Clustering

CSCI 5622

David Quigley

Spring 2020

The Machine Learning Space (Week 1)

	Supervised	**Unsupervised**
Discrete	*Classification*	*Clustering*
Continuous	*Regression*	*Dimensionality Reduction*

Supervised vs. Unsupervised Learning

Supervised – ground truth X (features), Y (answer)

- Trying to get a machine to predict Y based on X
- Building / implementing a function to map X to Y
  - Trying to capture the hidden structure

Supervised vs. Unsupervised Learning

Supervised – ground truth X (features), Y (answer)

- Trying to get a machine to predict Y based on X
- Building / implementing a function to map X to Y
  - Trying to capture the hidden structure

Discrete

Classification

- divide my samples into classes

Continuous

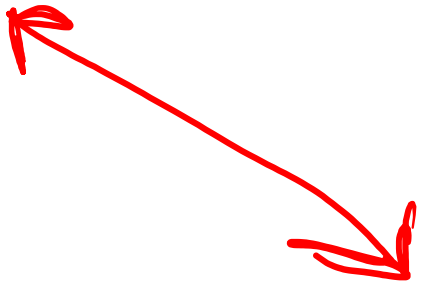
Regression

- generate a continuous value

Supervised vs. Unsupervised Learning

Supervised – ground truth X, Y

- Trying to get a machine to predict Y based on X
- Building / implementing a function to map X to Y
  - Trying to capture the hidden structure



Unsupervised – ground truth X

- No ground truth Y*
- Building / implementing a function to model hidden structure

*Ground truth may actually exist, we just a) can't capture it, or b) are choosing to not use it for our task

The Machine Learning Space (Week 1)

	Supervised	**Unsupervised**
Discrete	*Classification*	*Clustering*
Continuous	*Regression*	*Dimensionality Reduction*

Clustering

Clustering – ground truth X

- No ground truth Y*
- Trying to develop classes from the bottom up
  - Trying to build based on hidden structure

*Ground truth may actually exist, we just a) can't capture it, or b) are choosing to not use it for our task

Clustering

Clustering – ground truth X

- No ground truth Y*
- Trying to develop classes from the bottom up
  - Trying to build based on hidden structure

Finding clusters such that

- Examples within a cluster are similar
- Examples between clusters are dissimilar

minimize within cluster variance

maximize between cluster variance

*Ground truth may actually exist, we just a) can't capture it, or b) are choosing to not use it for our task

Clustering Examples

Clustering Examples

Clustering

Clustering – ground truth X

- No ground truth Y*
- Trying to develop classes from the bottom up
  - Trying to build based on hidden structure

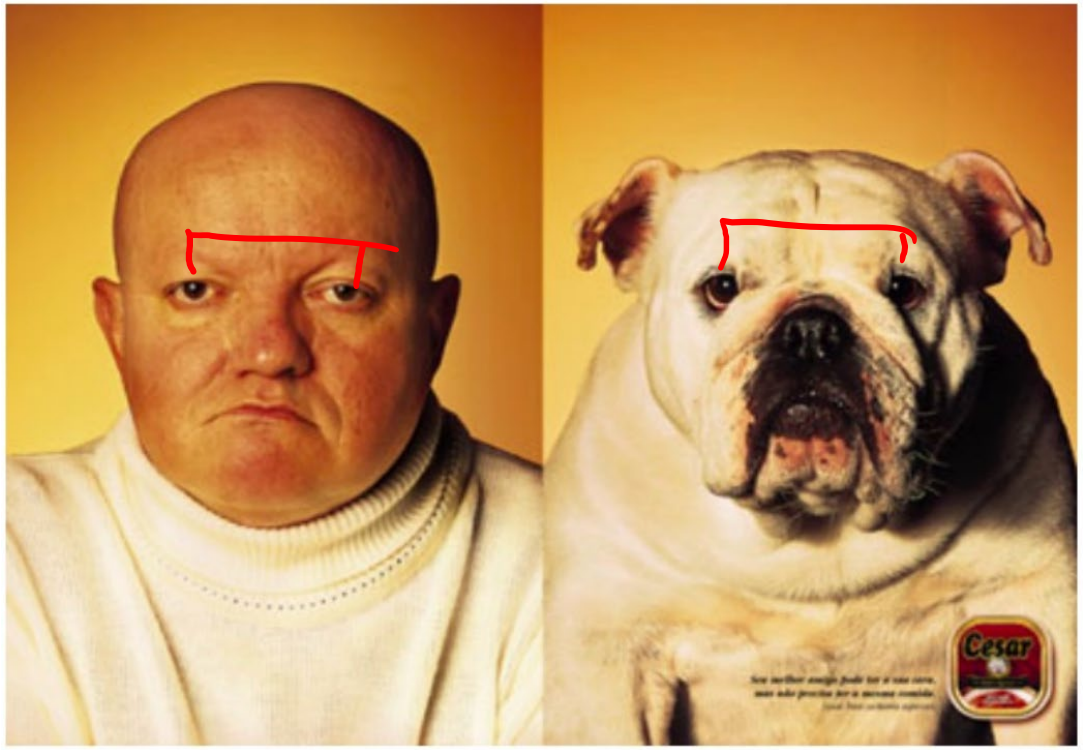
Finding clusters such that

- Examples within a cluster are similar
- Examples between clusters are dissimilar

*Ground truth may actually exist, we just a) can't capture it, or b) are choosing to not use it for our task

Clustering – Examples of Similarity

Eyeball test



Clustering – Examples of Similarity

Eyeball test

Not something we can calculate

Clustering – Examples of Similarity

Euclidian Distance = $||x - y||^2$

Clustering – Examples of Similarity

Euclidian Distance = $||x - y||^2$

Edit Distance: Number of character changes (insert, delete, edit) to convert string x to string y

Clustering – Examples of Similarity

Euclidian Distance = $||x - y||^2$

Edit Distance: Number of character changes (insert, delete, edit) to convert string x to string y

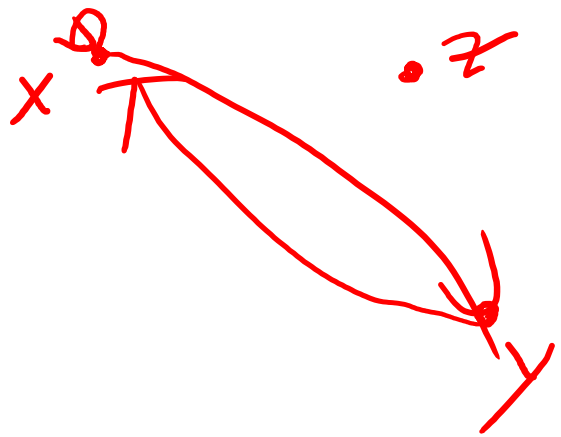
Sit to Cart

Cit

Cat

Cart

Clustering – Similarity Rules



Given a distance measure d(x,y)

Symmetry: d(x,y) = d(y,x)

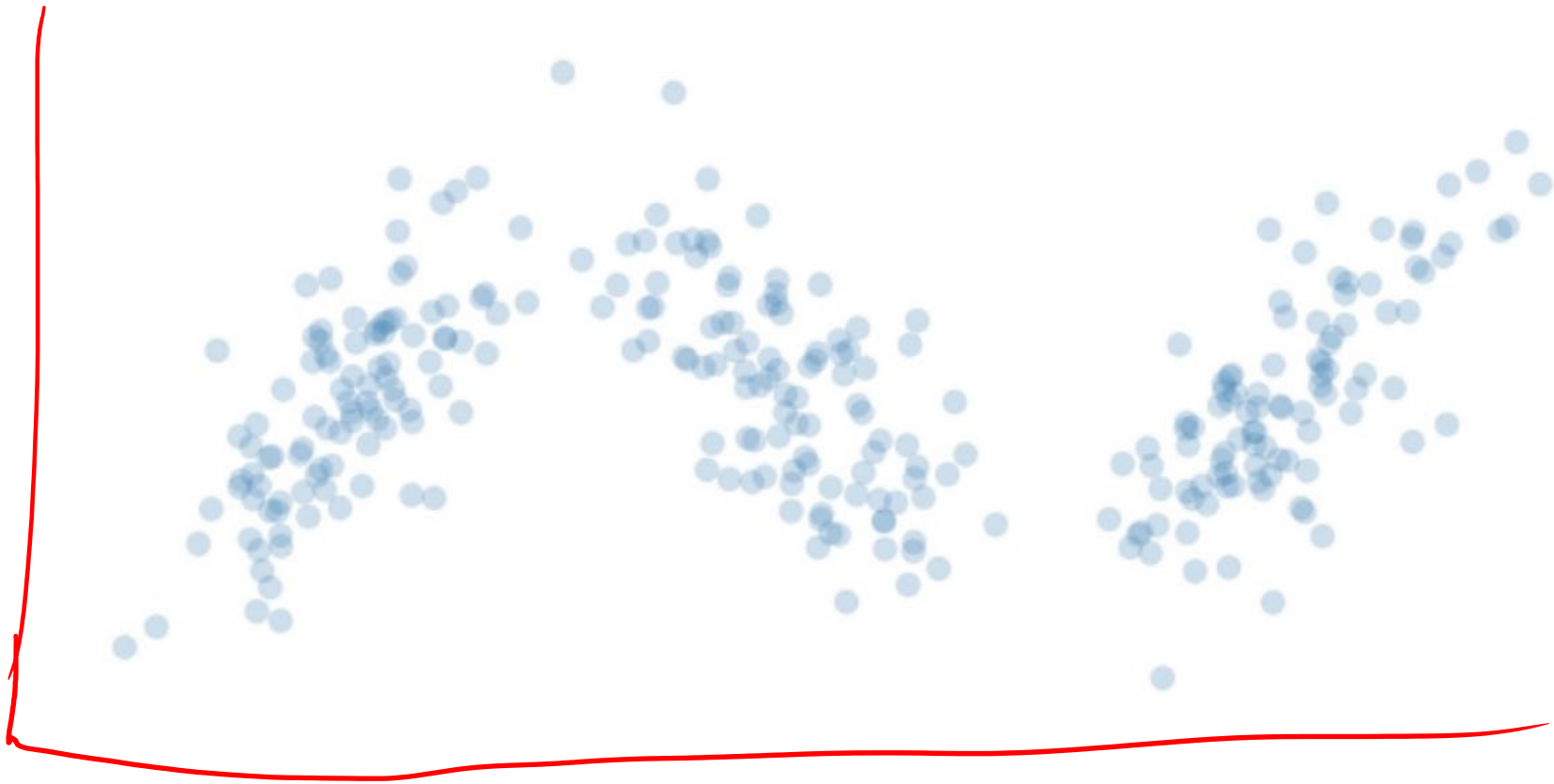
Self Consistency: d(x,x) = 0

Positivity: d(x,y) = 0 if x = y

Triangle Inequality: d(x,y) <= d(x,z) + d(z,y)

Clustering – K Means

Start with your Data



Clustering – K Means

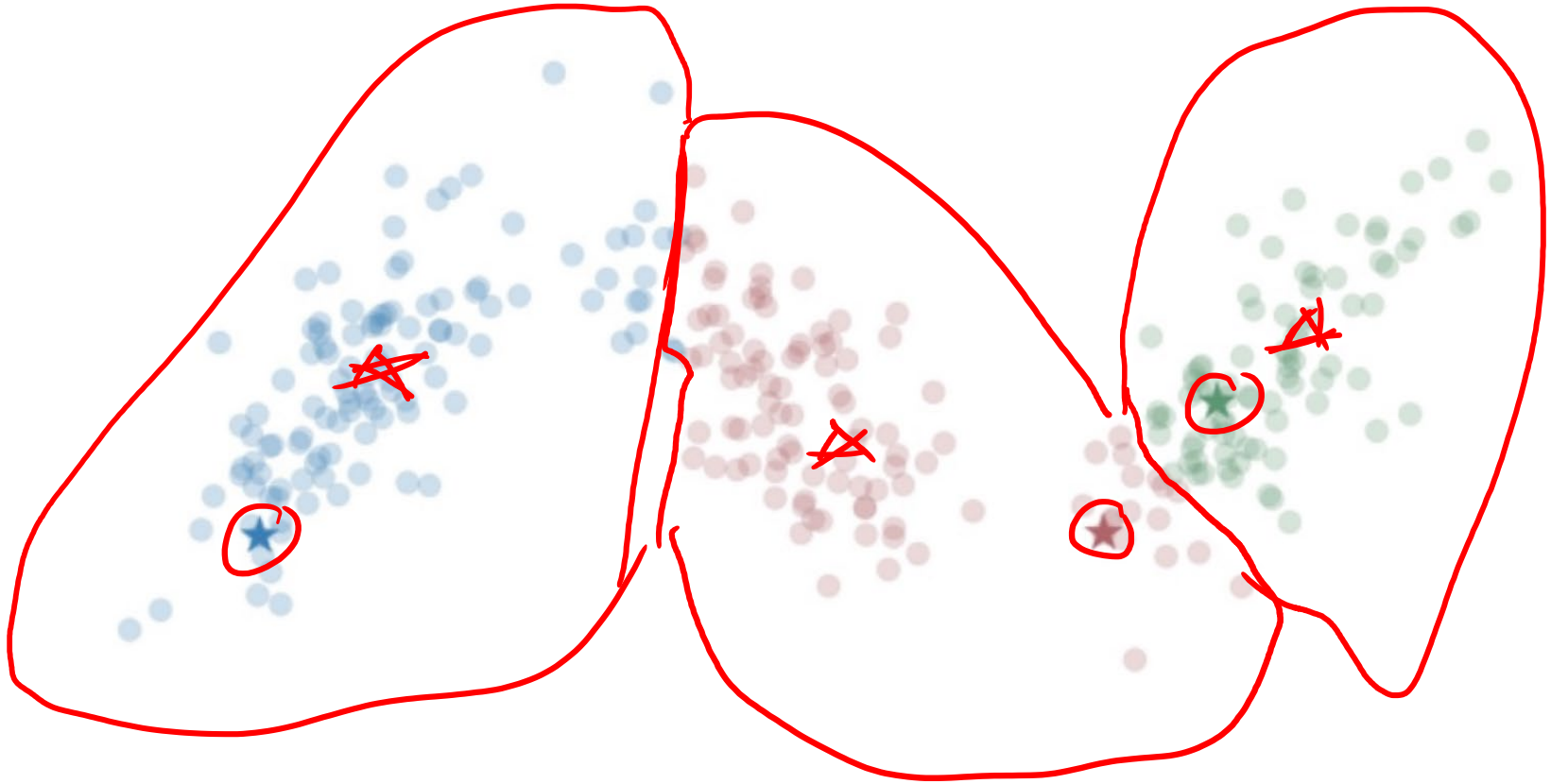
Start with your Data

Choose K centroids

(means)

Assign each point to its

nearest cluster



Clustering – K Means

Start with your Data

Choose K centroids

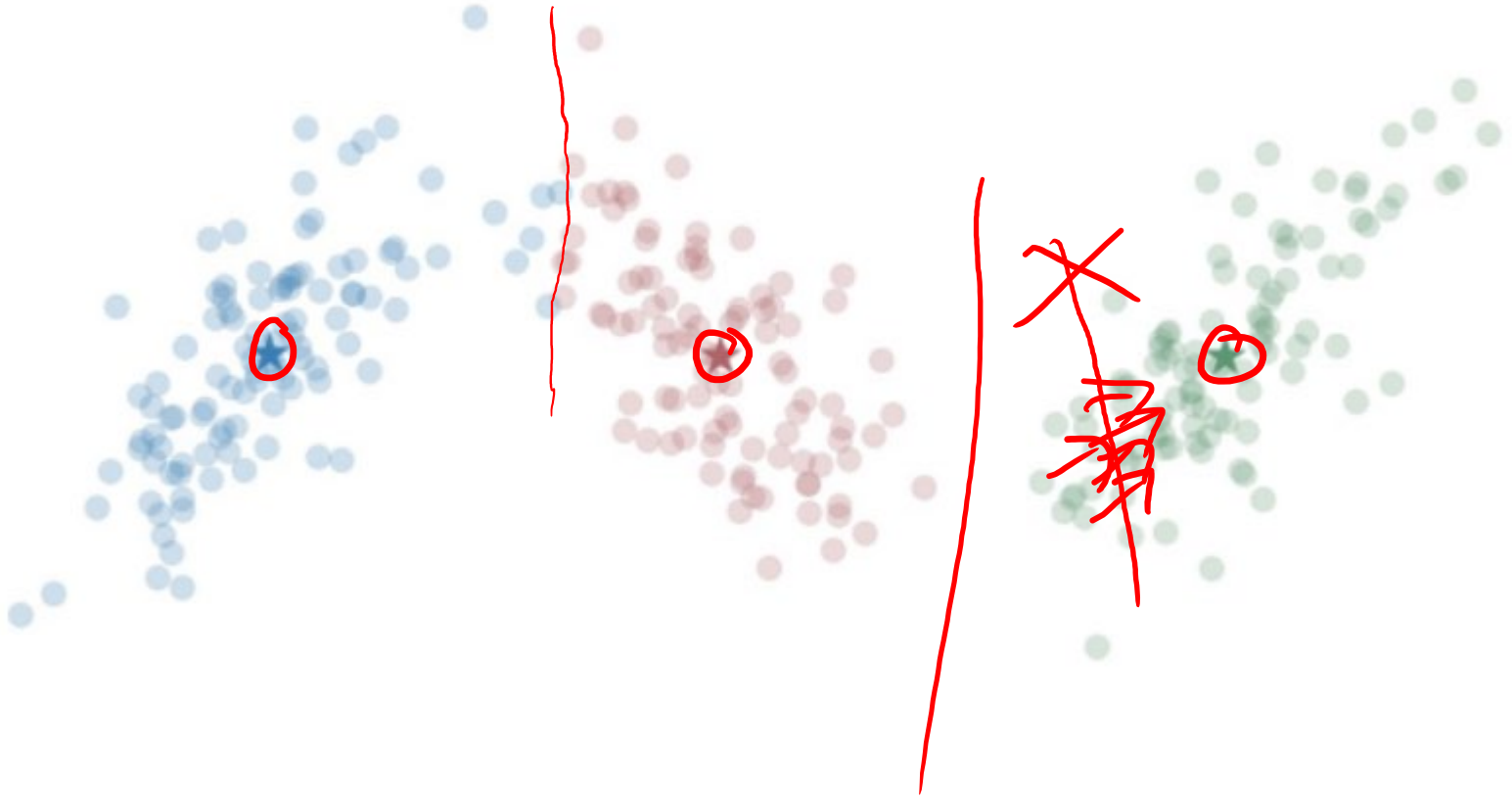
(means)

