MACHINE LEARNING AND DATA MINING WORKSHOP

Liz Lorenzi and Isaac Lavine

UNSUPERVISED LEARNING

Clustering
Dimension Reduction

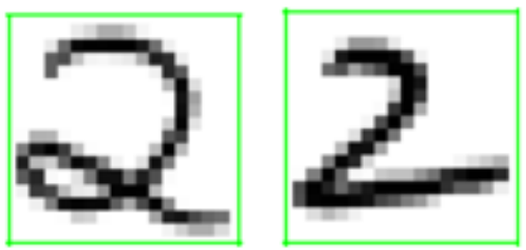
CLUSTERING: WHAT AND WHY?

- Clustering: task of dividing data into groups (clusters) based on similarity.
  - Similarity: points in any one group are more "similar" to each other than points outside the group

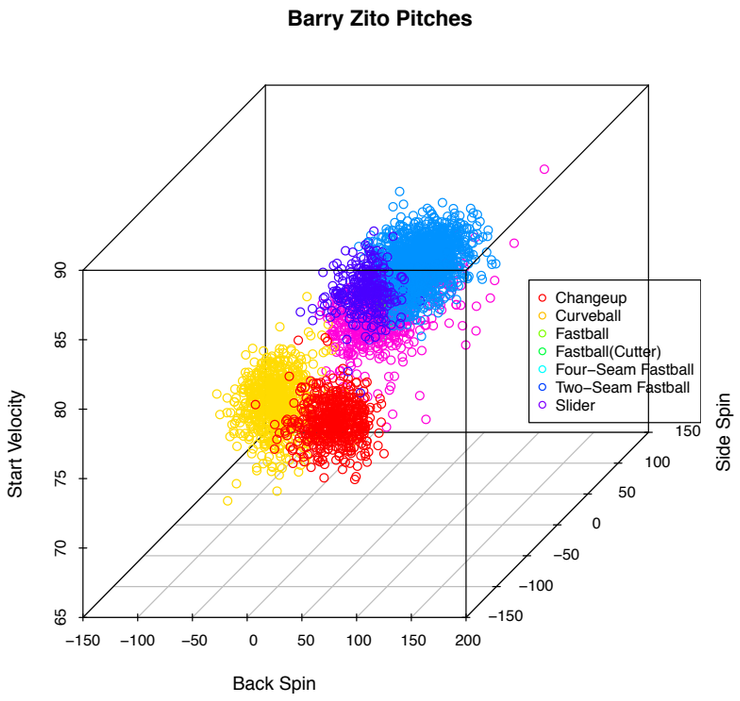
Why Cluster?

- Summary: Learn a reduced representation of the full data
- Discovery: Investigating further into the structure of the groups
  - e.g. finding students that make similar mistakes, finding songs that sound most alike
- Improves prediction

EXAMPLES OF CLUSTERING:



Example 1: Finding handwritten digits with similar structure, NOT identifying which number the handwritten digits are (that's classification)



Example 2: Learning structure of a pitcher's pitches, NOT classifying a pitch by type.

