STA 9890 - Statistical Learning for Data Mining

In-Class Test 3: Unsupervised Learning

This is a closed-note, closed-book exam.
You may not use any external resources.

Name: _____

Kaggle User Name: _____

Instructions

This exam will be graded out of **100 points**.

This exam is divided into three sections:

- True/False (24 points; 12 questions at two points each)
- Short Answer (36 points; 9 questions at 4 points each)
- Applications of Machine Learning (40 points; two open-ended questions)

You have 90 minutes to complete this exam from the time the instructor says to begin. The instructor will give time warnings at: 30 minutes, 15 minutes, 5 minutes, and 1 minute.

When the instructor announces the end of the exam, you must stop **immediately**. Continuing to work past the time limit may be considered an academic integrity violation.

Write your name on the line above *now* before the exam begins.

Each question has a dedicated answer space. Place all answers in the relevant spot. Answers that are not clearly marked in the correct location **will not** receive full credit. Partial credit may be given at the instructor's discretion.

Mark and write your answers clearly: if I cannot easily identify and read your intended answer, you will not get credit for it.

Additional pages for scratch work are included at the end of the exam packet.

This is a closed-note, closed-book exam.
You may not use any external resources.

True/False: 24 points total at 2 points each

For each question, **CIRCLE** your answer(s).

TF1. True/False: PCA can be applied without standardizing the data first.

TRUE FALSE

TF2. True/False: Sparse PCA can be used to induce *sparsity* in the rows, in the columns, or in both.

TRUE FALSE

TF3. True/False: The eigenvalues of a real-valued matrix are always non-negative.

TRUE FALSE

TF4. True/False: t-SNE is a linear dimension reduction technique.

TRUE FALSE

TF5. True/False: In *consensus clustering*, the cluster centroids are averaged over re-sampling to get a sense of cluster uncertainty.

TRUE FALSE

TF6. True/False: Lloyd's algorithm is not guaranteed to find the optimal clustering, except when initialized with K-means++.

TRUE FALSE

TF7. True/False: DBSCAN (Density-Based Clustering) and Spectral Clustering (via graph embeddings) can identify clusters of arbitrary (connected) shape.

TRUE FALSE

TF8. True/False: Complete linkage in hierarchical clustering is prone to a "chaining" effect (long stringy clusters).

TRUE FALSE

TF9. True/False: The "Elbow Plot" method is used to select the number of clusters that minimizes the within-cluster sum-of-squares.

TRUE FALSE

TF10. True/False: The sum of the squares of the singular values of a data matrix equals the total variance in that data.

TRUE FALSE

TF11. True/False: UMAP solutions are *nested* in that the k-dimensional projection can be found by taking the first k dimensions of the $k+1$-dimensional projection.

TRUE FALSE

TF12. True/False: A high silhouette score suggests that a data point is well clustered.

TRUE FALSE

Short Answer: 36 points total at 4 each

SA1. Suppose we perform PCA on a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ using the SVD of $\boldsymbol{X}$. Which of the following are true? Select all that apply.

a. ☐ The row patterns are orthogonal to the column patterns: $\boldsymbol{u}_i^\top \boldsymbol{v}_i = 0$ for all i

b. ☐ The scale factors d_i may be positive or negative.

c. ☐ The row patterns are orthogonal to each other $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ for $i \neq j$

d. ☐ The row patterns are orthogonal to residual matrix: $\boldsymbol{u}_i^\top \boldsymbol{X}_i = \boldsymbol{0}$ if $\boldsymbol{R}_i$ is the PCA residuals after the first i principal components are removed from $\boldsymbol{X}$

e. ☐ The units of $\boldsymbol{X}$ are comparable to those of the PCA term $\hat{\boldsymbol{X}}_i = d_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ for all i

f. ☐ The column patterns are orthogonal to each other $\boldsymbol{v}_i^\top \boldsymbol{v}_j = 0$ for $i \neq j$

SA2. *K-medoids* clustering is an analogue of K-means clustering, but it uses the *medoid*, a p-dimensional generalization of the median, instead of the mean as the cluster center.