Assign each point to its

nearest cluster

Re-center your centroids

(means)



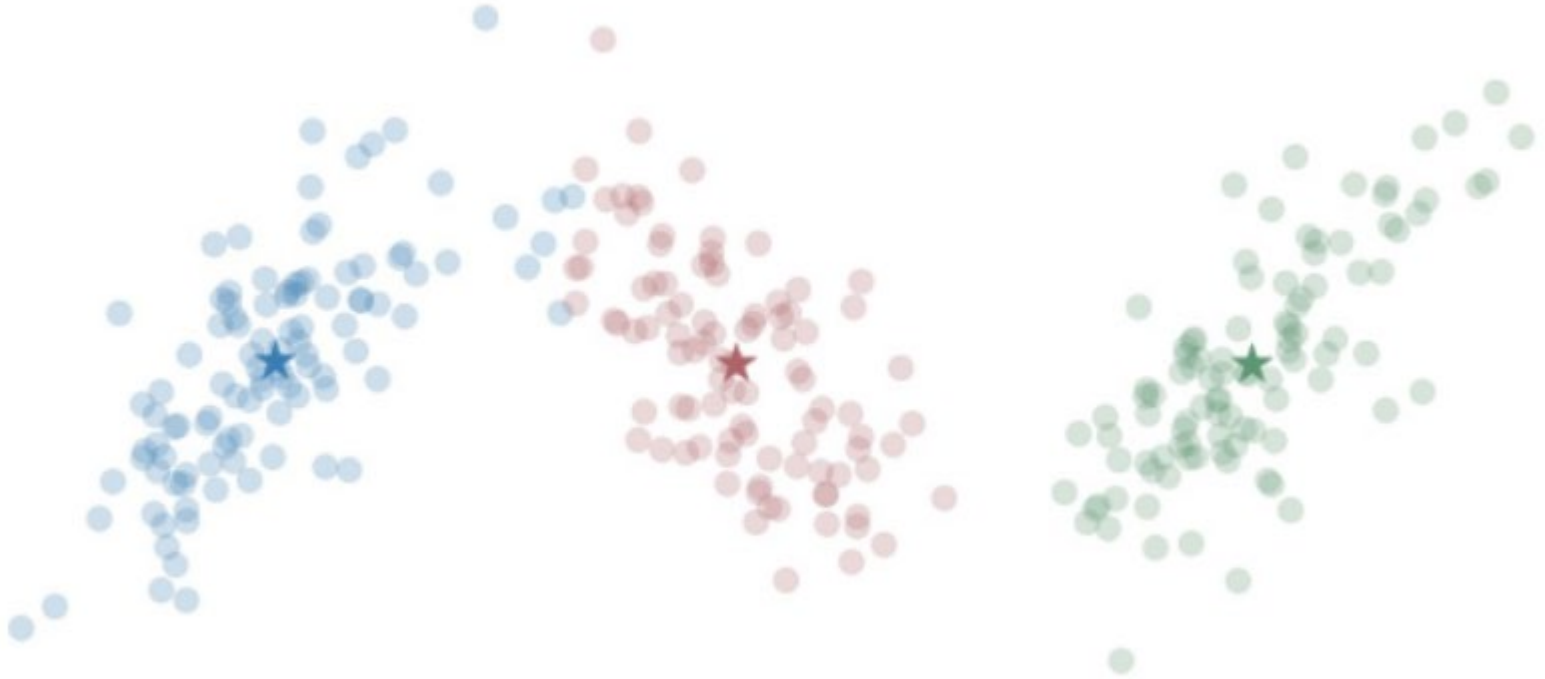
Clustering – K Means

Start with your Data

Choose K centroids

Until we converge:

- Reassign points
- Re-center



Clustering – K Means

Samples = matrix (of features X entries)

Cents = array of K random points (within min, max of each dimension)

While not max_iter, changed_points > 0, Cents – Old_cents > lamda:

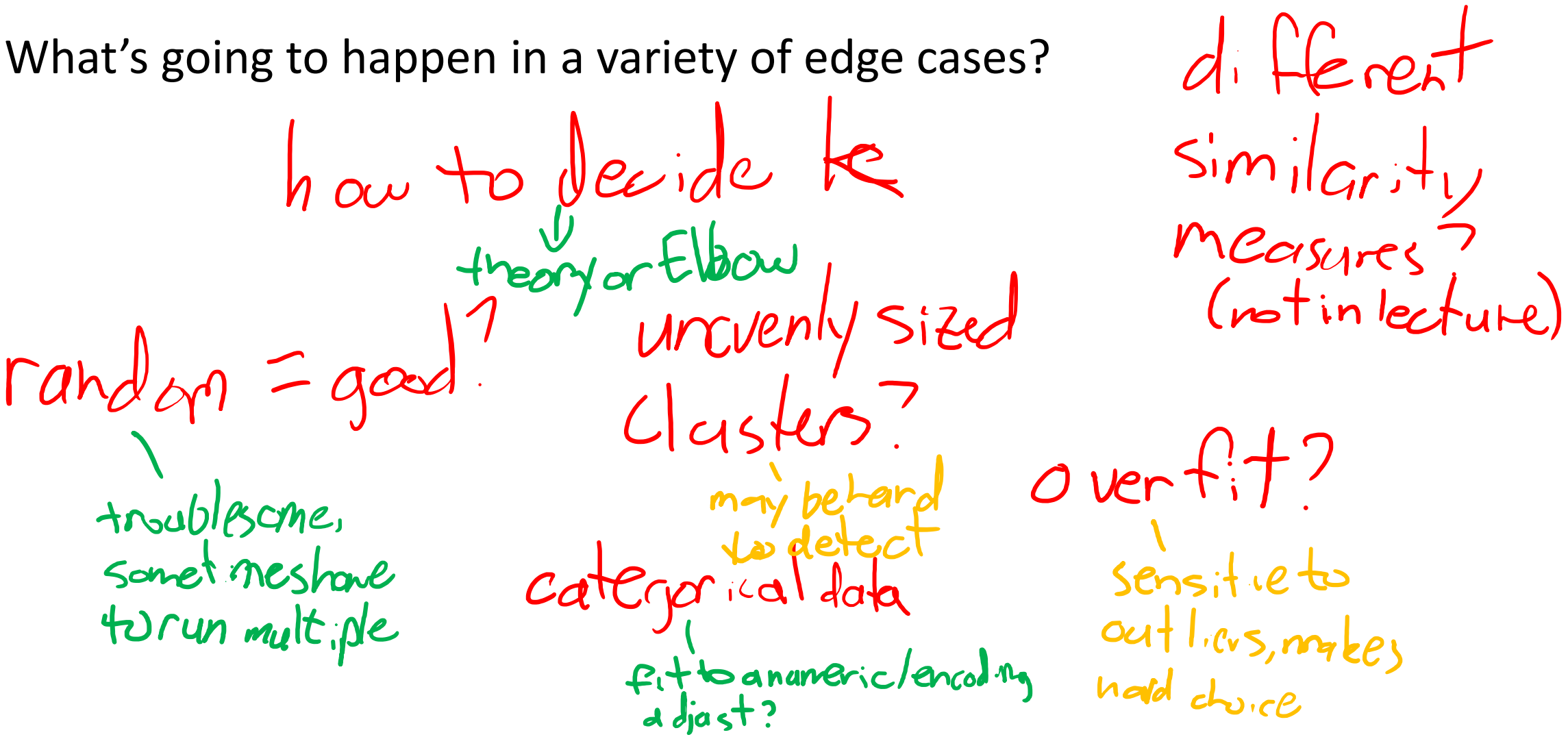
    Sample_cluster = array tracking nearest centroid (aka cluster)

    Cents ← recompute based on Sample_cluster (keep old to loop)

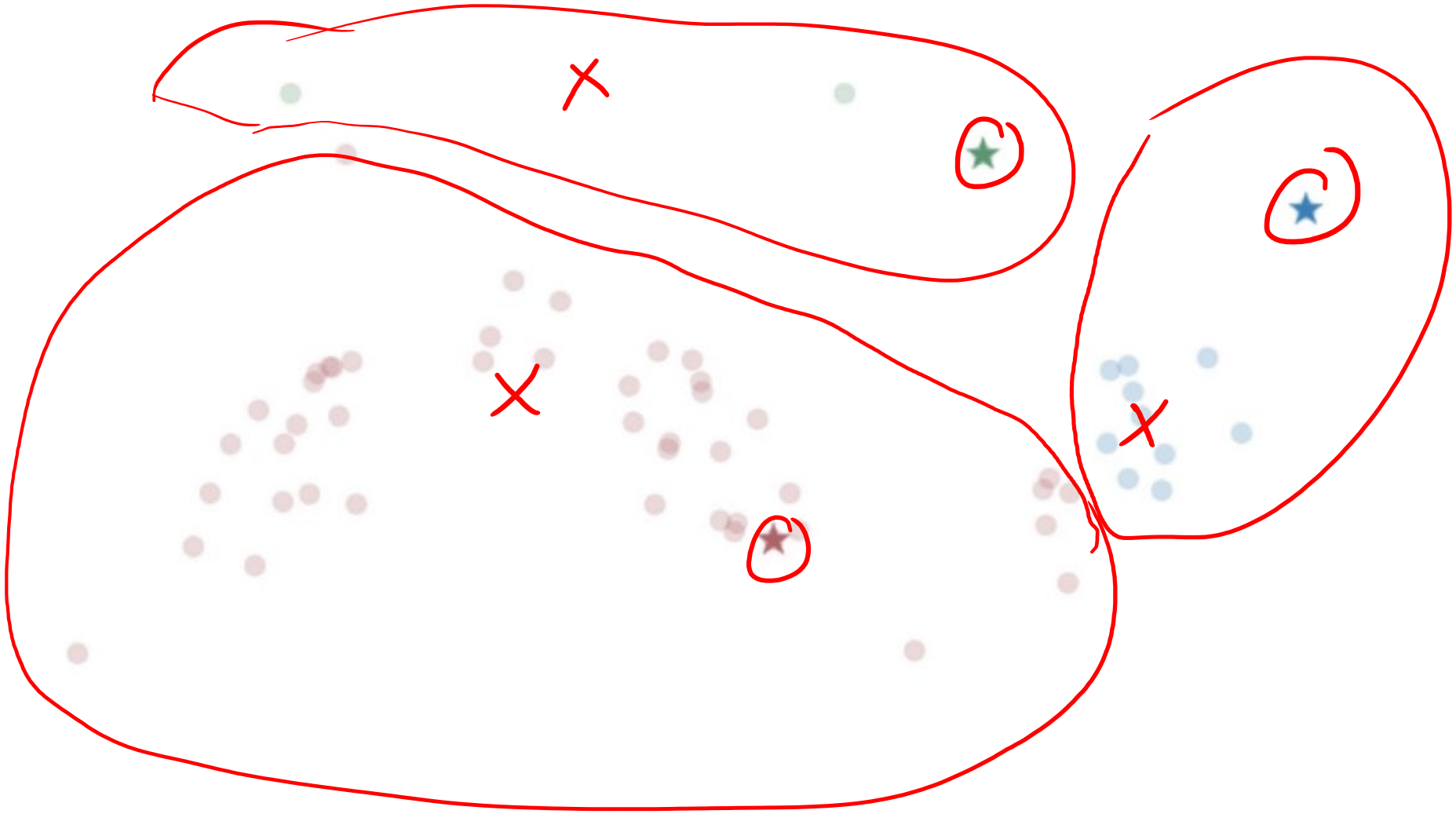
Return sample_cluster, Cents

K Means

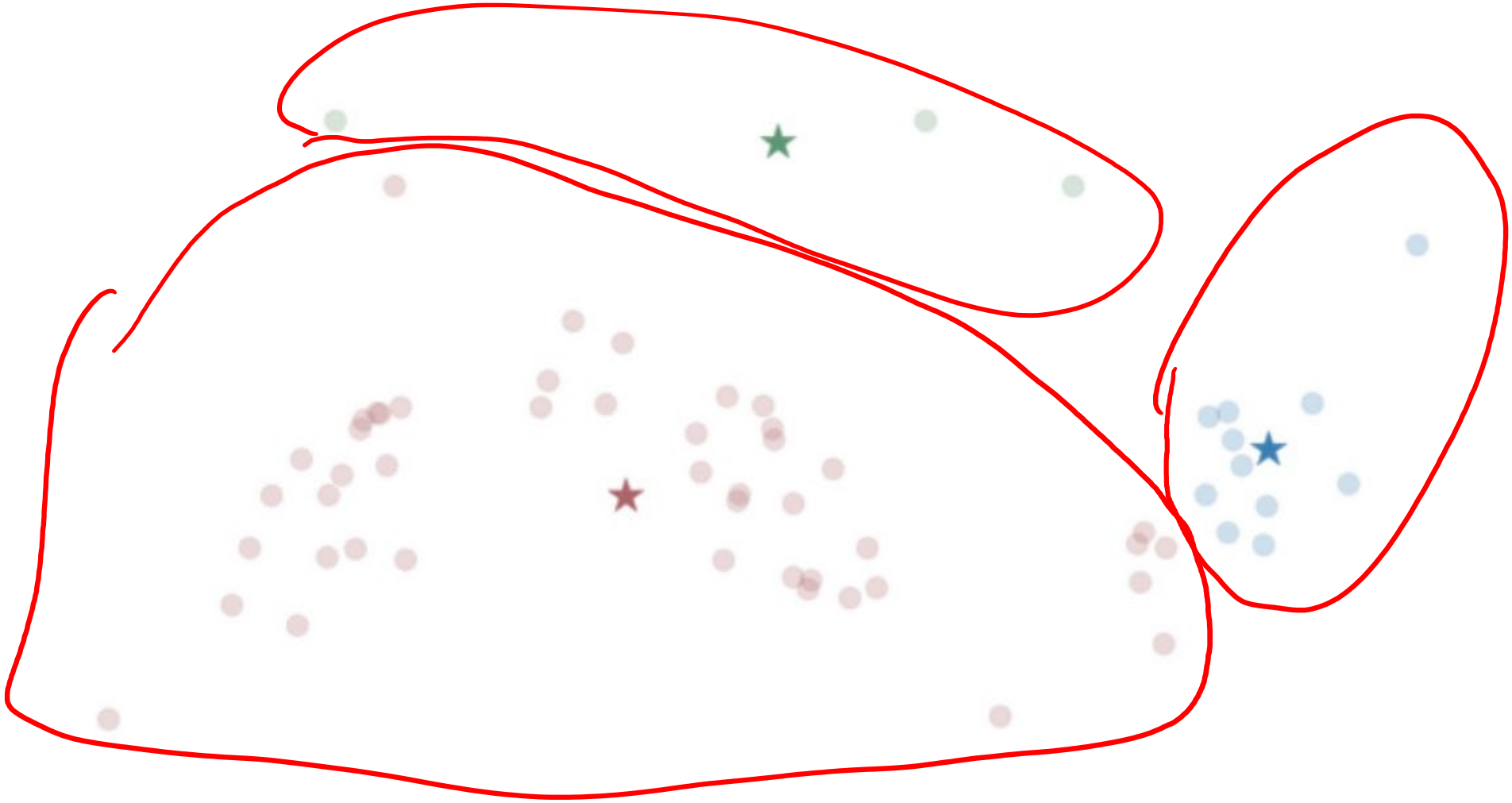
What's going to happen in a variety of edge cases?



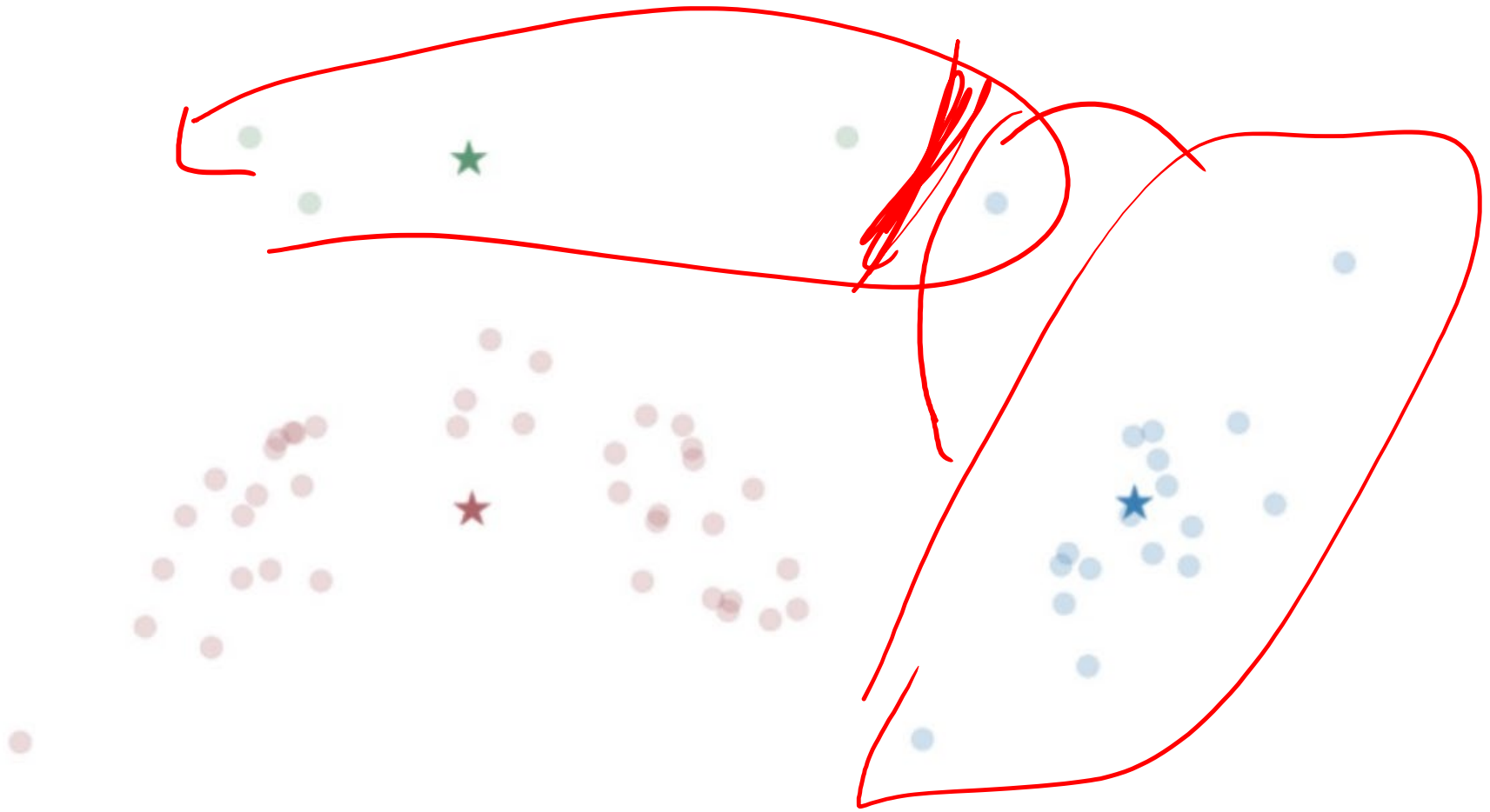
K Means - Outliers



K Means – Outliers



K Means – Outliers



K Means – Sensitive to Initial Centroids

K Means can be easily thrown off by outliers or nonlinearities

Can converge to a variety of points

- Converging to *Local Minimum*

Fix?