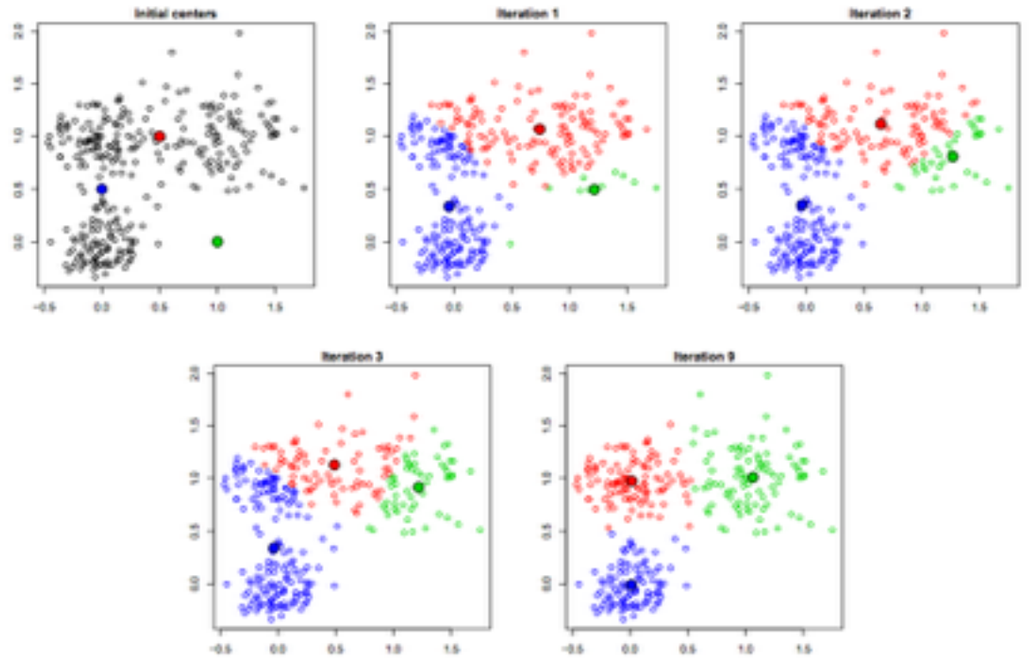
K-MEANS

- K-means: Find clusters by minimizing distance of points from K means

Properties:

1. *Must choose the number of clusters, K*
2. *Clusters depend on the initial starting position*

Here $X_i \in \mathbb{R}^2$, $n = 300$, and $K = 3$




```
km = kmeans(x, centers=k, nstart=10, algorithm="Lloyd")
```

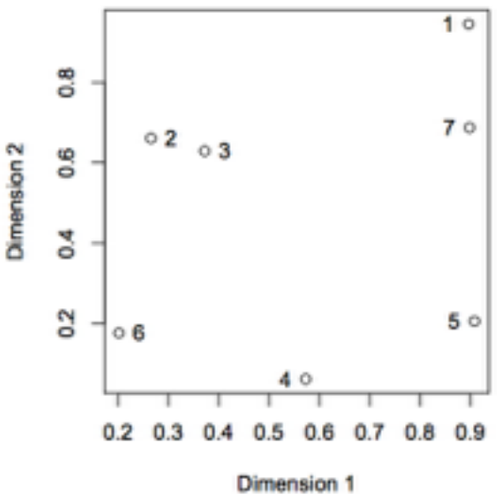
centers: number of means to learn (number of clusters)

nstart: number of random starts (often results of clusters are sensitive to starting position)

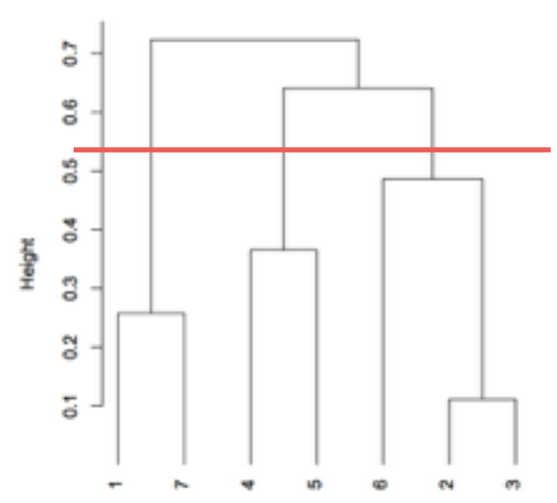
algorithm: use the "Lloyd" option

HIERARCHICAL CLUSTERING

- Agglomerative hierarchical clustering:
  - Start with each data point in their own cluster
  - Merge two groups with smallest dissimilarity until only one cluster remains



Step 1: $\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\}$;
Step 2: $\{1\},\{2,3\},\{4\},\{5\},\{6\},\{7\}$;
Step 3: $\{1,7\},\{2,3\},\{4\},\{5\},\{6\}$;
Step 4: $\{1,7\},\{2,3\},\{4,5\},\{6\}$;
Step 5: $\{1,7\},\{2,3,6\},\{4,5\}$;
Step 6: $\{1,7\},\{2,3,4,5,6\}$;
Step 7: $\{1,2,3,4,5,6,7\}$.




```
d = dist(x)
tree.avg = hclust(d, method="average")
plot(tree.avg)
```

DIMENSION REDUCTION

Dimension reduction: finding a lower-dimensional representation of your data.