Give one strength/advantage and one weakness/disadvantage of K-medoids as compared to K-means. Explain why these differences arise (you can't just state them).

Strength:

Weakness:

SA3. For each of the following unsupervised methods, mark L, G or M to indicate whether they are principally concerned with local or global structure or whether they attempt to balance a mixture of the two. *Hint*: there are 4 G, 3 L, and 1 M.

____ PCA

____ t-SNE

____ K-Means

____ Clustering via NN-Graph Embedding

____ Convex Clustering

____ UMAP

____ Density-Based Clustering

____ (Euclidean) Hierarchical Clustering

SA4. Suppose that the following data set $\boldsymbol{X} \in \mathbb{R}^{6\times 3}$ is observed:

$$\boldsymbol{X} = \begin{pmatrix} 22 & 38 & -41 \\ 18 & 42 & -39 \\ 28 & 44 & -32 \\ -18 & -42 & 39 \\ -12 & -36 & 48 \\ -22 & -38 & 41 \end{pmatrix} = \begin{pmatrix} \sqrt{6}/6 & 0 & 1/2 \\ \sqrt{6}/6 & 0 & -1/2 \\ \sqrt{6}/6 & \sqrt{2}/2 & 0 \\ -\sqrt{6}/6 & 0 & 1/2 \\ -\sqrt{6}/6 & \sqrt{2}/2 & \\ -\sqrt{6}/6 & 0 & -1/2 \end{pmatrix} \begin{pmatrix} 60\sqrt{6} & 0 & 0 \\ 0 & 12\sqrt{2} & 0 \\ 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} 1/3 & 2/3 & 2/3 \\ -2/3 & -1/3 & 2/3 \\ -2/3 & 2/3 & -1/3 \end{pmatrix}^{\top}$$

where the decomposition on the right is the *singular value decomposition* of $\boldsymbol{X}$.

What is the (marginal) percent of variance explained by the second principal component of $\boldsymbol{X}$? You may leave your answer unsimplified in terms of square roots and fractions.

SA5. Describe how PCA could be used as part of a *data imputation* process.

SA6. Write out Lloyd's Algorithm for K-means clustering. (Your answer does not need to be hyper-precise. Basic psuedo-code will suffice.)

Initialize:

Repeat:

- **Step 1: _____ Step**

- **Step 2: _____ Step**

Until:

SA7. Draw a basic diagram of an *autoencoder* architecture, labeling the main components and explaining what they each do.

SA8. Explain how density-based clustering methods (*e.g.*, DBSCAN) can be used for outlier detection.

SA9. Why is cross-validation generally **not** a suitable strategy for validating unsuperised learning results and what strategies could be used instead?

Applications of Machine Learning: 40 points total

AML1. In this *issue spotter* question, I will describe a theoretical application of unsupervised machine learning. As described, this application falls short of "best practices" in several regards. Identify *four* places where this pipeline could be improved and recommend alternative (superior) practices. [20 points total] For each issue, you will receive:

- 1 point for identifying a valid issue
- (Up to) 2 points for explaining what is *wrong* with the described practice and *what impact / bias / error* would be induced
- (Up to) 2 points for accurately describing an alternative approach

Please note that there are more than four possible issues in this scenario; identify only the four for which you think you can provide the most accurate and insightful analysis.

A climate analyst has data on early summer Atlantic sea-surface temperatures (SST) measured *via* 500 ocean buoys (floating measurement devices) over a period of 75 years and wants to see if SST can be used to predict the number of hurricanes that form in the following (late summer) hurricane season. The main goal of this project is developing an easily understood set of characteristics that can be used as an early warning indicator by the general public. As such, interpretability is a major goal of the project and "black-box" methods cannot be used. Hurricane counts are highly noisy, so the analyst first wants to investigate SST data in an unsupervised fashion to see if there are distinct types of SST patterns and then to see if those patterns correlate with hurricane counts.