K Means – Sensitive to Initial Centroids

K Means can be easily thrown off by outliers or nonlinearities

Can converge to a variety of points

- Converging to *Local Minimum*

Fix?

Run several times (random initialization) and choose best

K Means - # Clusters

K Means - # Clusters

Choose a theoretically grounded value

- based on the “secret” ground-truth classification

- based on a hypothesis from other research

- based on prior work

K Means - # Clusters

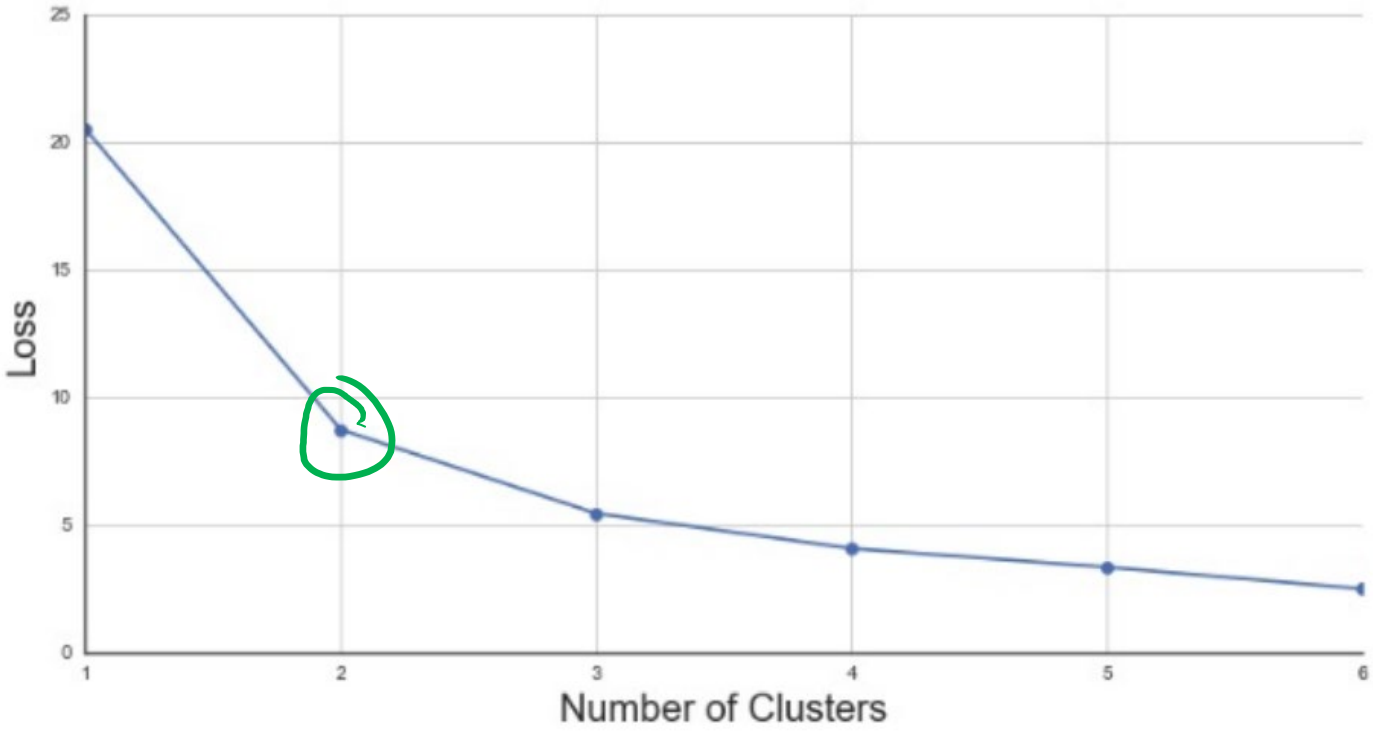
min within var
max between var

Elbow Method

- Choose the number of clusters with the most change in the derivative of the loss (i.e. choose the most bent elbow)

L2 Cost Function

$$\sum_{x \in S_j} (x - \mu_j)^2$$



K Means – Categorical Data

Will this approach work?

K Means – Categorical Data

Solution: Encode same or different / create one-hot vectors

K Means – Categorical Data

Solution: Encode same or different / create one-hot vectors

Improvement: adjust distance based on how common an option is

- If 90% of your sample is non-smoking, being a smoking sample is more of an oddity, so perhaps it should differentiate a cluster

K Means – Categorical Data

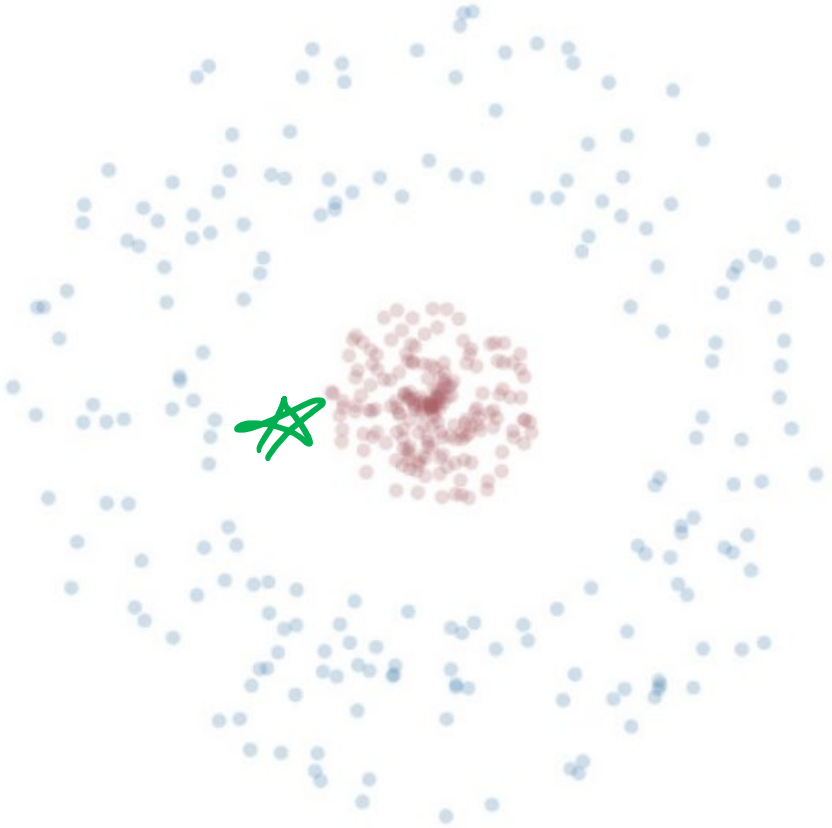
Solution: Encode same or different / create one-hot vectors

Improvement: adjust distance based on how common an option is

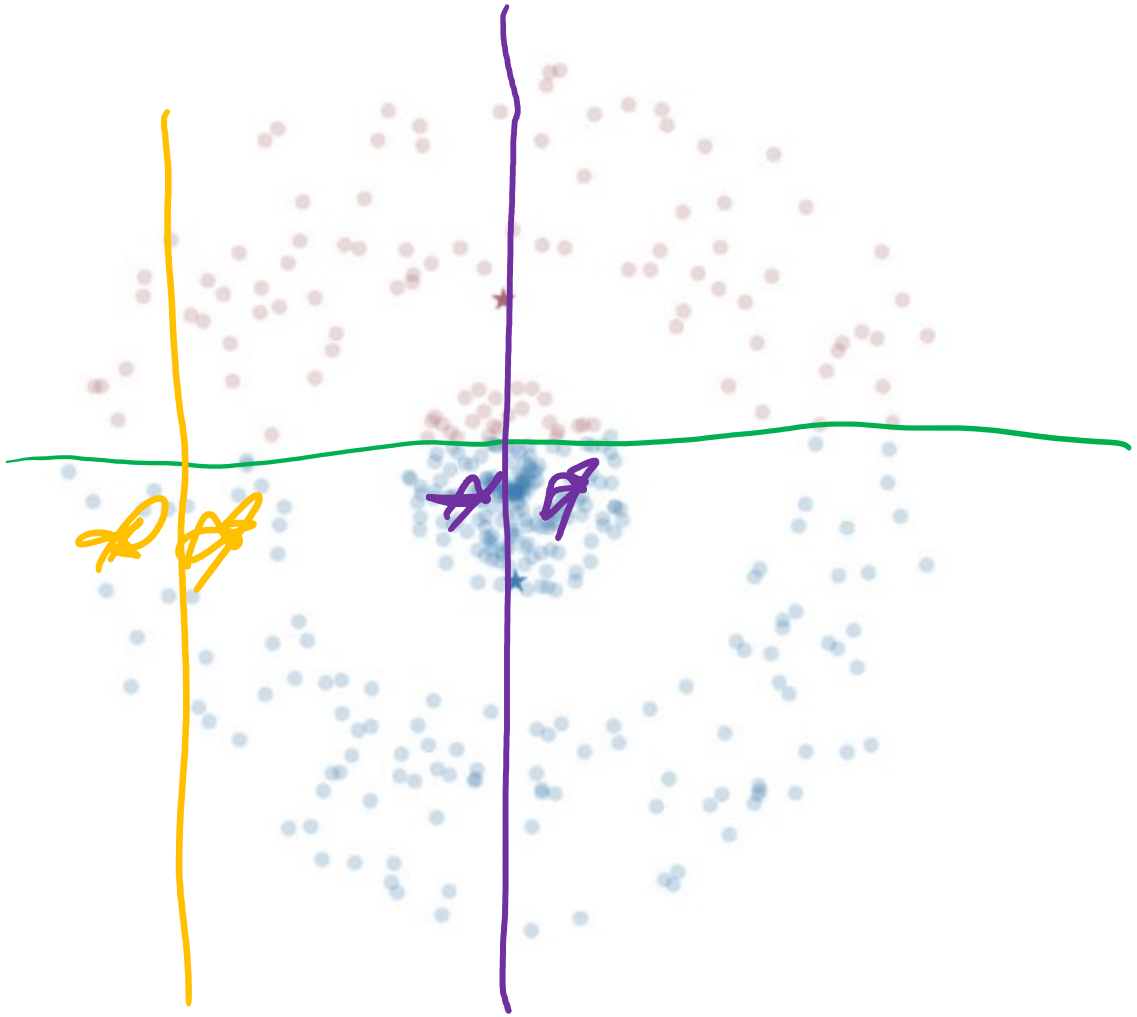
- If 90% of your sample is non-smoking, being a smoking sample is more of an oddity, so perhaps it should differentiate a cluster
- K Means – adjust based on popularity for non-binary
- K Modes – simplify as binary, either most popular choice or not

https://shapeofdata.wordpress.com/2014/03/04/k-modes/

K Means – Non-Convex Clusters



K Means – Non-Convex Clusters



K Means – Convex Clusters Solutions

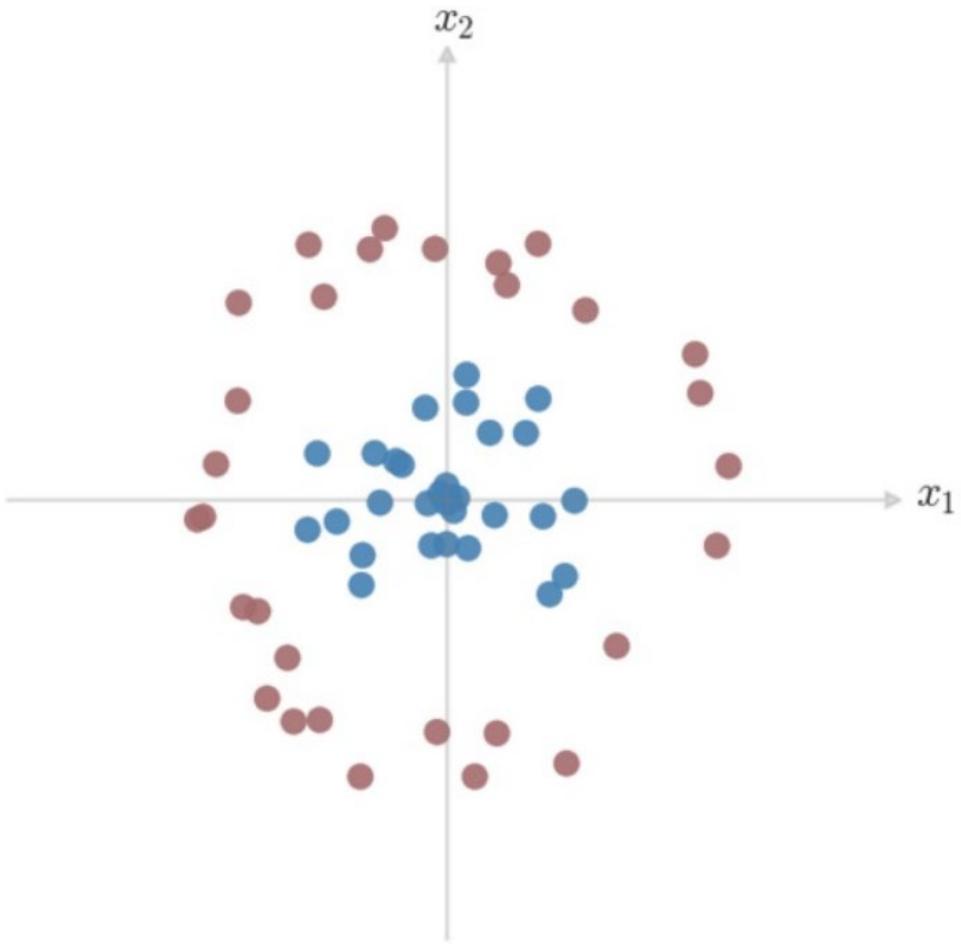
Transform your data?

Dimension Projection

How would we separate these classes?

What kind of boundary can we give this space?

Can we express that boundary in terms of our given features?

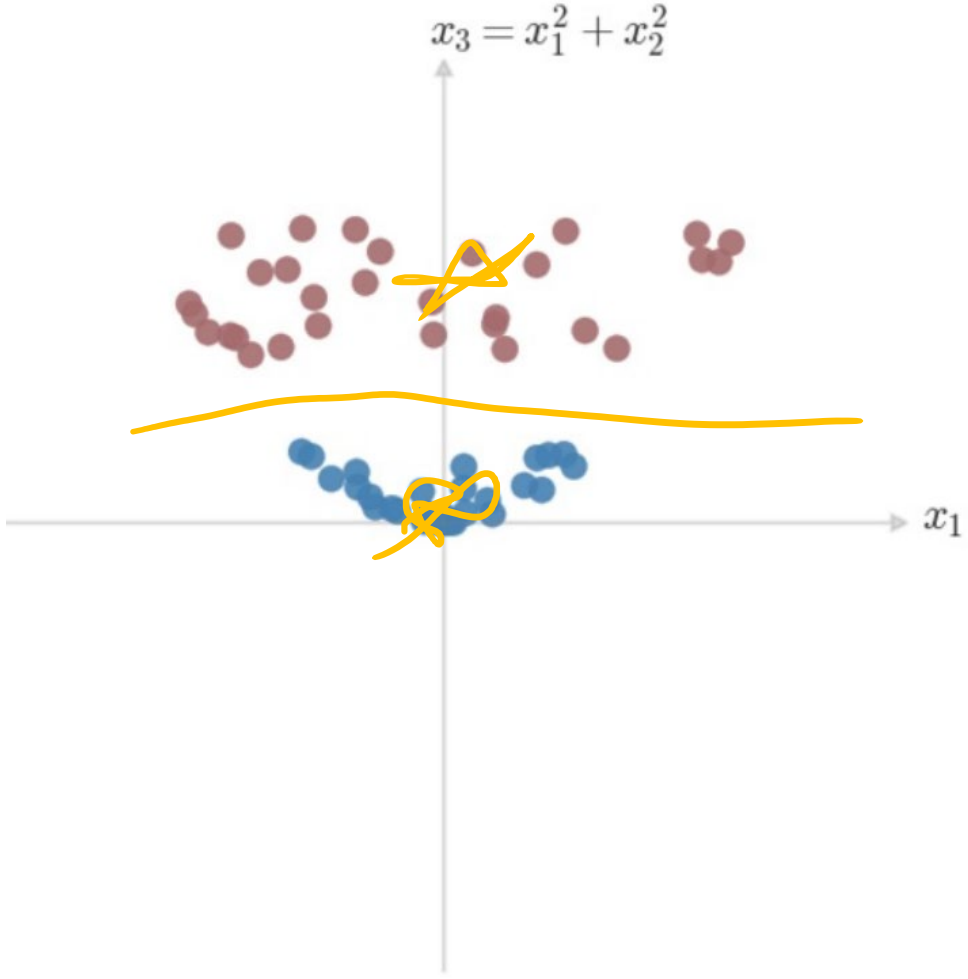


Dimension Projection

How would we separate these classes?

What kind of boundary can we give this space?

Can we express that boundary in terms of our given features?



K Means – Convex Clusters Solutions

Transform your data, add iterative testing?