Principal Component Analysis:

1. Center your data, $X \in \mathbb{R}^{n \times p}$
2. Let $S = X^T X$
3. Perform eigendecomposition of S, so that:

$$S = V \Lambda V^T$$

$$V^T V = I$$

$$\Lambda = \text{diag}(\lambda_1, ..., \lambda_p)$$

$$S v_j = \lambda_j v_j$$

PCA CONTINUED

The first principal component direction of X is the unit vector $v_1 \in \mathbb{R}^p$ *that maximizes the sample variance of* $Xv_1 \in \mathbb{R}^n$ *when compared to all other unit vectors*

$$v_1 = \underset{\|v\|_2=1}{\operatorname{argmax}} (Xv)^T(Xv)$$

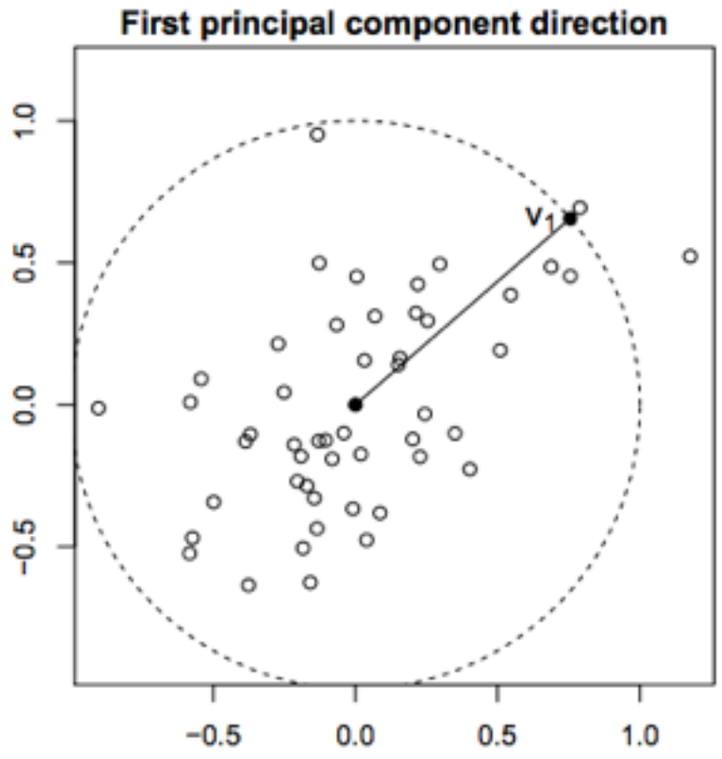
The first principal component is therefore:

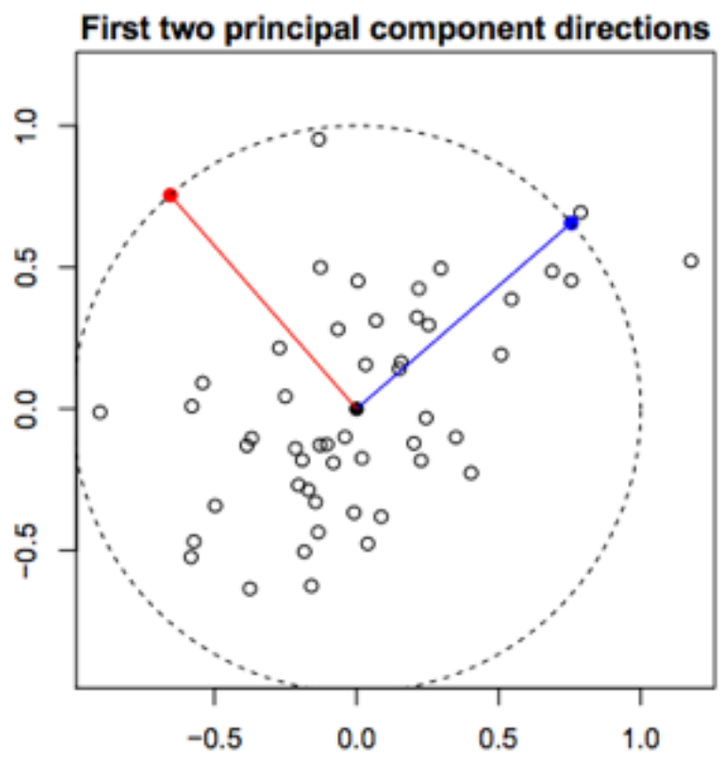
$$Xv_1 \in \mathbb{R}^n$$

The amount of variance explained by v_1 *is* d_1^2/n

$$d_1 = \sqrt{(Xv_1)^T(Xv_1)}$$

EX. PCA






```
pc = prcomp(x)
dirs = pc$rotations # loadings/directions
scrs = pc$scores # scores
```