Because the data is measured at 500 different spatial points, the analyst first performs PCA on the raw data to reduce the dimensionality of the data down to 5 PCs. The analyst then discovers that these 5 PCs capture 90% of the variation in the data, so proceeds assuming that enough signal has been captured. The analyst next performs single-linkage hierarchical clustering on a matrix formed from the 5 $\boldsymbol{u}$-vectors (row factors) from PCA and the number of hurricanes from each year (6 columns, 75 rows). To determine the cut point of the dendrogram (and hence the number of clusters), the analyst selects the number of clusters that minimizes the within-cluster sum of squares. Finally, the analyst computes the average number of hurricanes for cluster (*i.e.*, the total number of hurricanes that occurred in a year assigned to that cluster, divided by the number of points in that cluster) and performs an ANOVA to see if the between-cluster differences in hurricane count are statistically significant.

Because the ANOVA indicates that the clusters are statistically significant and distinct from each other, the analyst believes the PCA derived features can be used as an effective set of predictors. He builds an OLS model that takes the five predictors as inputs and predicts the number of hurricanes as the output; because he has already seen that the PCs capture differences in hurricane count, he does not perform any additional model evaluation or optimization to avoid overfitting. Now that the model has been built, the analyst wants to ensure the features are as interpretable as possible. Because 5 PCs are hard to interpret, he uses t-SNE to reduce them to two features. He sees that the first t-SNE feature is correlated with average temperature across the whole Atlantic, while the second t-SNE feature seems to capture years in which there are significant differences between equatorial and arctic temperatures. Based on this information, he reports that a model can be built using these two features and that it will have statistically significant predictive power for the number of hurricanes in a given year.

Mistake #1:

Potential Adverse Impact #1:

Fix #1:

Mistake #2:

Potential Adverse Impact #2:

Fix #2:

Mistake #3:

Potential Adverse Impact #3:

Fix #3:

Mistake #4:

Potential Adverse Impact #4:

Fix #4:

AML2. For this problem, I will describe a scenario in which unsupervised learning may be useful to achieve one or more practical aims. You should describe, in reasonable detail, how you would approach this problem. Be sure to justify each step (say *why* you are making particular choices). [20 points total, 4 points for each subsection]

Note that this section will likely take longer to answer fully than any previous section, so budget your time wisely.

You are a data analyst at a startup that sells over 200 flavored varieties of sparkling water over the internet. The marketing team wants to offer a temporary discount to certain customers on certain items with the hope that these customers will try additional flavors and, in the long-run, lead to increased purchasing and increased revenues. Specifically, they plan to offer targeted "buy a case of Flavor X and get a case of Flavor Y for free" coupons to customers who have purchased X in the past and who are likely to also enjoy Y. You have been asked to help them identify the customers and the flavors on which this promotion should be offered.

You have two data sets available to you for this task:[1]

- A 200×30 table of the 200 flavors your company sells (rows) and the amount of each of your 30 flavoring agent that goes into each flavor (columns). Examples of "flavoring agents" include vanilla and passionfruit, which are used to create vanilla, passionfruit, and vanilla-passionfruit flavors for sale. Flavoring agents not used in a particular flavor are recorded as 0's.
- A $100,000 \times 200$ table of how much each customer (row) has bought of each flavor (column). Customers who have never bought a flavor are recorded as having purchased 0 cans of that flavor.

1. Your first task is to identify at least two groups of flavors to highlight in your advertising promotion. How would you process the first table and what techniques would you apply? Be careful to justify your choice (including stating any important assumptions that might need to be checked) and specify any important pre-processing steps.

[1] You may assume that the universe of flavors and customers did not change over the period in which this data was collected. *I.e.*, you don't have to worry about new flavors being introduced or a brand-new customer.

