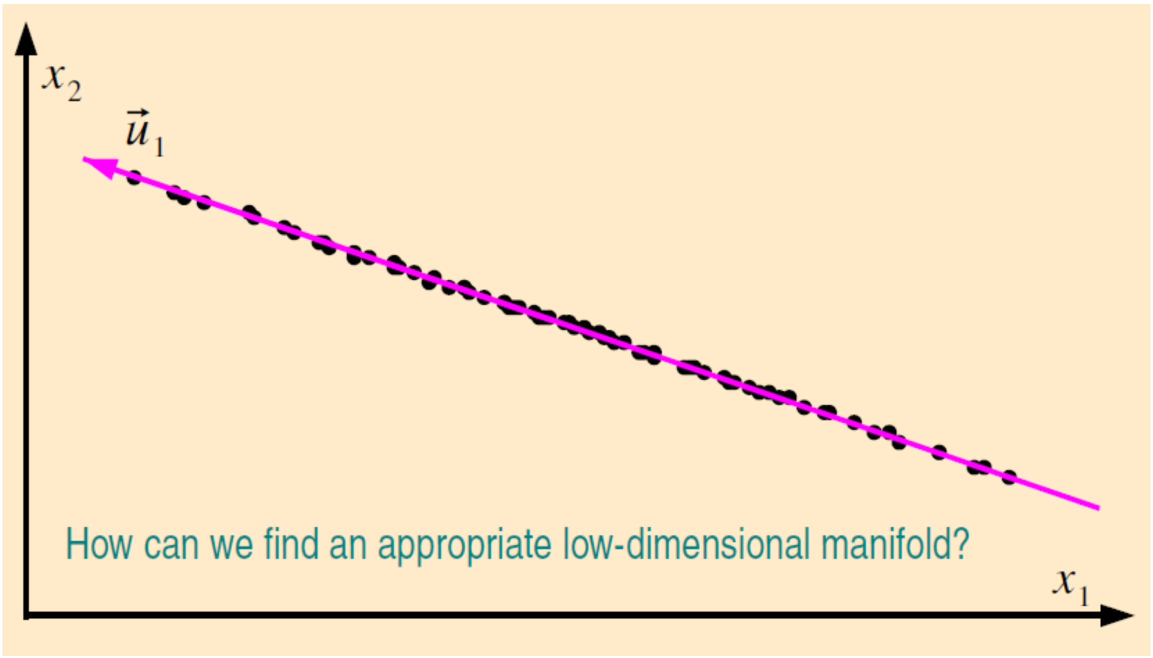


$$y_k = y_k(\vec{x}\ ;\ \vec{w})$$

Low-dimensional Projection

- Generalization involves *throwing information away*
- In many cases, we can visualize this process as **projecting** out variance(s) in the data that doesn't help us generalize