K Means – Pros & Cons

PROS

Simple to understand & implement

Very fast to converge

Can make theoretical sense when mapping to cluster features

K Means – Pros & Cons

PROS

Simple to understand & implement

Very fast to converge

Can make theoretical sense when mapping to cluster features

CONS

Doesn’t handle non-convex clusters well

Sensitive to oddities from outliers, initialization

Makes a hard choice (assigns to one and only one cluster)

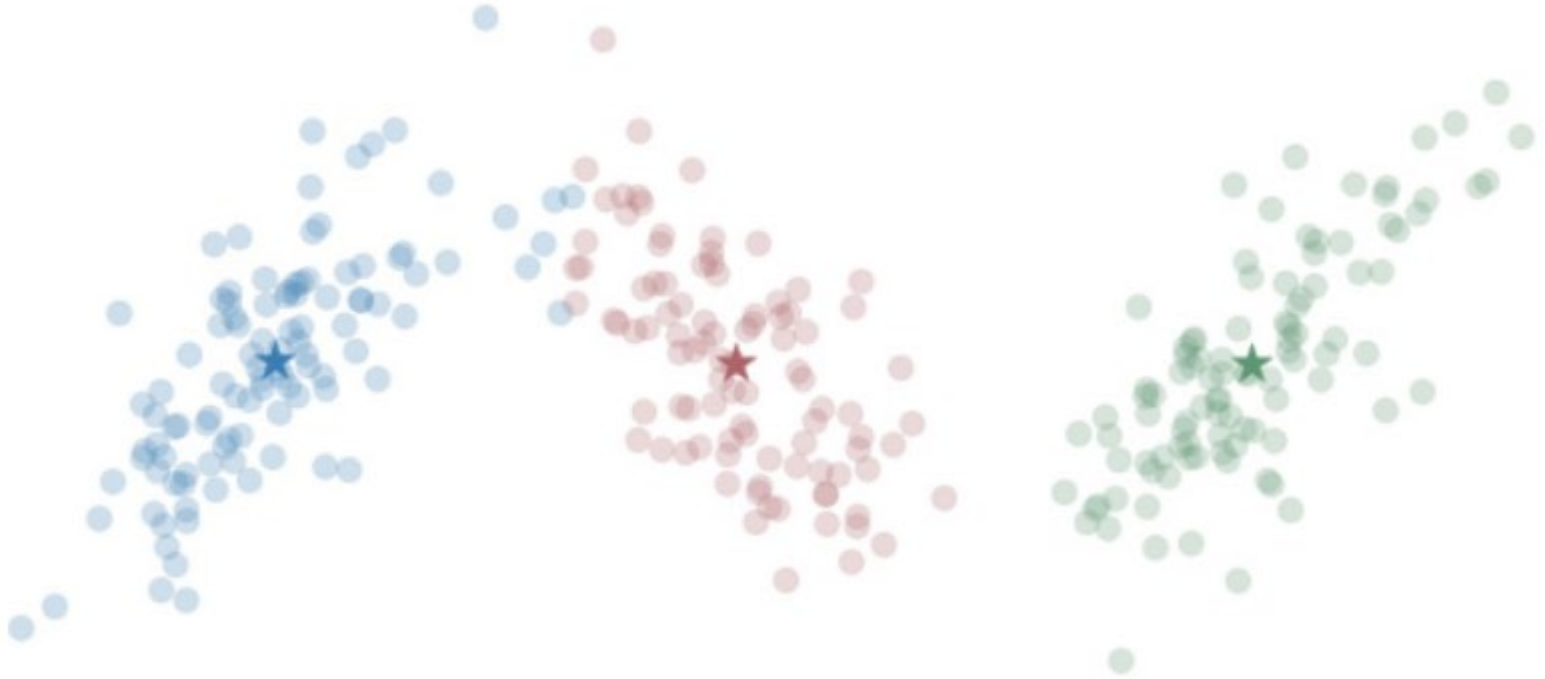
Clustering – K Means

Start with your Data

Choose K centroids

Until we converge:

- Reassign points
- Re-center



Generalizing K Means

Is there a way we can discuss clusters without a hard decision boundary?

Generalizing K Means

Is there a way we can discuss clusters without a hard decision boundary?

Add a probability to the assignment!

Generalizing K Means

Is there a way we can discuss clusters without a hard decision boundary?

Add a probability to the assignment!

$$p(x_i, z_i) = p(x_i|z_i)p(z_i)$$

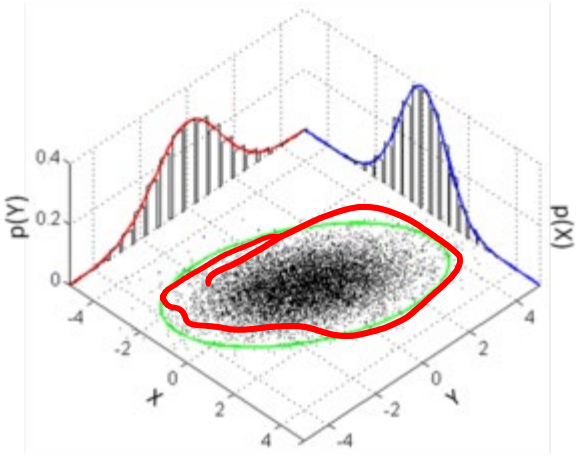
Where x_i is my sample, z_i is the assigned cluster (z = {1…K})

(to make cluster assignments, we just chooze z_i with the highest probability for each x)

Generalizing K Means – Gaussian Distribution

$$p(x_i, z_i) = p(x_i|z_i)p(z_i)$$

Where x_i is my sample, z_i is the assigned cluster (z = {1…K})



Treat clusters as a Multivariate Gaussian Distribution

$$N(\mu_k, \Sigma_k)$$

μ_k= mean vector (a.k.a. centroid)

Σ_k= covariance matrix

π= prior for z

Generalizing K Means – Gaussian Distribution

$$p(x_i, z_i) = p(x_i|z_i)p(z_i)$$

Where x_i is my sample, z_i is the assigned cluster ($z_i \in \{1 \dots K\}$)

Mahalanobis distance – "distance" from x to μ

*technically this is the square of the mahalanobis distance…

Treat clusters as a Multivariate Gaussian

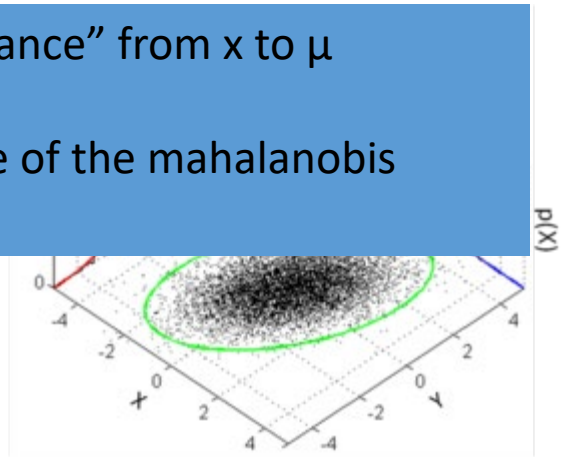
$$N(\mu_k, \Sigma_k)$$

μ_k= mean vector (a.k.a. centroid)

Σ_k= covariance matrix

π= prior for z

$$p(x|z_i = k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$



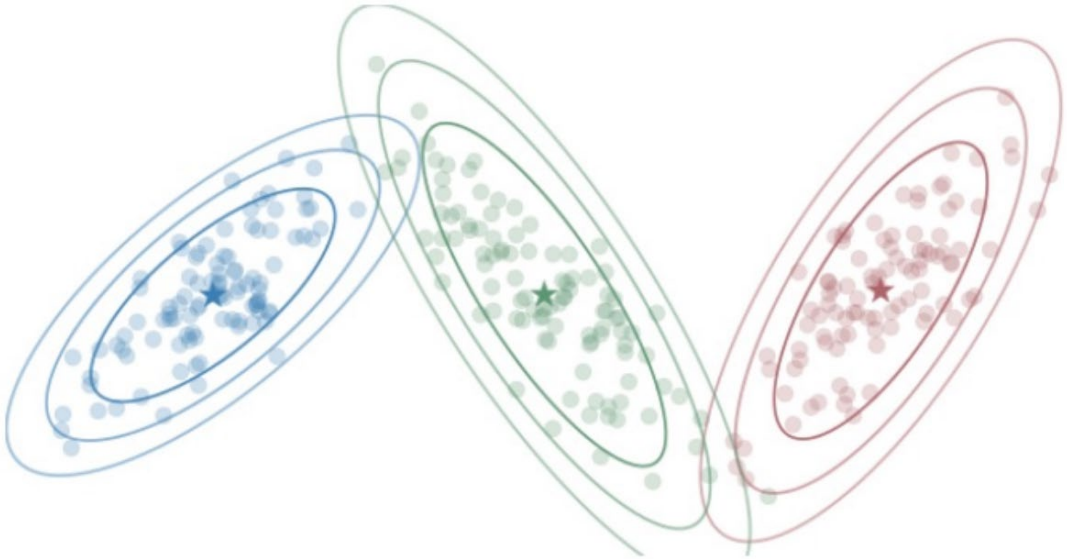
Gaussian Distribution – Gaussian Mixture Model

We can define the probability a set of points *x* is sampled from a multivariate gaussian…

Gaussian Distribution – Gaussian Mixture Model

We can define the probability a set of points *x* is sampled from a multivariate gaussian…

Let’s try to find a set (mixture) of gaussians that best model the data!



Gaussian Mixture Model – Maximum Likelihood

Define the log-likelihood of all our variables, and maximize that!

$$l(\pi,\mu,\Sigma)$$

Gaussian Mixture Model – Maximum Likelihood

Define the log-likelihood of all our variables, and maximize that!

$$l(\pi,\mu,\Sigma) = \sum_{i=1}^{m} \log(p(x_i|\pi,\mu,\Sigma))$$

But we also don't have our cluster assignments (i.e. which element of π,μ,Σ do we associate our x with)

Gaussian Mixture Model – Maximum Likelihood

Define the log-likelihood of all our variables, and maximize that!

$$l(\pi,\mu,\Sigma) = \sum_{i=1}^{m} \log(p(x_i|\pi,\mu,\Sigma))$$

$$l(\pi,\mu,\Sigma) = \sum_{i=1}^{m} log \sum_{k=1}^{K} p(x_i|z_k = k,\pi,\mu,\Sigma)\, p(z_k = k|\pi)$$

Gaussian Mixture Model – Maximum Likelihood

Define the log-likelihood of all our variables, and maximize that!

$$l(\pi,\mu,\Sigma) = \sum_{i=1}^{m} \log(p(x_i|\pi,\mu,\Sigma))$$

$$l(\pi,\mu,\Sigma) = \sum_{i=1}^{m} log \sum_{k=1}^{K} p(x_i|z_k = k,\pi,\mu,\Sigma)\, p(z_k = k|\pi)$$

But we can't vary our cluster assignments *and* our variables π,μ,Σ

Vary EITHER cluster OR π μ Σ

Gaussian Mixture Model – Expectation Maximization

Hold Z fixed, maximize our variables

$$r_{ik} = p(z_i = k \mid \mathbf{x}_i, \pi, \mu, \Sigma)$$

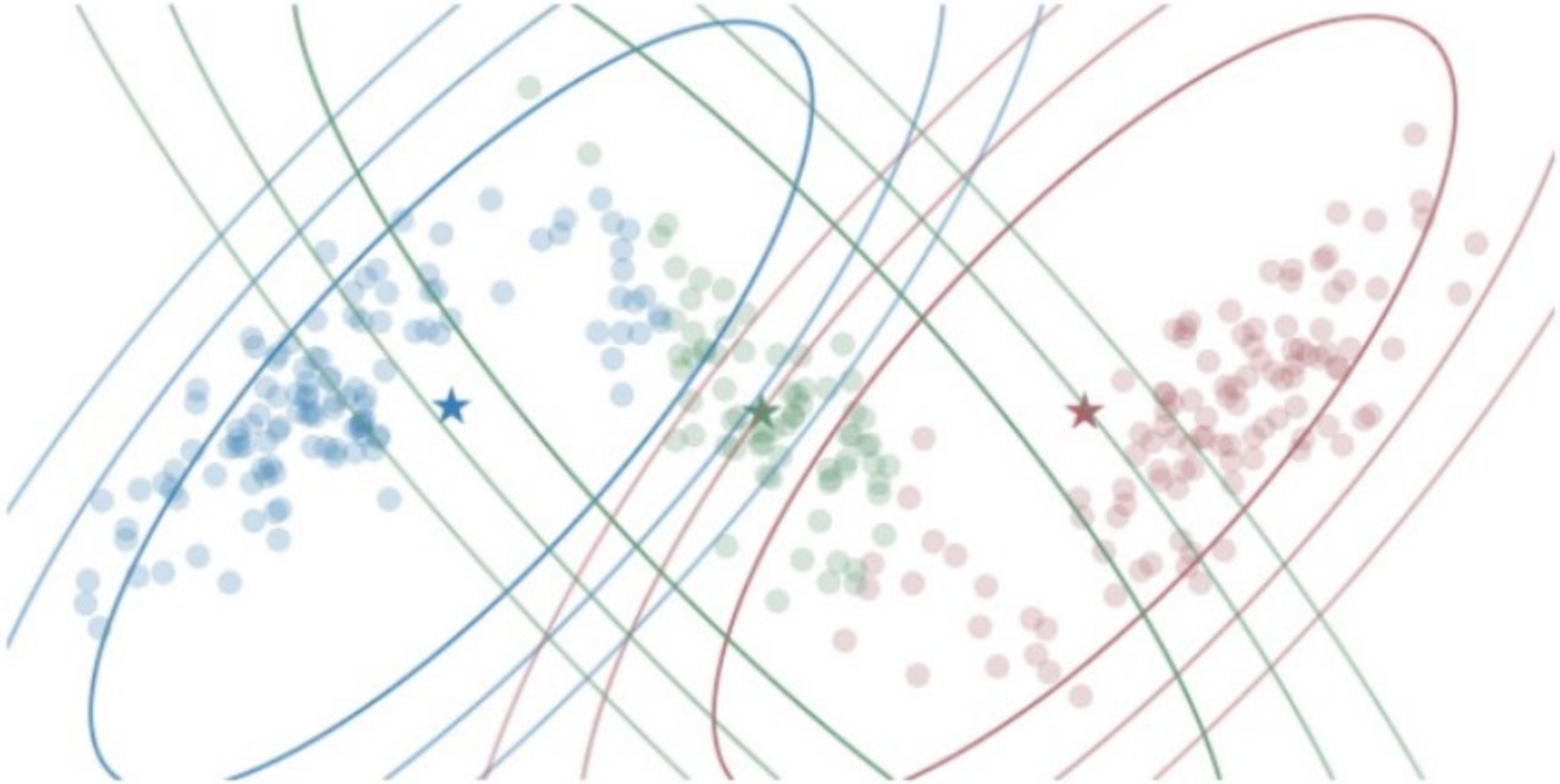
$$\pi_k = \frac{1}{m} \sum_{i=1}^{m} r_{ik}$$

$$\mu_k = \frac{\sum_{i=1}^{m} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{m} r_{ik}}$$

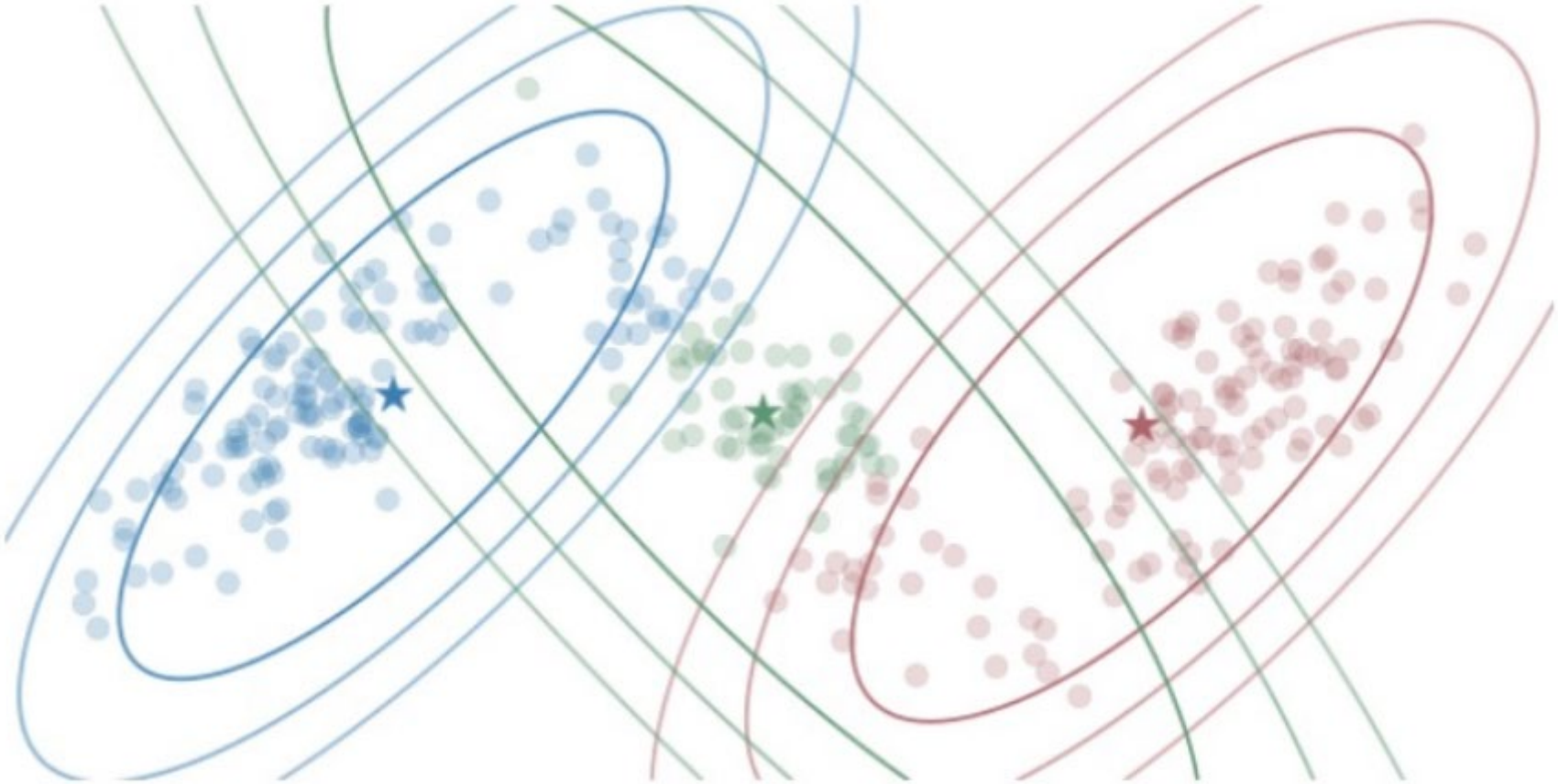
$$\Sigma_k = \frac{\sum_{i=1}^{m} r_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{m} r_{ik}}$$