2. Your next task is to identify at least two groups of customers to whom to target your advertising promotion. How would you process the second table and what techniques would you apply? Be careful to justify your choice (including stating any important assumptions that might need to be checked) and specify any important pre-processing steps.

3. Now that you have identified groups of customers and flavors, how would you decide which flavors to promote to which customers? Describe your process in moderate detail. (Note that you do not need to advertise to *all* customers or for *all* flavors. Your goal is to outline a well-supported strategy, not to burn though as much budget as possible.)

4. In order to get sign-off from the higher-ups to launch your advertising campaign, you need to make sure your clusters have easy *and valid* interpretation. How would you modify your answer to Step (a) or Step (b) [your choice] to provide highly-interpretable results and what strategies could be used to validate that interpretation?

5. An important part of any advertising campaign is evaluation of results. What data would you encourage your company to collect to determine the efficacy of the promotion and how would you plan analyze it in six months?

(Blank page for scratch work - not graded)

(Blank page for scratch work - not graded)

(Blank page for scratch work - not graded)

STA9890 - Test 1 - Formula Sheet

Linear Algebra:

- A *n-vector* is an ordered set of n (real) numbers: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, with addition $\boldsymbol{x} + \boldsymbol{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$ and vector (inner / dot) product: $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^n x_i y_i$
- Vector norms: $\|\boldsymbol{x}\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$ with $\|\boldsymbol{x}\|_\infty = \max_i\{|x_i|\}$ and $\|\boldsymbol{x}\|_0 =$ Number of non-zero elements o
- An $m \times n$ matrix is a 2D array of real numbers with m rows and n columns:

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$

- A matrix-vector product takes an n-vector as input and gives an m-vector as output:

$$\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{A}_{1\cdot} \cdot \boldsymbol{x}, \boldsymbol{A}_{2\cdot}\boldsymbol{x}, \ldots, \boldsymbol{A}_{m\cdot} \cdot \boldsymbol{x}) \in \mathbb{R}^m$$

- We can multiply an $m \times n$ matrix with an $n \times p$ matrix - note that the 'inner' dimensions must match:

$$\boldsymbol{A}\boldsymbol{B} = \begin{pmatrix} \boldsymbol{A}_{1\cdot} \cdot \boldsymbol{B}_{\cdot 1} & \boldsymbol{A}_{1\cdot} \cdot \boldsymbol{B}_{\cdot 2} & \ldots & \boldsymbol{A}_{1\cdot} \cdot \boldsymbol{B}_{\cdot n} \\ \boldsymbol{A}_{2\cdot} \cdot \boldsymbol{B}_{\cdot 1} & \boldsymbol{A}_{2\cdot} \cdot \boldsymbol{B}_{\cdot 2} & \ldots & \boldsymbol{A}_{2\cdot} \cdot \boldsymbol{B}_{\cdot n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{m\cdot} \cdot \boldsymbol{B}_{\cdot 1} & \boldsymbol{A}_{m\cdot} \cdot \boldsymbol{B}_{\cdot 2} & \ldots & \boldsymbol{A}_{m\cdot} \cdot \boldsymbol{B}_{\cdot n} \end{pmatrix} \in \mathbb{R}^{m \times p}$$

