• May 1, **CODY**, Error Backpropagation, Bishop 5.3, and Support Vector Machines (SVM) Bishop Ch 7.
• May 3, **Class HW** SVM, PCA, and K-means, Bishop Ch 12.1, 9.1
• May 8, **CODY** Machine Learning for finding oil, focusing on 1) robust seismic denoising/interpolation using structured matrix approximation 2) seismic image clustering and classification, using t-SNE(t-distributed stochastic neighbor embedding) and CNN. **Weichang Li**, Goup Leader Aramco, Houston.
• May 10, **Class HW** **First distribution of final projects**. Bishop Ch 9
• May 15, **CODY** Seismology and Machine Learning, **Daniel Trugman** (half class), ch 9
• May 17, **Class HW** Ocean acoustic source tracking. Final projects. The main goal in the last 3 weeks is the Final project. ch 9
• May 22, Dictionary learning, **Mike Bianco** (half class), Graphical models Bishop Ch 8
• May 24, Graphical models Bishop Ch 8
• May 31, **No Class Workshop**, **Big Data and The Earth Sciences: Grand Challenges Workshop**
• June 5, Discuss workshop. Spiess Hall open for project discussion 11am-
• June 7, **Workshop report. No class**
• **June 12** Spiess Hall open for project discussion 9-11:30am and 2-7pm
• June 16 Final report delivered. Beer time
SVMs are Perceptrons!

• SVM’s use each training case, $x$, to define a feature $K(x, .)$ where $K$ is user chosen.
  – So the user designs the features.

• SVM do “feature selection” by picking support vectors, and learn feature weighting from a big optimization problem.

• So an SVM is a clever way to train a standard perceptron.
  – All that a perceptron cannot do, cannot be done by SVM’s (but it’s a long time since 1969 so people have forgotten this).

• SVM DOES:
  – Margin maximization
  – Kernel trick
  – Sparse
SVM summarized--- Only kernels

• Minimize with respect to $\mathbf{w}, w_0$

$$C \sum_n^N \zeta_n + \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{(Bishop 7.21)}$$

• Solution found in dual domain with Lagrange multipliers
  
  $- a_n, n = 1 \cdots N$ and

• This gives the support vectors $S$

$$\hat{\mathbf{w}} = \sum_{n \in S} a_n t_n \phi(x_n) \quad \text{(Bishop 7.8)}$$

• Used for predictions

$$\hat{y} = w_0 + \mathbf{w}^T \phi(x) = w_0 + \sum_{n \in S} a_n t_n \phi(x_n)^T \phi(x)$$

$$= w_0 + \sum_{n \in S} a_n t_n k(x_n, x) \quad \text{(Bishop 7.13)}$$
case 'Classify'
    % train
    model = svmtrain(Y, X,['-c 7.46 -g ' gamma ' -q ' kernel]);
    % predict
    [predict_label,~,~] = svmpredict(rand([length(Y),1]), X, model,'-q');

>> model
    model = struct with fields:
    Parameters: [5×1 double]
    nr_class: 2
    totalSV: 36
    rho: 8.3220
    Label: [2×1 double]
    sv_indices: [36×1 double]
    ProbA: []      ProbB: []
    nSV: [2×1 double]
    sv_coef: [36×1 double]
    SVs: [36×2 double]
Can be inner product in \textbf{infinite} dimensional space

Assume \(x \in \mathbb{R}^1\) and \(\gamma > 0\).

\[
e^{-\gamma \|x_i - x_j\|^2} = e^{-\gamma (x_i - x_j)^2} = e^{-\gamma x_i^2 + 2\gamma x_i x_j - \gamma x_j^2}
\]

\[
e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 + \frac{2\gamma x_i x_j}{1!} + \frac{(2\gamma x_i x_j)^2}{2!} + \frac{(2\gamma x_i x_j)^3}{3!} + \cdots\right)
\]

\[
e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}} x_i \cdot \sqrt{\frac{2\gamma}{1!}} x_j + \sqrt{\frac{(2\gamma)^2}{2!}} x_i^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}} x_j^2
\]