GMM – EM in practice

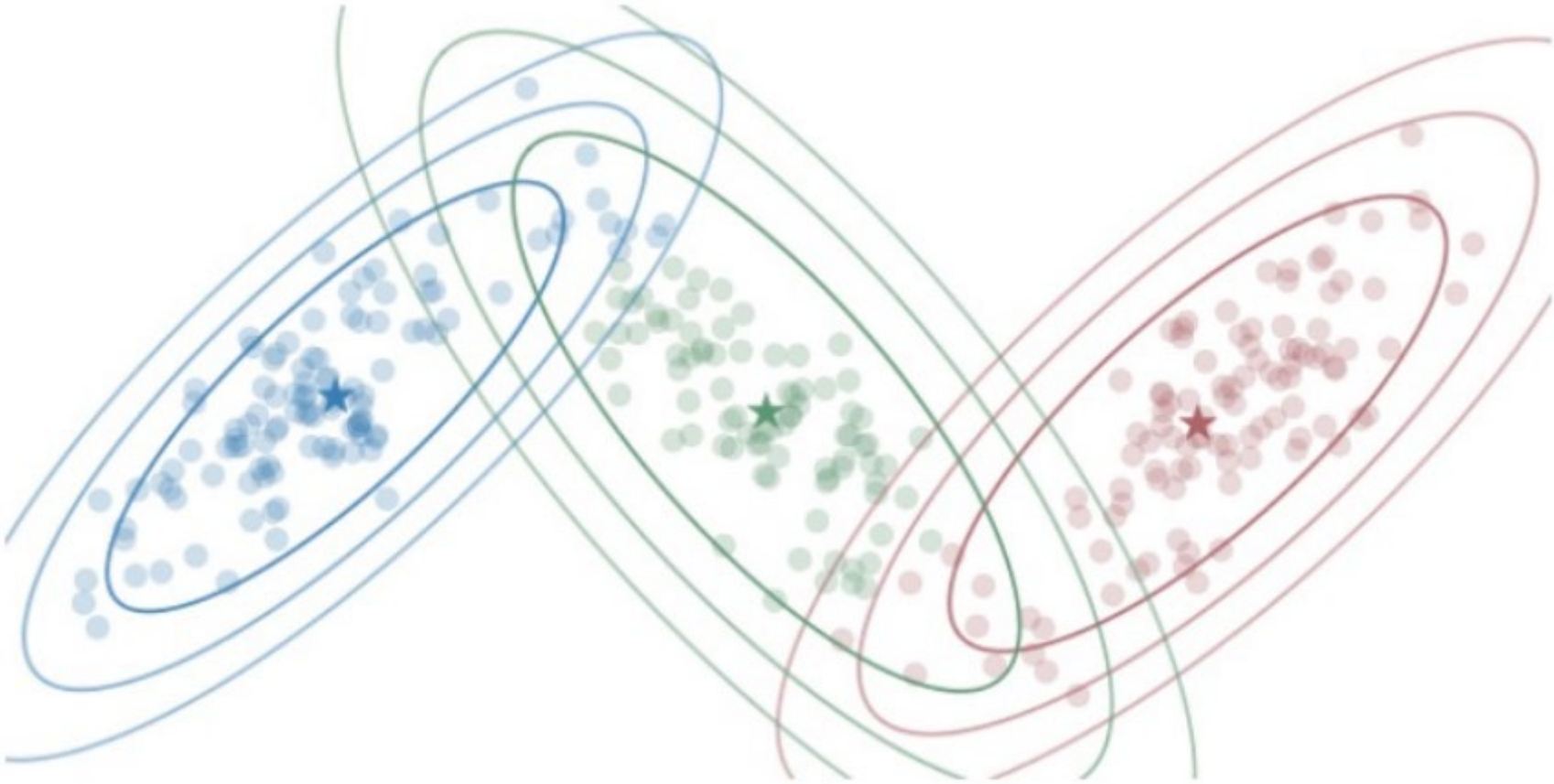
GMM – EM in practice



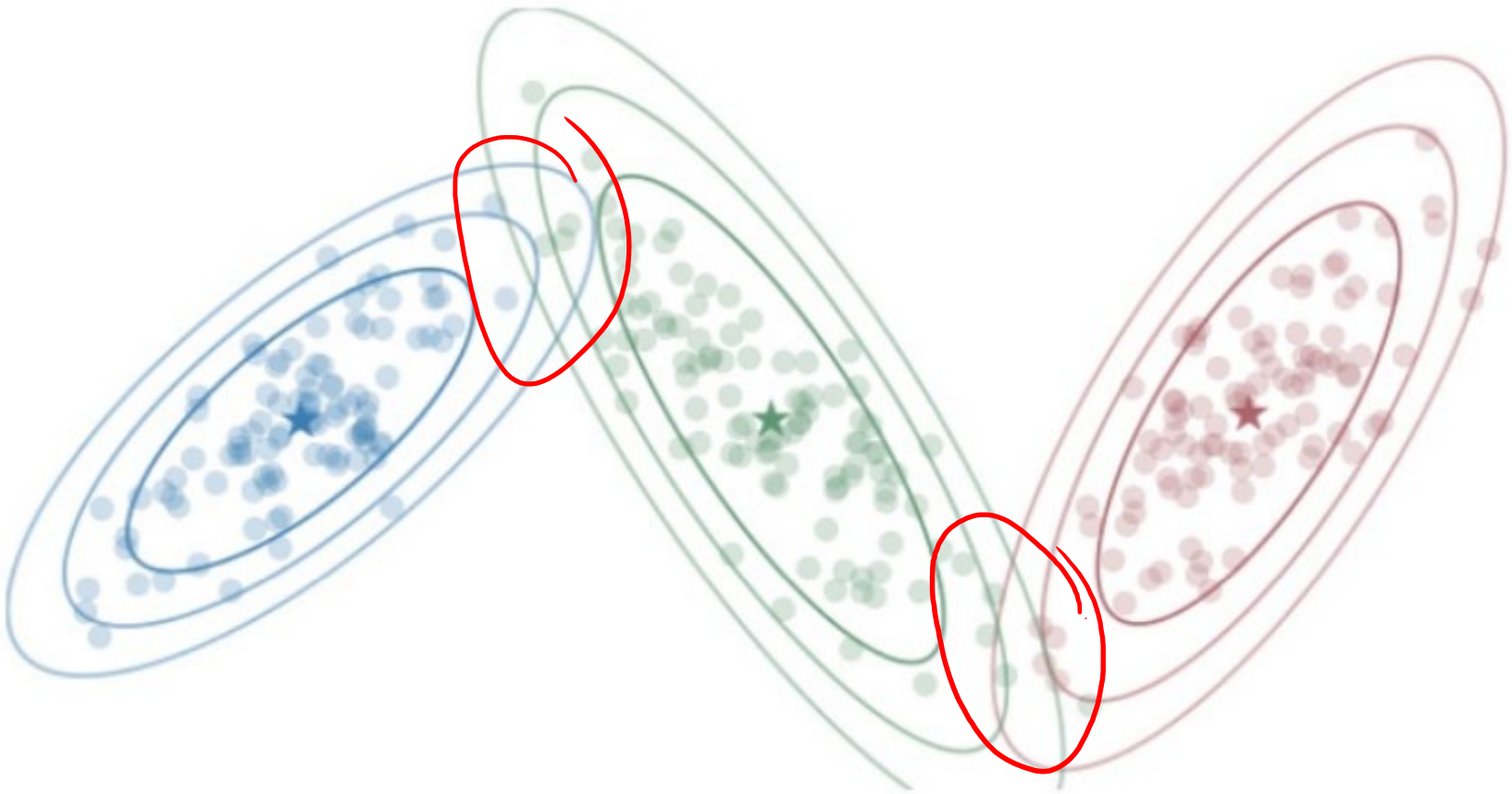
GMM – EM in practice



GMM – EM in practice



GMM – EM in practice



Principal Components / Singular Value Decomposition

Classification of these data?

- Can we predict?
  - Can we assign?
  - Can we identify?

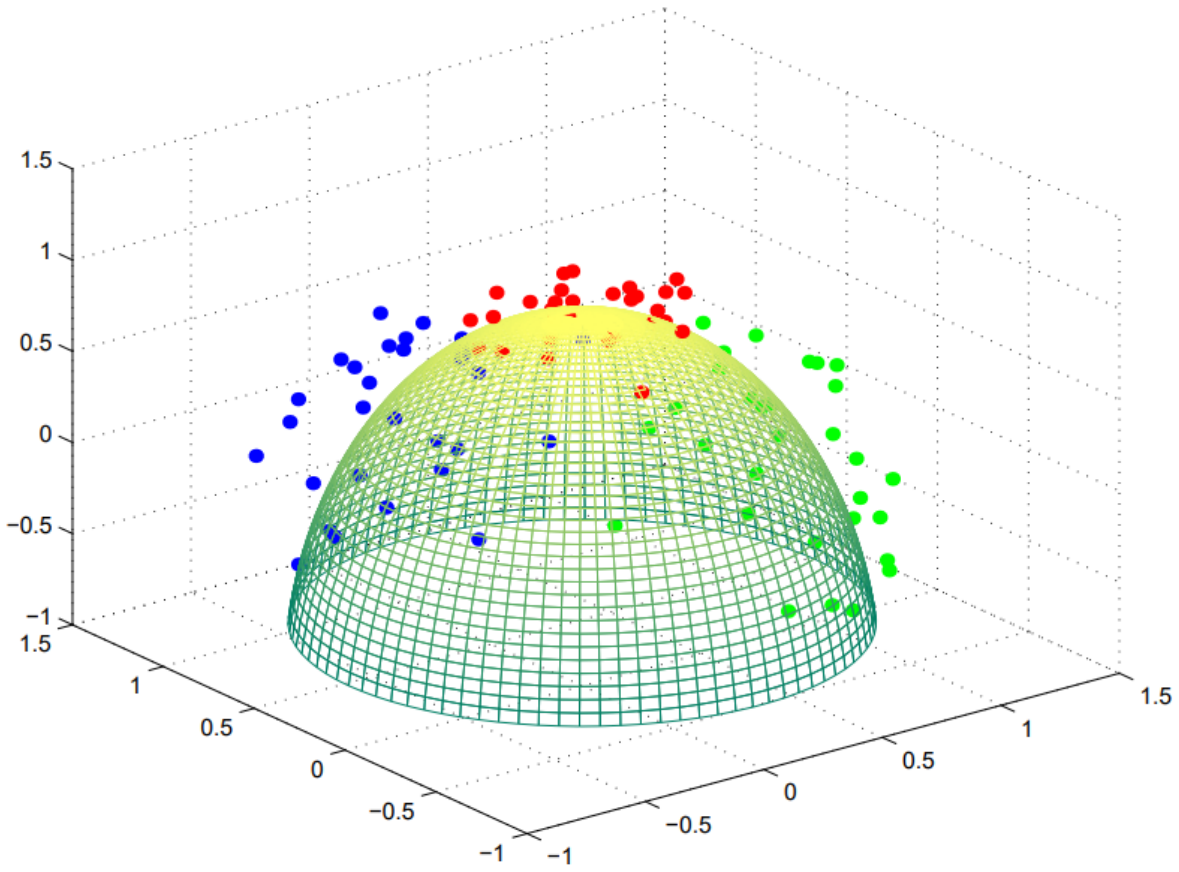


FIGURE 14.15. *Simulated data in three classes, near the surface of a half–sphere.*

Principal Components – A different objective

- What is unique?
  - What makes an X?

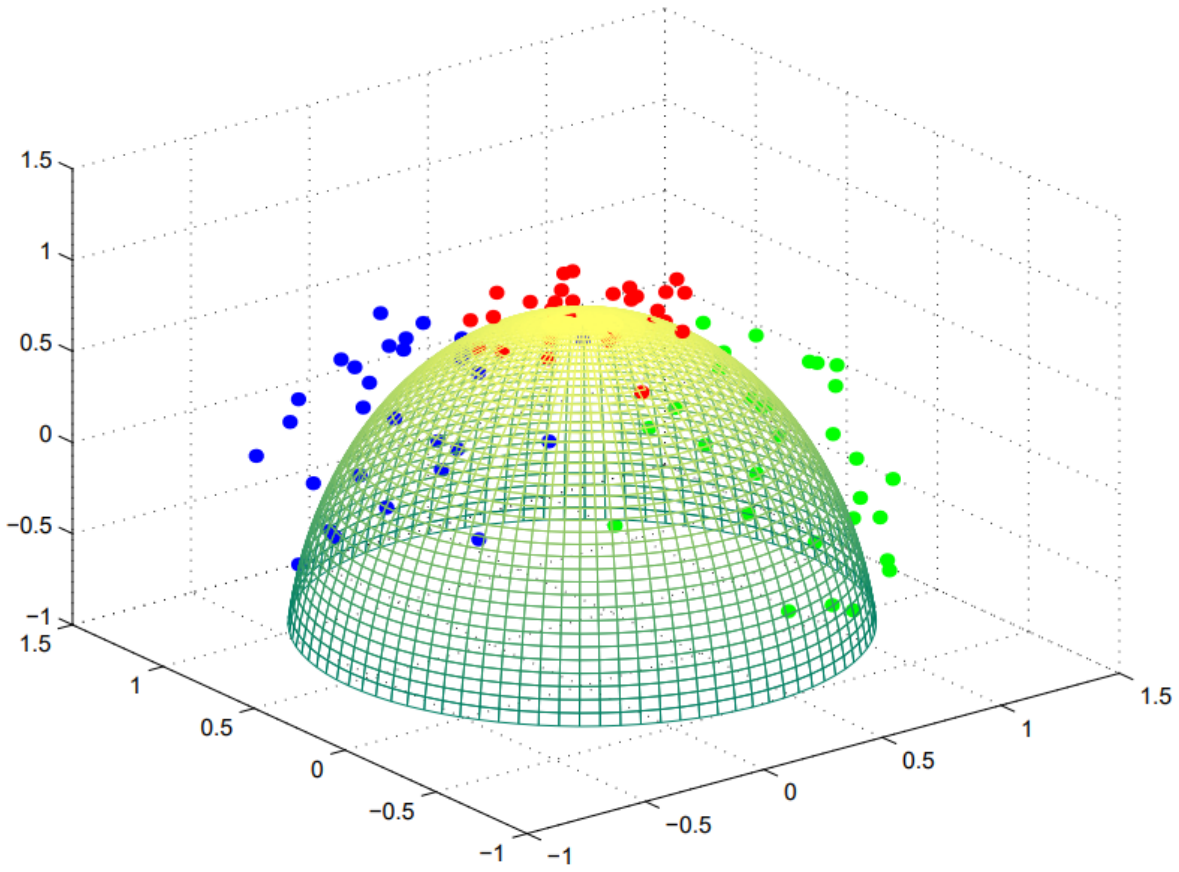
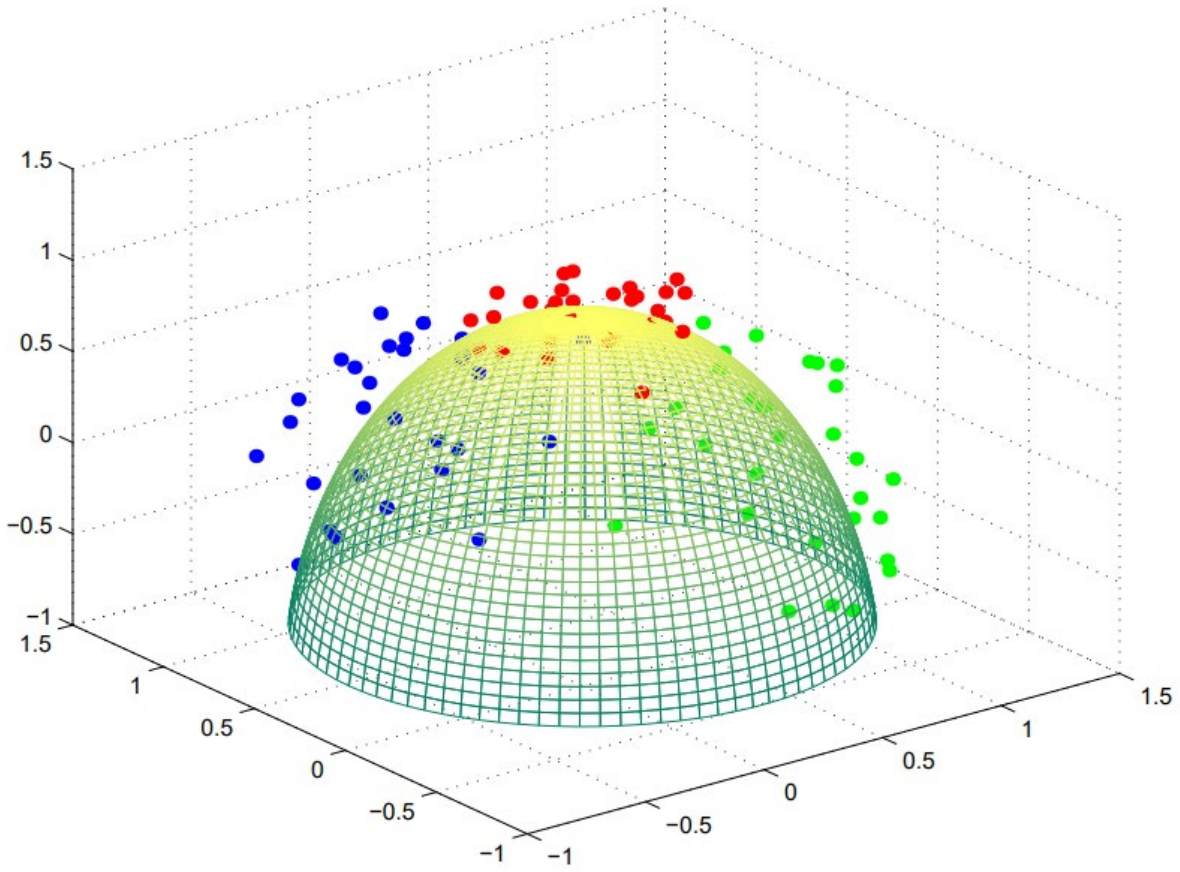
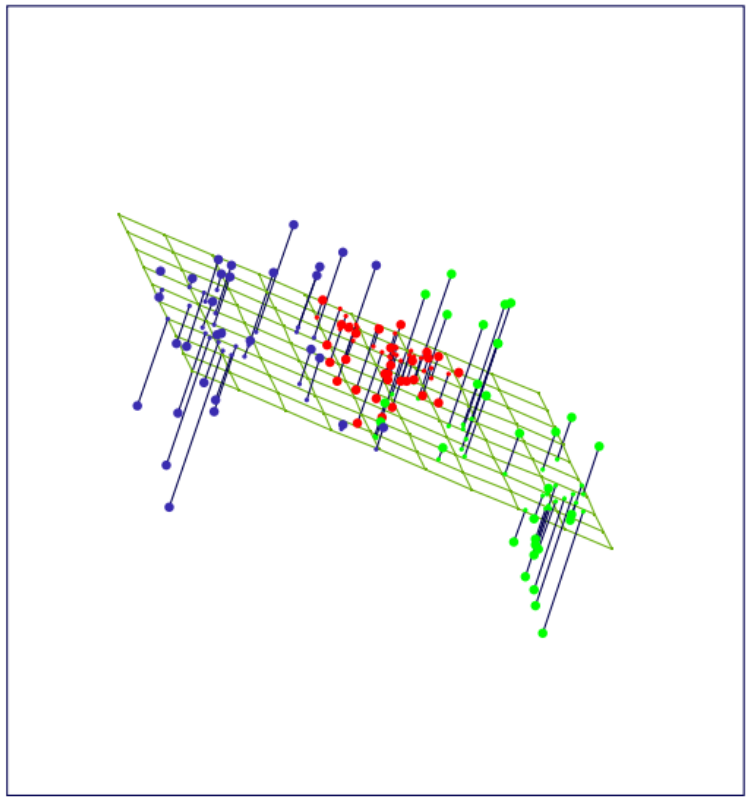