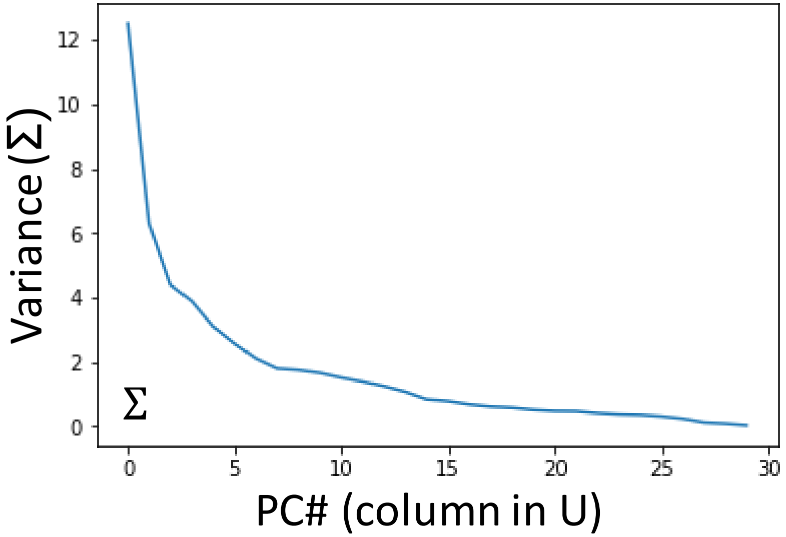


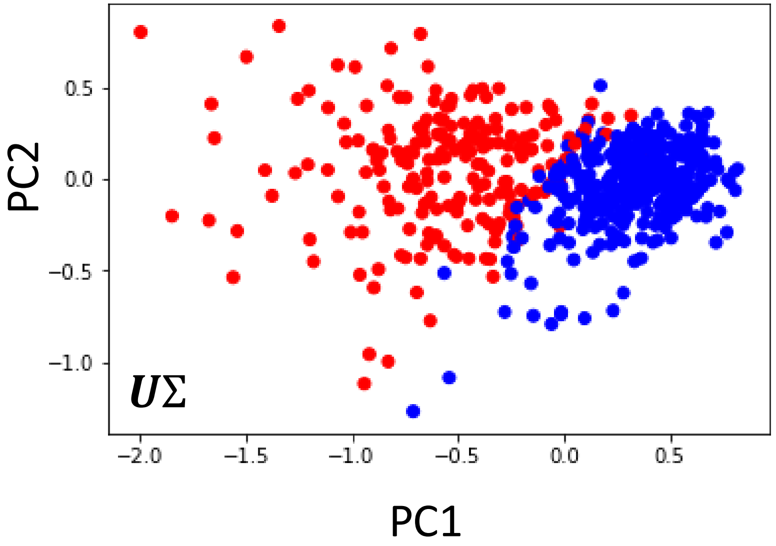
Principal Component Analysis

- In future lectures we will see how *high-dimensional* projections make hard problems easier to solve, but *low-dimensional* projections help with **generalization**
  - In some sense, this is the “throwing information away for learning” idea that we discussed before
- Normally, we perform PCA **after** we have *standardized* our data
  - PCA is sensitive to large magnitude features
  - We won’t perform any standardization in the lab assignment because:
    - The Iris/MNIST data sets (see the lab) contains similar kinds of features
    - I already did that for you on the cancer data we were using
    - We will study standardization in future lectures

How do we do that?

- Singular value decomposition (rectangular)
  - The singular value decomposition of a matrix is: $$\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}$$
  - Where **Σ** is a rectangular diagonal matrix of singular values, and **U** and **V** are the left-singular and right-singular vectors of **A**, respectively.

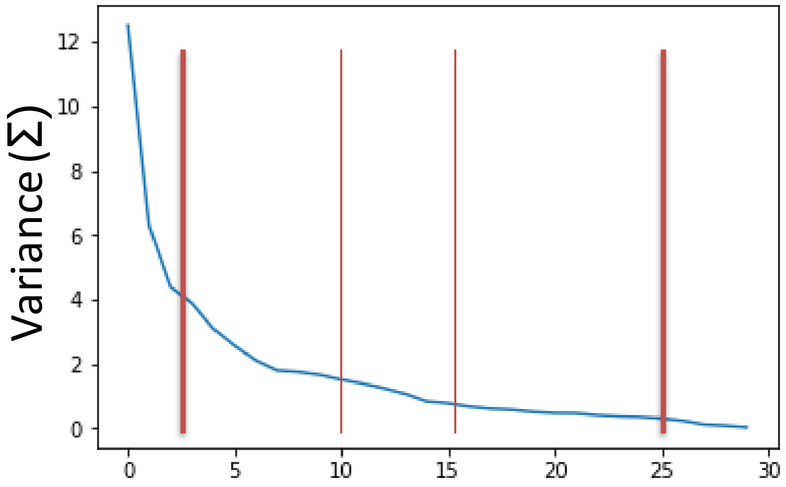




What is this used for?

- Most common uses of PCA:
  - Visualization (top 2-3)
  - Noise Reduction (bottom)
  - Dimensionality Reduction (tradeoff efficiency)
- We only used the first two in the homework because we were interested only in visualization…
- Removing noise may (or may not) improve generalization…
- Reducing dimensions may (or may not) improve generalization or training time…

$\boldsymbol{U}\Sigma$



Visual

Reduce?

Noise

Remember to run PCA on your *entire data set* before transforming it into train/validation/test sets!