\[
+ \sqrt{\frac{(2\gamma)^3}{3!}} x_i^3 \cdot \sqrt{\frac{(2\gamma)^3}{3!}} x_j^3 + \cdots\right) = \phi(x_i)^T \phi(x_j),
\]

where

\[
\phi(x) = e^{-\gamma x^2} \left[1, \sqrt{\frac{2\gamma}{1!}} x, \sqrt{\frac{(2\gamma)^2}{2!}} x^2, \sqrt{\frac{(2\gamma)^3}{3!}} x^3, \cdots\right]^T.
\]
Finding the Decision Function

- **w**: maybe *infinite* variables
- The dual problem

\[
\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\
\text{subject to} \quad 0 \leq \alpha_i \leq C, \ i = 1, \ldots, l \\
y^T \alpha = 0,
\]

where \( Q_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) \) and \( e = [1, \ldots, 1]^T \)

- At optimum

\[
w = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i)
\]

- A *finite* problem: #variables = #training data

Using these results to eliminate \( w, b, \) and \( \{\xi_n\} \) from the Lagrangian, we obtain the dual Lagrangian in the form

\[
\tilde{L}(\alpha) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \quad (7.32)
\]
Unsupervised vs Supervised Learning:

- Most of this course focuses on *supervised learning* methods such as regression and classification.

- In that setting we observe both a set of features $X_1, X_2, \ldots, X_p$ for each object, as well as a response or outcome variable $Y$. The goal is then to predict $Y$ using $X_1, X_2, \ldots, X_p$.

- Here we instead focus on *unsupervised learning*, where we observe only the features $X_1, X_2, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable $Y$. 
The Goals of Unsupervised Learning

• The goal is to discover interesting things about the measurements: is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?
• We discuss two methods:
  • *principal components analysis*, a tool used for data visualization or data pre-processing before supervised techniques are applied, and
  • *clustering*, a broad class of methods for discovering unknown subgroups in data.
Challenge of unsupervised learning

- Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response.
- But techniques for unsupervised learning are of growing importance in a number of fields:
  - subgroups of breast cancer patients grouped by their gene expression measurements,
  - groups of shoppers characterized by their browsing and purchase histories,
  - movies grouped by the ratings assigned by movie viewers.

However, it is often easier to work with unlabeled data.
- Think of the movie rating
- Past science history
Principal Components Analysis

Dimensionality reduction
Data-compression
less storage and easy learning
Data visualization

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.

- Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.
PCA don’t work well for classification
Principal Components Analysis: details

- The *first principal component* of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$ Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p $$

that has the largest variance. By *normalized*, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11} \ \phi_{21} \ \ldots \ \phi_{p1})^T$.

- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.
Computation of Principal Components

• Suppose we have a $n \times p$ data set $X$. Since we are only interested in variance, we assume that each of the variables in $X$ has been centered to have mean zero (that is, the column means of $X$ are zero).

• We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip} \quad (1)$$

for $i = 1, \ldots, n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

• Since each of the $x_{ij}$ has mean zero, then so does $z_{i1}$ (for any values of $\phi_{j1}$). Hence the sample variance of the $z_{i1}$ can be written as $\frac{1}{n} \sum_{i=1}^{n} z_{i1}^2$. 
Computation: continued

- Plugging in (1) the first principal component loading vector solves the optimization problem

\[
\max_{\phi_{11}, \ldots, \phi_{p1}} \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1.
\]

- This problem can be solved via a singular-value decomposition of the matrix $X$, a standard technique in linear algebra.

- We refer to $Z_1$ as the first principal component, with realized values $z_{11}, \ldots, z_{n1}$
Further principal components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are *uncorrelated* with $Z_1$.
- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \ldots + \phi_{p2}x_{ip},$$

where $\phi_2$ is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$. 
Test Data

Mean

\(\lambda_1 = 3.4 \cdot 10^5\)

\(\lambda_2 = 2.8 \cdot 10^5\)

\(\lambda_3 = 2.4 \cdot 10^5\)

\(\lambda_4 = 1.6 \cdot 10^5\)

(a)

(b)
• PCA VS Regression

• PCA vs Fisher

• PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
• Clustering looks for homogeneous subgroups among the observations.
Proportion Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.