FIGURE 14.15. *Simulated data in three classes, near the surface of a half–sphere.*

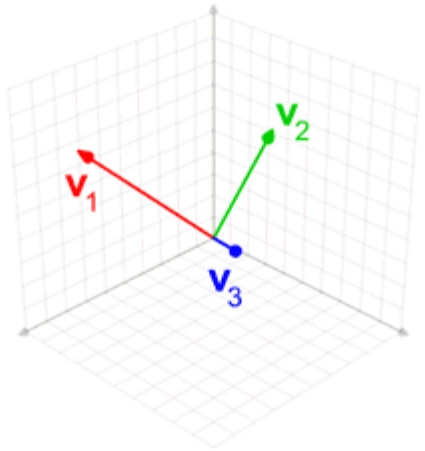
Step 1 – Build a Linear Model Approximation





- Implementation details are outlined in ESL Textbook

Recall : Gram-Schmidt Process

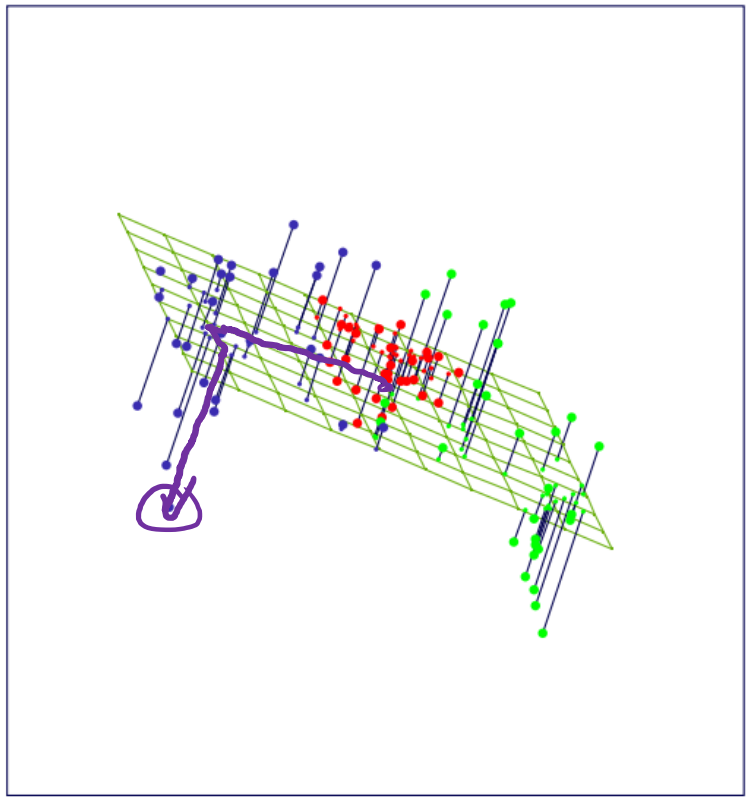


Step 1 – Build a Linear Model Approximation

- End with the *Singular Value Decomposition*

$$X = UDV^T$$

- X = our observations*
  - *N* (number of points) by *p* (dimensions)
- U = orthog. matrix, left singular vector columns
  - *N by p*
- D = diagonal matrix, singular values
  - *p by p*
- V^T = orthog. matrix, right singular vect. columns
  - *p by p*



**obs. have been re-centered to a new origin*

Step 2 – Extract the q principal components you seek

- $X_q = U_q D_q$

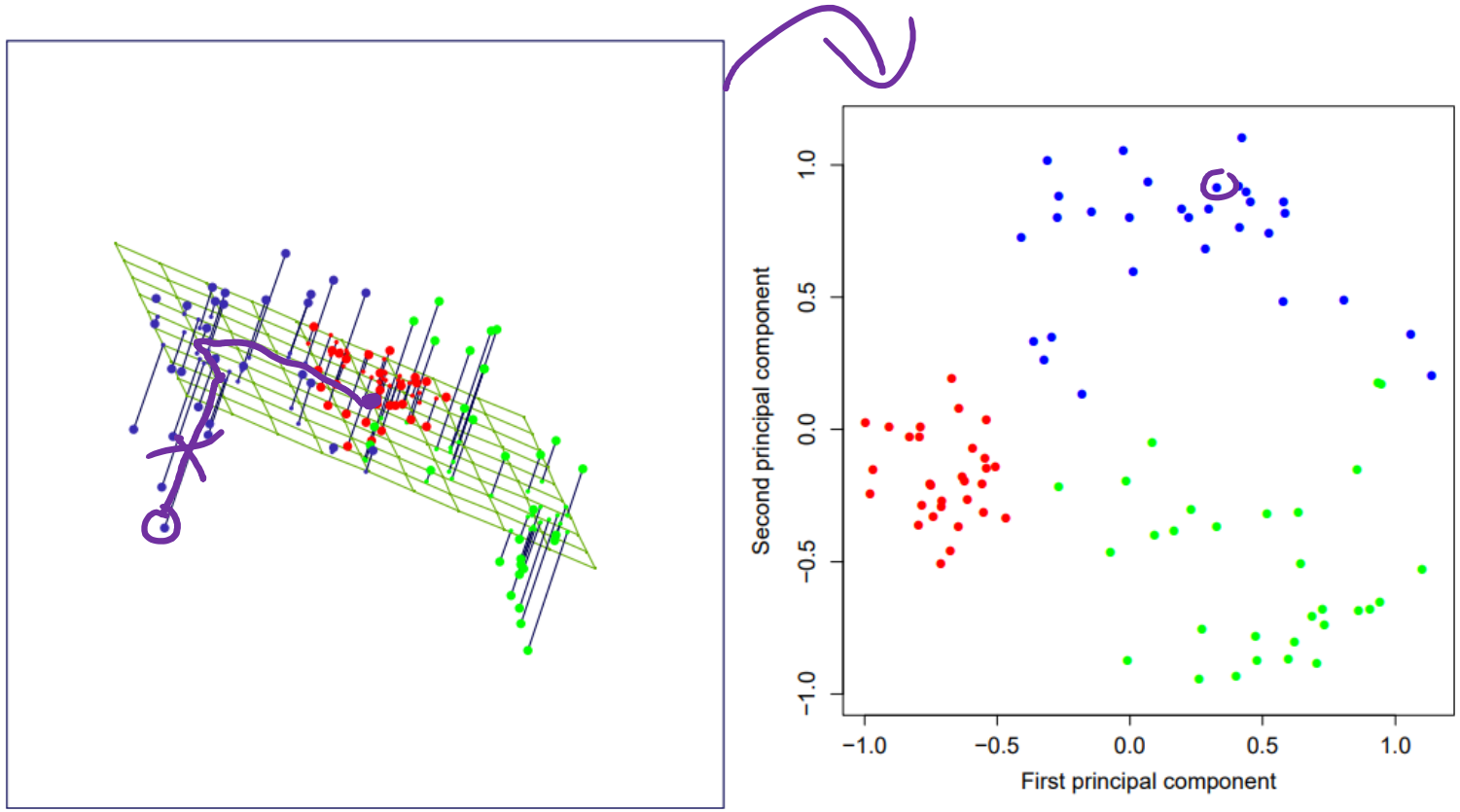


FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$*, the first two principal components of the data.*

PCA In Action

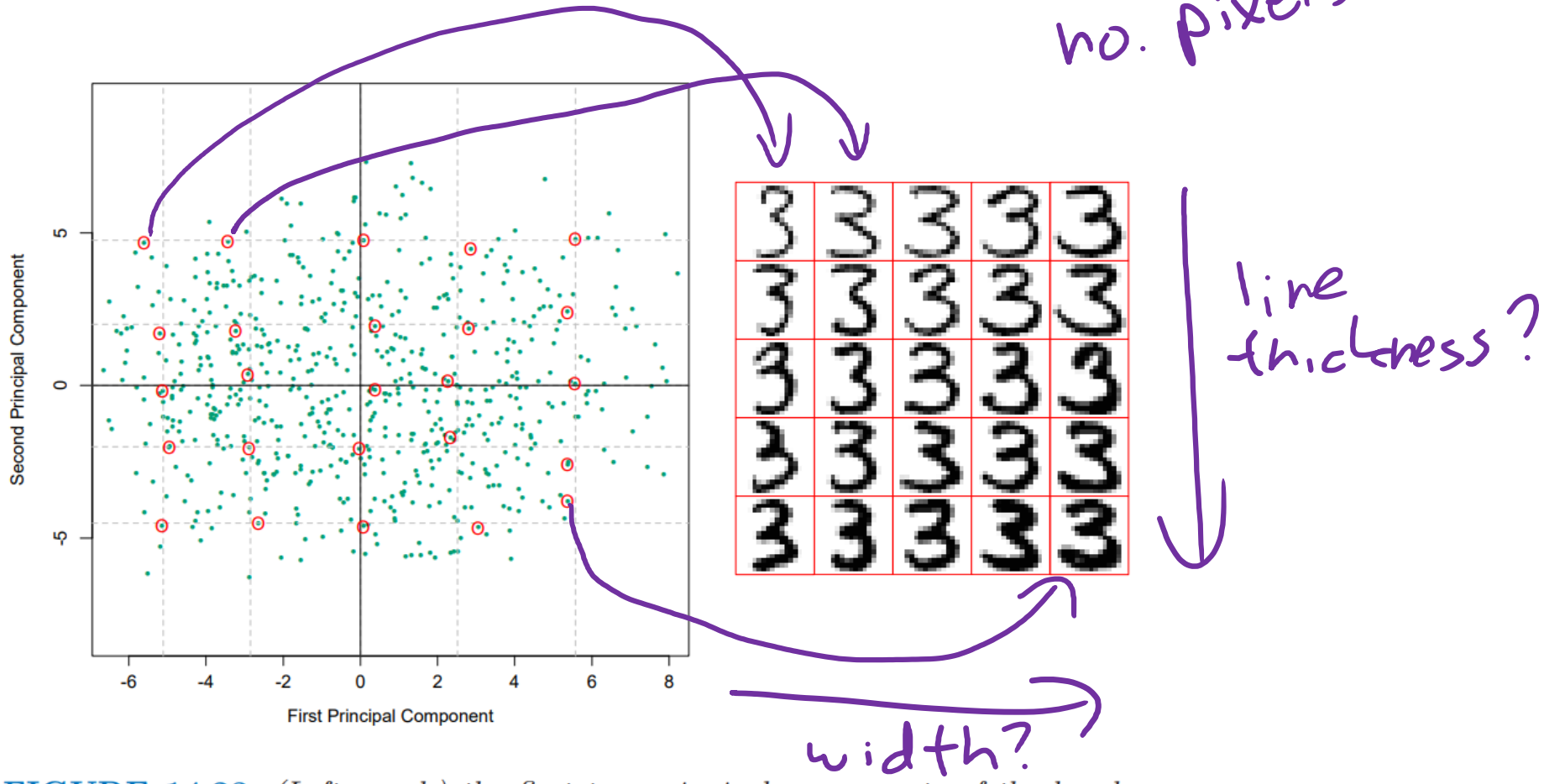
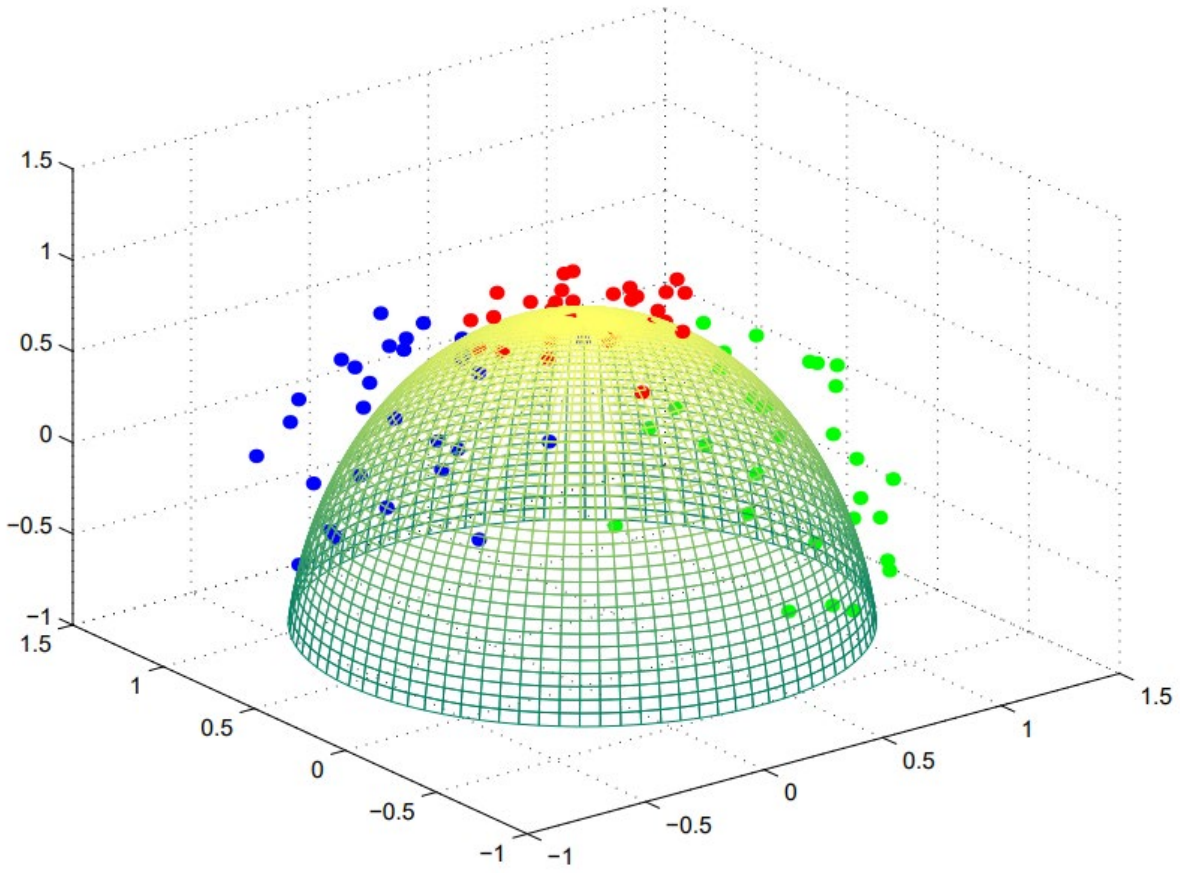
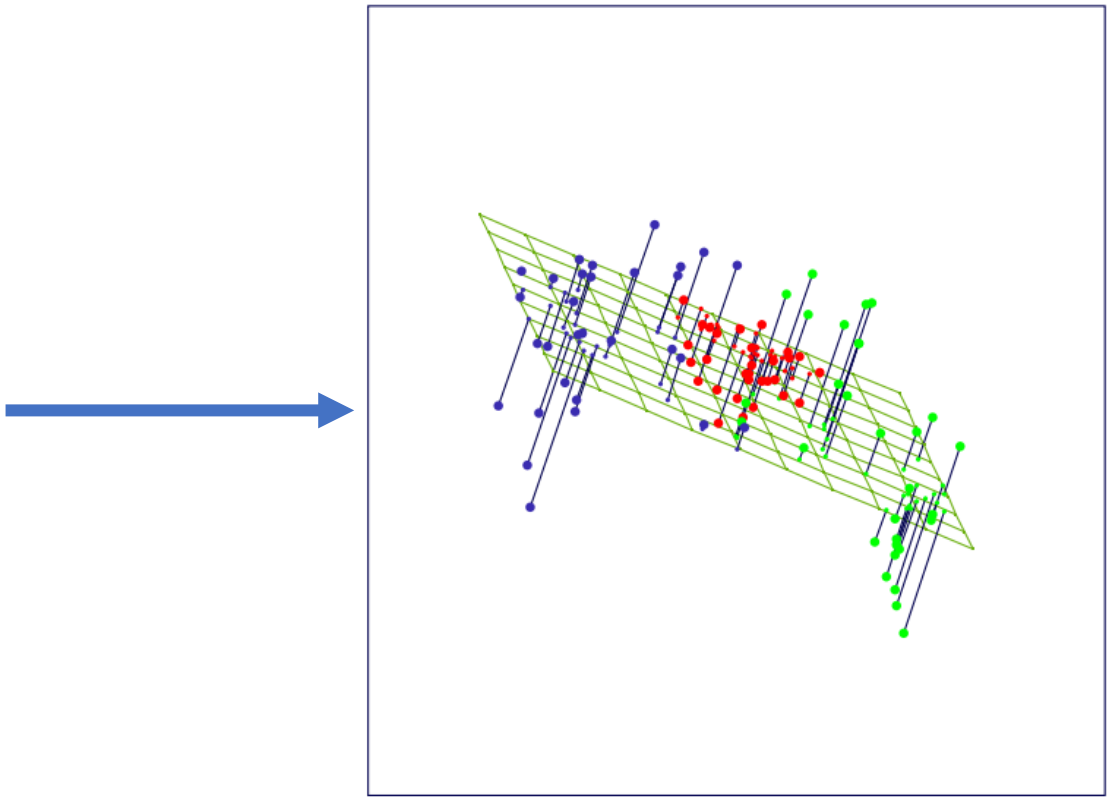


FIGURE 14.23. *(Left panel:) the first two principal components of the hand-written threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

Was this the best fit?