R CODE: PCA

- Examine the rows and columns of the USArrests data
  - `states= row.names(USArrests); summary(USArrests)`
- Perform PCA using prcomp (prcomp centers and scales the data for you when scale=TRUE).
  - `pr.out = prcomp(USArrests, scale=TRUE)`
- Examine the loadings matrix:
  - `pr.out$rotations`
- Plot the principal components:
  - `biplot(pr.out, scale=0)`

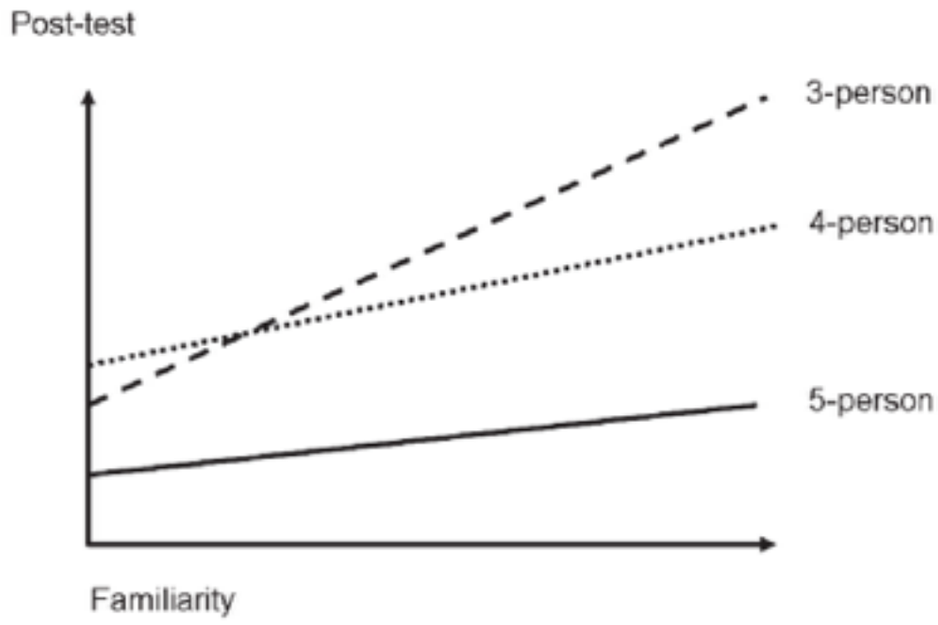
SUPERVISED LEARNING IN CLUSTER SETTINGS

MULTI-LEVEL/HIERARCHICAL MODELS

- Often data contains nested structure among the data points, such that we know there exists inherent groups among the rows
  - e.g. modeling student test scores from different schools, measuring effect of drug on patients from different hospitals
- 2 ways in R to implement hierarchical models: lme4 - frequentist way to model hierarchies, rstan/r2jags - Bayesian models that allow flexible modeling of the subgroups

HIERARCHICAL MODELING – TERMINOLOGY

- A typical linear model is composed of **fixed effects**, or effects of variables that are fixed for all data points
- When we know we have groups in our data, we may want to know the effects within each group - **random effects**
  - **Random Effects** come in 2 different forms:
    - **Random Intercepts:** allowing group-specific intercepts
      - Adjustment for when the means of the groups are different
    - **Random Slopes:** allows coefficients/slopes to be different per group
      - Adjustment for when groups have different relationships with the covariates



HIERARCHICAL MODELING IN R

- Load in lme4:

```
library(lme4)
```

- Load in some data:

```
install.packages("mlmRev")
library(mlmRev)
data(Exam)
```

- Fit linear model with no group effects to data:

```
lmer(normexam ~ standLRT, data=Exam)
```

- Fit random intercepts only model:

```
lmer(normexam ~ (1 | school), data=Exam)
```

- Fit random intercept plus fixed effect:

```
lmer(normexam ~ standLRT + (1 | school), data=Exam)
```

- Fit random intercept plus random slope:

```
lmer(normexam ~ standLRT + (1+standLRT | school), data=Exam)
```

- Fit full model:

```
lmer(normexam ~ standLRT * schavg + (1 + standLRT | school), data=Exam)
```

WORKSHOP: PCA/CLUSTERING