- The total variance present in a data set (assuming that the variables have been centered to have mean zero) is defined as
  \[ \sum_{j=1}^{p} \text{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2, \]
  and the variance explained by the \( m \)th principal component is
  \[ \text{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2. \]

- It can be shown that \( \sum_{j=1}^{p} \text{Var}(X_j) = \sum_{m=1}^{M} \text{Var}(Z_m), \) with \( M = \min(n - 1, p). \)
Proportion Variance Explained: continued

- Therefore, the PVE of the $m$th principal component is given by the positive quantity between 0 and 1

\[
\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}.
\]

- The PVEs sum to one. We sometimes display the cumulative PVEs.
How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
  - \textit{Why not?}
  - When could we use cross-validation to select the number of components?
- the “scree plot” on the previous slide can be used as a guide: we look for an “elbow”.
Clustering

- *Clustering* refers to a very broad set of techniques for finding *subgroups*, or *clusters*, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other,
- It make this concrete, we must define what it means for two or more observations to be *similar* or *different*.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.
Two clustering methods

- In *K-means clustering*, we seek to partition the observations into a pre-specified number of clusters.
- In *hierarchical clustering*, we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a *dendrogram*, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to $n$. 
K-means

\[ J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2 \]  
\hspace{1cm} (9.1)

Solving for \( r_{nk} \)

\[ r_{nk} = \begin{cases} 
1 & \text{if } k = \arg \min_j \| x_n - \mu_j \|^2 \\
0 & \text{otherwise.} 
\end{cases} \]  
\hspace{1cm} (9.2)

Differentiating for \( \mu_k \)

\[ 2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0 \]  
\hspace{1cm} (9.3)

which we can easily solve for \( \mu_k \) to give

\[ \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}. \]  
\hspace{1cm} (9.4)
A simulated data set with 150 observations in 2-dimensional space. Panels show the results of applying $K$-means clustering with different values of $K$, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the $K$-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.
1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:
   
   2.1 For each of the $K$ clusters, compute the cluster centroid. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.
   
   2.2 Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).
Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective (4) at each step. \textit{Why?} Note that

\[
\frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2,
\]

where \( \bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij} \) is the mean for feature \( j \) in cluster \( C_k \).

- however it is not guaranteed to give the global minimum. \textit{Why not?}
The progress of the K-means algorithm with $K=3$.

- **Top left:** The observations are shown.
- **Top center:** In Step 1 of the algorithm, each observation is randomly assigned to a cluster.
- **Top right:** In Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.
- **Bottom left:** In Step 2(b), each observation is assigned to the nearest centroid.
- **Bottom center:** Step 2(a) is once again performed, leading to new cluster centroids.
- **Bottom right:** The results obtained after 10 iterations.
Different starting values

*K*-means clustering performed six times on the data from previous figure with $K = 3$, each time with a different random assignment of the observations in Step 1 of the *K*-means algorithm.

Above each plot is the value of the objective (4). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters.

Those labeled in red all achieved the same best solution, with an objective value of 235.8.
Hierarchical Clustering

- \( K \)-means clustering requires us to pre-specify the number of clusters \( K \). This can be a disadvantage (later we discuss strategies for choosing \( K \)).

- **Hierarchical clustering** is an alternative approach which does not require that we commit to a particular choice of \( K \).

- In this section, we describe *bottom-up* or *agglomerative* clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.
Hierarchical Clustering Algorithm

The approach in words:

- Start with each point in its own cluster.
- Identify the closest two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.
45 observations generated in 2-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.
45 observations generated in 2-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

- **Left:** Dendrogram obtained from hierarchically clustering the data from previous slide, with complete linkage and Euclidean distance.

- **Center:** The dendrogram from the left-hand panel, cut at a height of 9 (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.

- **Right:** The dendrogram from the left-hand panel, now cut at a height of 5. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.
• NOT USED