Principal Surface Analysis

- End objective – a smoothed curve that best approximates our data

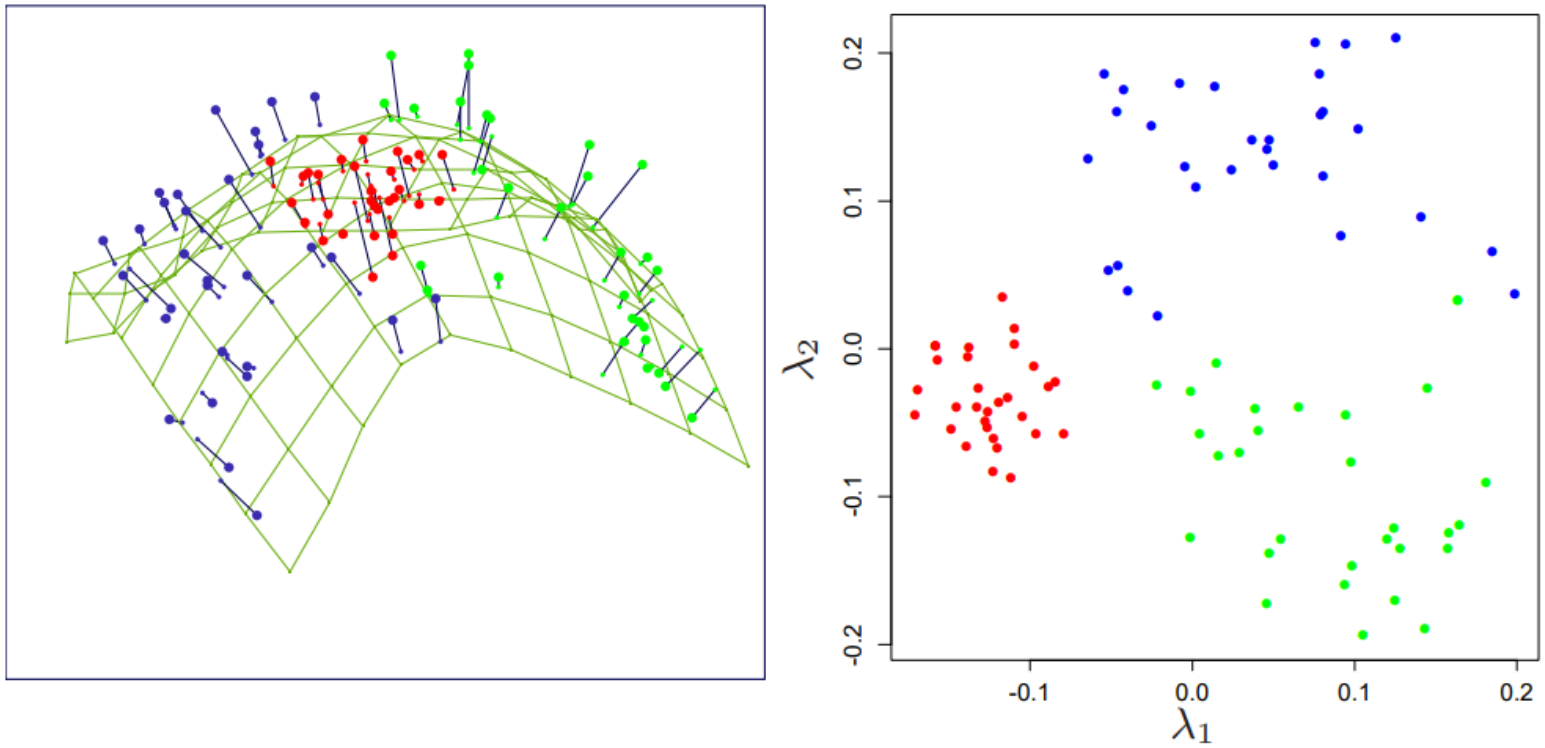
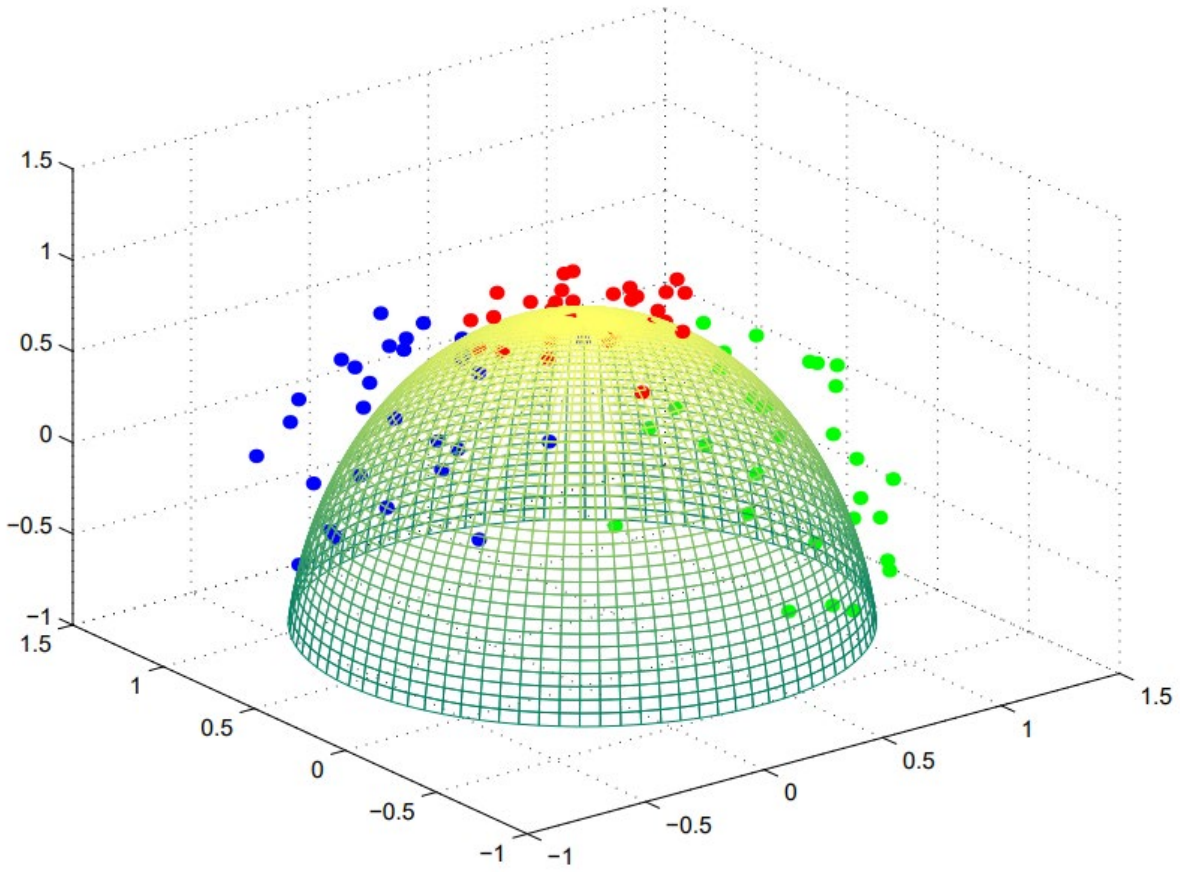
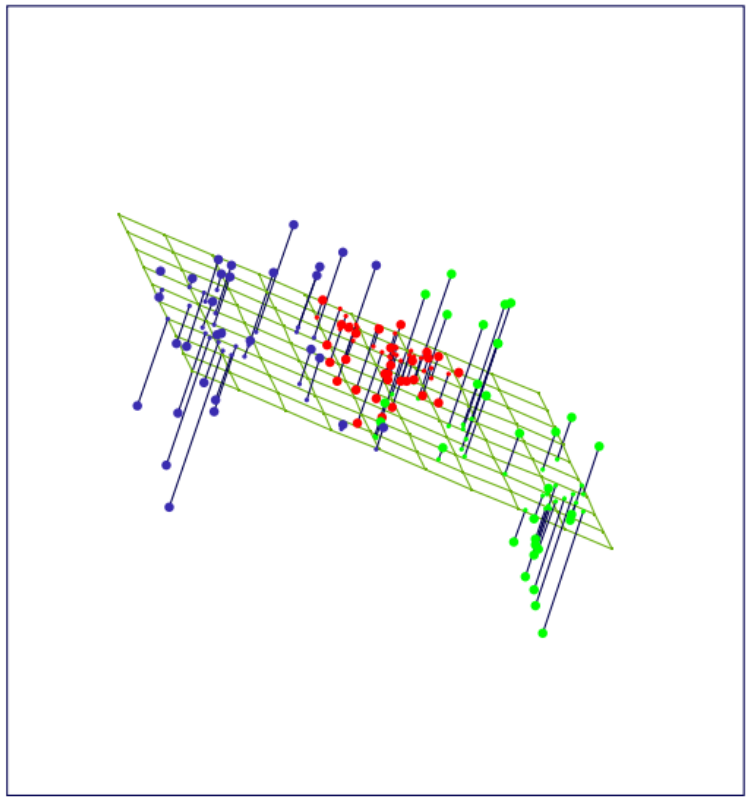


FIGURE 14.28. *Principal surface fit to half-sphere data. (Left panel:) fitted two-dimensional surface. (Right panel:) projections of data points onto the surface, resulting in coordinates* $\hat{\lambda}_1, \hat{\lambda}_2$.

Step 1 - Build a Linear Model Approximation





Step 2 – Bend it! (like Beckham?)

Iteratively:

Hold λ fixed, enforce self consistency on your curve

$$f_j(\lambda) \leftarrow E(X_j \,|\lambda(X) = \lambda); j = 1, 2, \ldots, p$$

Hold the curve fixed, minimize your λ (find the closest point on the curve for each data point)

$$\hat{\lambda}_f(x) \leftarrow argmin_{\lambda'} ||x - \hat{f}(\lambda')||2$$

Principal Surface Analysis

- End objective – a smoothed curve that best approximates our data

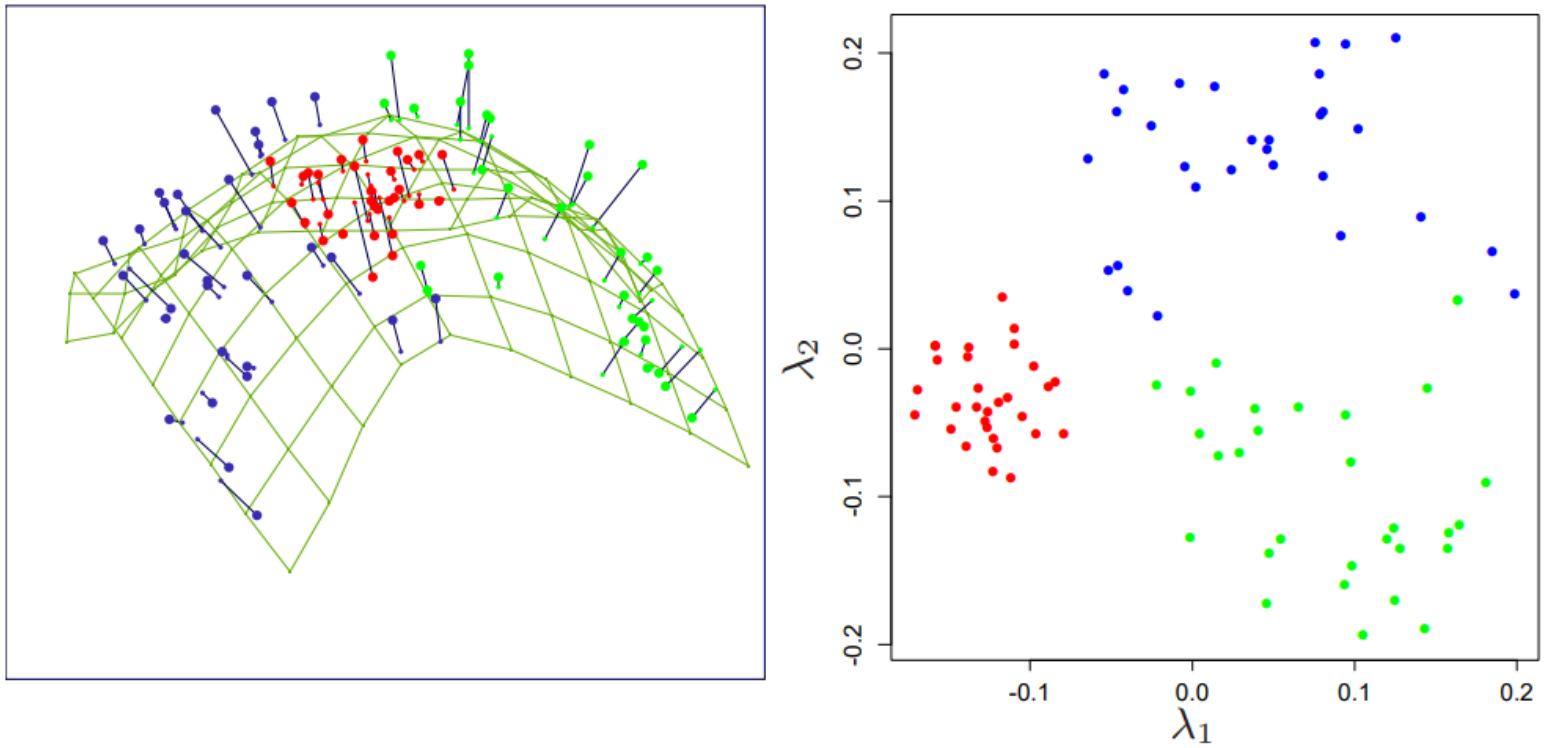
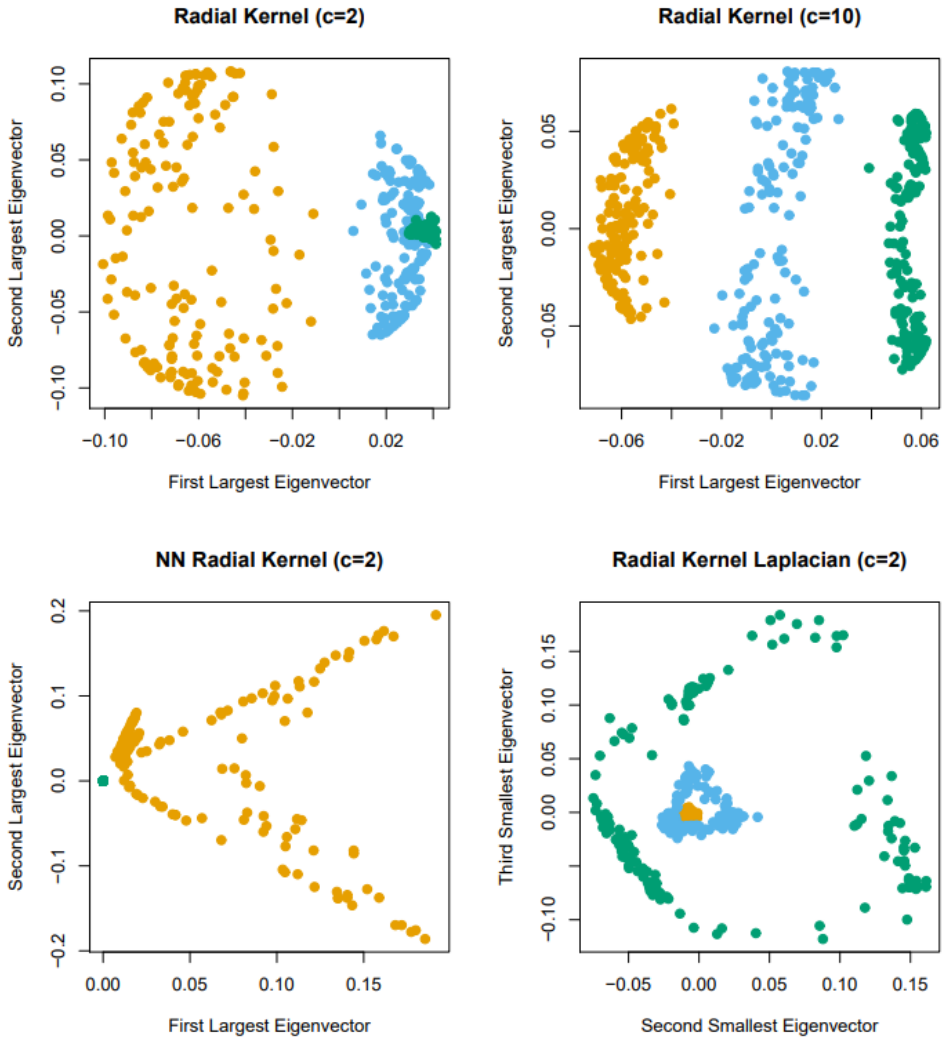


FIGURE 14.28. *Principal surface fit to half-sphere data. (Left panel:) fitted two-dimensional surface. (Right panel:) projections of data points onto the surface, resulting in coordinates* $\hat{\lambda}_1, \hat{\lambda}_2$.

Kernel PCA


Radial Kernel (c=2)
Second Largest Eigenvector
First Largest Eigenvector
Radial Kernel (c=10)
Second Largest Eigenvector
First Largest Eigenvector
NN Radial Kernel (c=2)
Second Largest Eigenvector
First Largest Eigenvector
Radial Kernel Laplacian (c=2)
Third Smallest Eigenvector
Second Smallest Eigenvector

An Application of Clustering

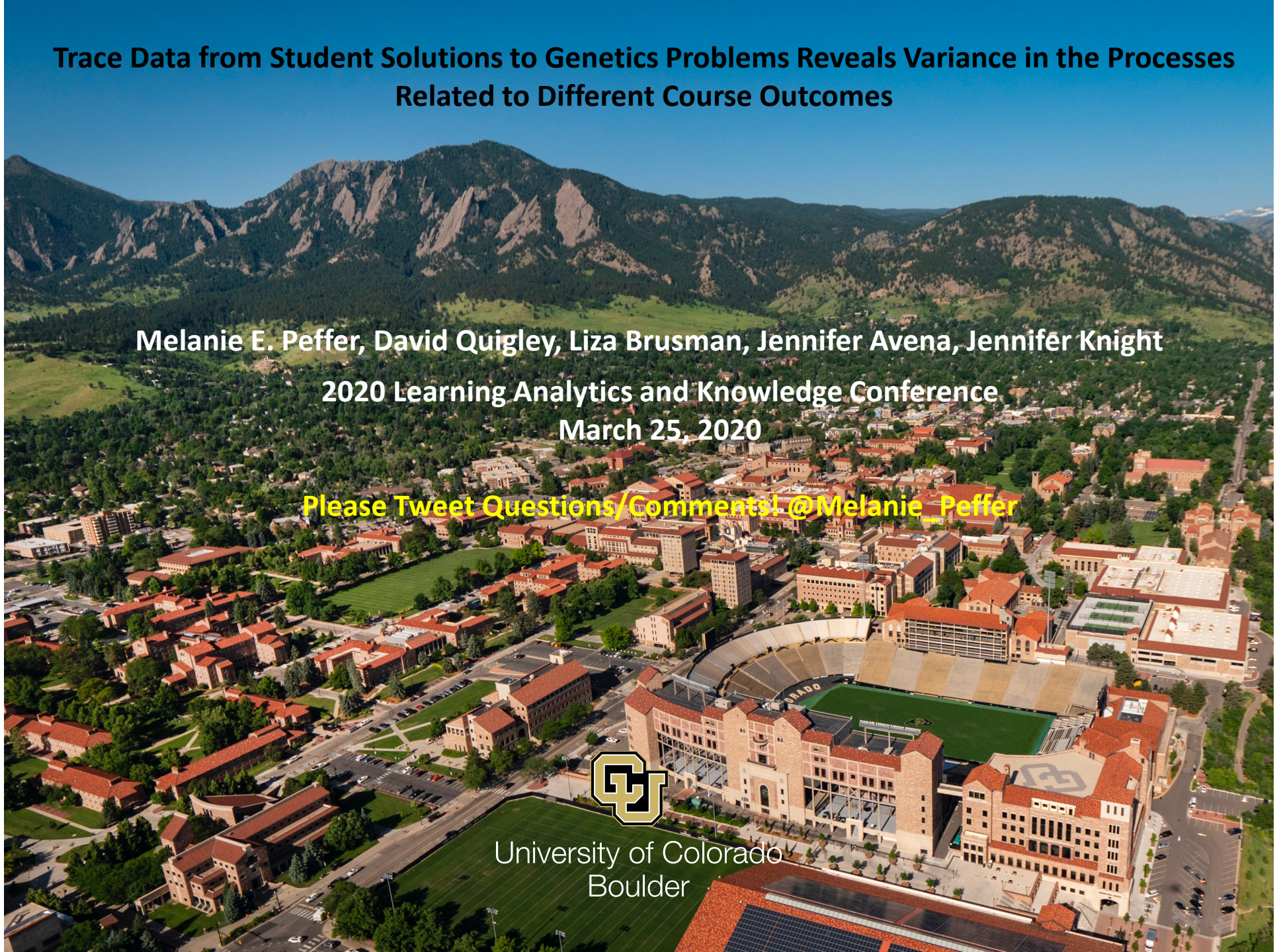
What I did last Summer

An Application of Clustering

~~What~~ One portion of one of the projects I did last ~~Summer~~ Fall

Presented at Learning Analytics & Knowledge 2020

Trace Data from Student Solutions to Genetics Problems Reveals Variance in the Processes Related to Different Course Outcomes



Melanie E. Peffer, David Quigley, Liza Brusman, Jennifer Avena, Jennifer Knight

2020 Learning Analytics and Knowledge Conference
March 25, 2020

Please Tweet Questions/Comments! @Melanie_Peffer

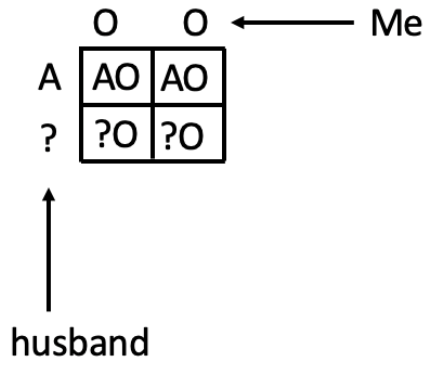
University of Colorado
Boulder

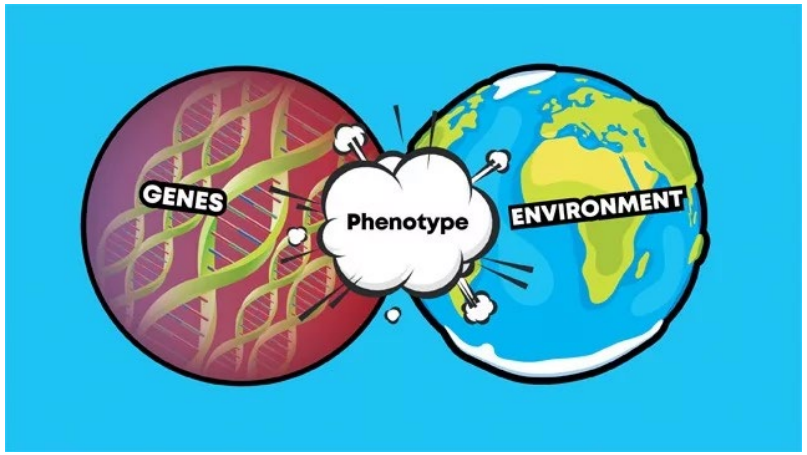
Problem Solving is Integral to Genetics Literacy

- Problem Solving: process of finding a solution to an unstructured or complex situation

- Genetics relies heavily on problem solving
  - No longer limited to classrooms

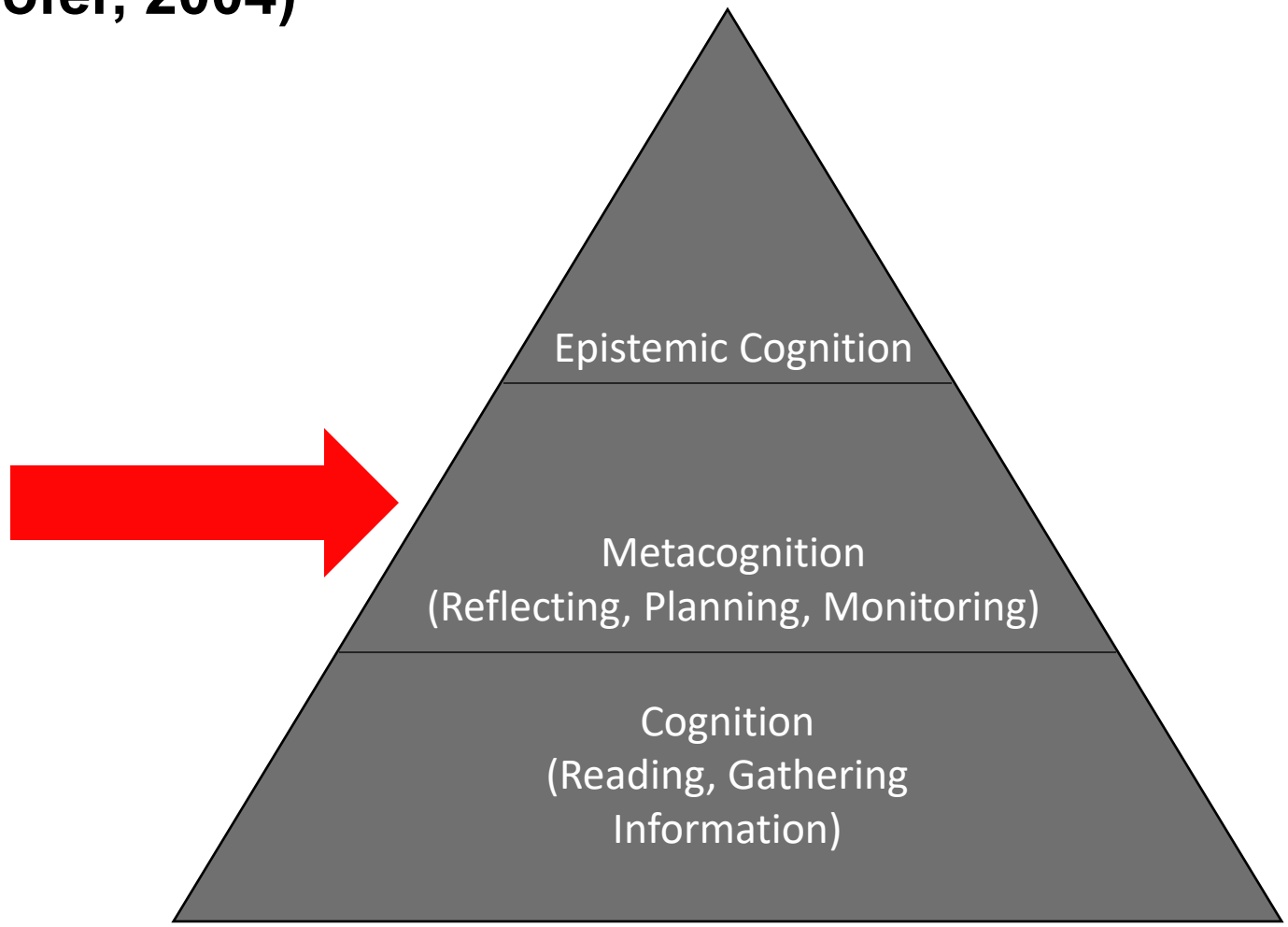
- Problem solving tends to be difficult for students
  - Genetics problem solving is particularly challenging





https://www.technologynetworks.com/genomics/articles/genotype-vs-phenotype-examples-and-definitions-318446

Three Level Model of Processing During Problem Solving (Hofer, 2004)

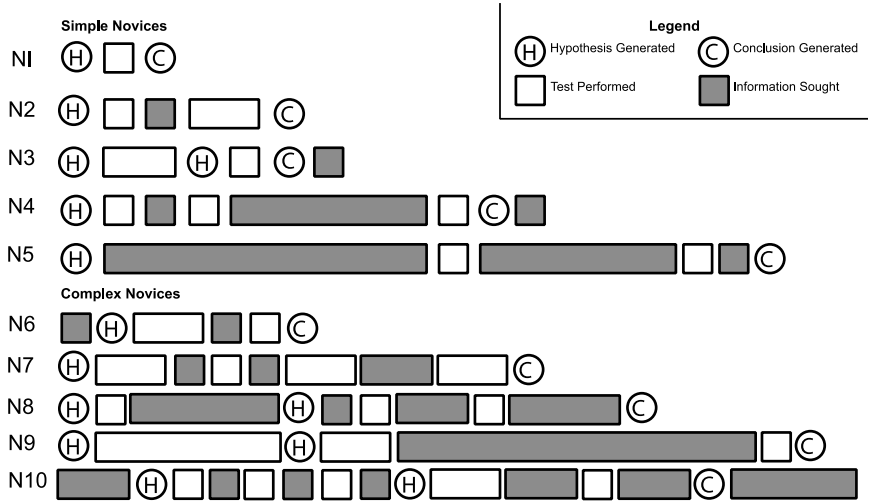


Technological Solutions to Measuring Latent Constructs

- Measurement of cognition via student traces

- Epistemological beliefs about science as seen through inquiry practices
  - Expert/Novice studies of inquiry (Peffer & Ramezani, 2019)

- Use of learning analytics for assessment of epistemological beliefs
  - NLP of discourse during inquiry (Peffer & Kyle, 2017; LAK ‘17 Best Paper Nominee)
  - K-means clustering of inquiry profiles (Peffer, Quigley, and Mowstofi, 2019; LAK ‘19)



Current Study

- Working knowledge of genetics and ability to solve genetics problems is crucial in today's society
- Metacognition is a crucial skill in successful problem solving, but is difficult to measure
- Prior work suggests that trace data can be used to understand latent constructs
- **Can trace data reveal cognitive processes in action during genetics problem solving?**

Methods

- Participants
  - n = 295 undergraduate introductory genetics students

- Intervention
  - Extra credit genetics problems
  - “Think aloud” protocol while solving

- Outputs
  - Problem correctness
  - Genetics Concept Assessment (Smith, Wood, and Knight, 2008)
  - Final exam score

Analysis

- Coding
  - Emergent coding approach
  - Identified 19 processes that fell under 5 categories

Methods

Category	Subcategories
Conclusion	Eliminating, Claiming
Metacognition	Monitoring, Checking, Planning
Orientation	Clarifying, Recalling Information, Identifying similarities between question types
Reasoning	Justifying answers with evidence, reasoning
Execution	Using Information, Calculating, Creating a visual representation, Restating the process

Analysis

- Coding
  - Emergent coding approach
  - Identified 19 processes that fell under 5 categories

- Feature Transformation
  - Each process assigned a single letter code (CMORE)
  - Sequences of processes generated
  - Built counts of all possible action bigrams to capture sequences

- Cluster Analysis
  - K-means clustering using Weka
  - Built clusters using bigram features
  - Cluster assignment related to outcomes

Bigram Clustering Reveals Differences in Processes and Probability of Answering Problems Correctly

Feature	
CC	
CM	
CO	
CR	
CE	
MC	
MM	
MO	
MR	
ME	
OC	
OM	
OO	
OR	
OE	
RC	
RM	
RO	
RR	
RE	
EC	
EM	
EO	
ER	
EE	
Count	
Probability of Problem Correctness	
Average Score on GCA	
Average Score on FE	

C=conclusion, M=metacognition, O=Orientation, R=Reasoning, E=Execution

Bigram Clustering Reveals Differences in Processes and Probability of Answering Problems Correctly

Feature	Cluster 1 Average	Cluster 2 Average	Cluster 3 Average
CC			
CM			
CO			
CR			
CE			
MC			
MM			
MO			
MR			
ME			
OC			
OM			
OO			
OR			
OE			
RC			
RM			
RO			
RR			
RE			
EC			
EM			
EO			
ER			
EE			
Count			
Probability of Problem Correctness			
Average Score on GCA			
Average Score on FE			

C=conclusion, M=metacognition, O=Orientation, R=Reasoning, E=Execution

Bigram Clustering Reveals Differences in Processes and Probability of Answering Problems Correctly

Feature	Cluster 1 Average	Cluster 2 Average	Cluster 3 Average
CC	0.12		
CM	0.26		
CO	0.07		
CR	0.44		
CE	0.14		
MC	0.11		
MM	0.09		
MO	1.41		
MR	0.15		
ME	0.62		
OC	0.35		
OM	0.87		
OO	1.2		
OR	0.31		
OE	0.89		
RC	0.45		
RM	0.28		
RO	0.06		
RR	0.09		
RE	0.85		
EC	0.56		
EM	0.79		
EO	0.30		
ER	0.84		
EE	1.13		
Count	159		
Probability of Problem Correctness	0.67		
Average Score on GCA	76.62		
Average Score on FE	75.10		

C=conclusion, M=metacognition, O=Orientation, R=Reasoning, E=Execution

Bigram Clustering Reveals Differences in Processes and Probability of Answering Problems Correctly

Feature	Cluster 1 Average	Cluster 2 Average	Cluster 3 Average
CC	0.12	0.12	
CM	0.26	0.03	
CO	0.07	0.03	
CR	0.44	0.63	
CE	0.14	0.22	
MC	0.11	0.05	
MM	0.09	0.02	
MO	1.41	0.14	
MR	0.15	0.04	
ME	0.62	0.06	
OC	0.35	0.16	
OM	0.87	0.10	
OO	1.2	0.30	
OR	0.31	0.19	
OE	0.89	0.35	
RC	0.45	1.04	
RM	0.28	0.04	
RO	0.06	0.03	
RR	0.09	0.12	
RE	0.85	0.29	
EC	0.56	0.11	
EM	0.79	0.06	
EO	0.30	0.12	
ER	0.84	0.65	
EE	1.13	0.54	
Count	159	643	
Probability of Problem Correctness	0.67	0.64	
Average Score on GCA	76.62	73.34	
Average Score on FE	75.10	72.41	

C=conclusion, M=metacognition, O=Orientation, R=Reasoning, E=Execution

Bigram Clustering Reveals Differences in Processes and Probability of Answering Problems Correctly

Feature	Cluster 1 Average	Cluster 2 Average	Cluster 3 Average
CC	0.12	0.12	0.02
CM	0.26	0.03	0.02
CO	0.07	0.03	0.01
CR	0.44	0.63	0.14
CE	0.14	0.22	0.07
MC	0.11	0.05	0.00
MM	0.09	0.02	0.01
MO	1.41	0.14	0.05
MR	0.15	0.04	0.02
ME	0.62	0.06	0.15
OC	0.35	0.16	0.01
OM	0.87	0.10	0.09
OO	1.2	0.30	0.39
OR	0.31	0.19	0.10
OE	0.89	0.35	0.70
RC	0.45	1.04	0.04
RM	0.28	0.04	0.01
RO	0.06	0.03	0.04
RR	0.09	0.12	0.07
RE	0.85	0.29	1.28
EC	0.56	0.11	1.02
EM	0.79	0.06	0.07
EO	0.30	0.12	0.21
ER	0.84	0.65	1.15
EE	1.13	0.54	1.30
Count	159	643	712
Probability of Problem Correctness	0.67	0.64	0.56
Average Score on GCA	76.62	73.34	73.59
Average Score on FE	75.10	72.41	72.59

C=conclusion, M=metacognition, O=Orientation, R=Reasoning, E=Execution