  Consider n-vectors as *one-column* matrices to make all of these definitions consistent. Requiring the dimensions in multiplication to align is a good way to verify linear algebra claims. (*E.g.*, $\boldsymbol{A}\boldsymbol{A}$ does not work for non-square $\boldsymbol{A}$)
- A matrix inverse satisfies $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$. Only full-rank square matrices have inverses
- An (square) orthogonal matrix $\boldsymbol{Q}$ satisfies $\boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$. If we take the first $n' \leq n$ columns (rows) of an orthogonal matrix we have $\boldsymbol{Q}_{1:n'\cdot}\boldsymbol{Q}_{1:n'\cdot}^\top = \boldsymbol{I}_{n' \times n'}$ so it's transpose-inverse along the 'short-side'
- Any matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has a *singular value decomposition*: $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ where $r = \min\{m, n\}$, $\boldsymbol{D}$ is a non-negative diagonal $r \times r$ matrix, $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ is the first r columns of an orthogonal $m \times m$-matrix, and $\boldsymbol{V} \in \mathbb{R}^{n \times r}$ is the first r columns of an orthogonal $n \times n$ matrix
- Distributive rules: $(\boldsymbol{A}\boldsymbol{B})^\top = \boldsymbol{B}^\top\boldsymbol{A}^\top$ and $(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$ (if all defined)

Matrix Calculus:

- Quadratics: $f(\boldsymbol{x}) = \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} \implies \nabla f(\boldsymbol{x}) = 2\boldsymbol{A}\boldsymbol{x}$; $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 = 2\boldsymbol{x}$
- Chain rule: $g(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x}) \implies \nabla g(\boldsymbol{x}) = \boldsymbol{A}^\top(\nabla f)(\boldsymbol{A}\boldsymbol{x})$

Convexity:

- A function $f : \mathbb{R}^p \to \mathbb{R}$ is *convex* if

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) \text{ for all } \lambda \in [0,1], \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$$

  If f is convex and second-differentiable at a point, its second derivative matrix is *positive semi-definite*

- A set $\mathcal{C} \in \mathbb{R}^p$ is convex if

$$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C} \implies \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in \mathcal{C} \text{ for all } \lambda \in [0,1], \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$$

- If $\nabla f(\boldsymbol{x}_*) = 0$ for convex $f(\cdot)$, then $\boldsymbol{x}_*$ is a global minimizer of $f(\cdot)$

Gradient Methods:

- Given an optimization problem $\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$, gradient descent works by repeating the following update:

$$\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} - c \nabla f(\boldsymbol{x}^{(k)})$$

  If $c > 0$ is sufficiently small and $\mathcal{C} = \mathbb{R}^p$, $\boldsymbol{x}^{(k)}$ will converge to a minimizer of f

STA9890 - Test 2 - Formula Sheet

Ordinary Least Squares:

- $\hat{\beta} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$

Penalties:

- Best Subsets: $\|\boldsymbol{\beta}\|_0$. Number of non-zero elements (non-convex), induces sparsity in $\boldsymbol{\beta}$
- Lasso: $\|\boldsymbol{\beta}\|_1$. Tightest convex relaxation of best subsets
- Ridge: $\frac{1}{2}\|\boldsymbol{\beta}\|_2^2$. Very nice to work with (differentiable). $\frac{\partial}{\partial \boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2^2 = 2\boldsymbol{\beta}$.
- Elastic Net: α-weighted combination of ridge and lasso $\alpha\|\boldsymbol{\beta}\|_1 + \frac{(1-\alpha)}{2}\|\boldsymbol{\beta}\|_2^2$

Classification:

- (True/False) (Positive/Negative) = (Correct/Incorrect) Prediction
- Generative: $p(X|Y) \implies p(Y|X)$ via prior and Bayes' Rule. Discriminative: model $p(Y|X)$ directly.
- Bayes' Rule:

$$p(A|B) = \frac{p(B|A) * p(A)}{P(B)} = \frac{p(B|A) * p(A)}{P(B|A) * P(A) + P(B|A^c) + P(A^c)}$$

Non-Linearity:

- Feature expansion and engineering: fit linear models to non-linear parts
- Splines: piecewise polynomial models with additional smoothness constraints
- Kernel methods: feature expansion made 'easy'. Replace inner product with a 'kernel function'.

Ensembles:

- Stacking: Linear combination of base learners. Typically non-negative and sum-to-one constrained
- Bagging (Bootstrap Aggregation): building an ensemble by averaging bootstrapped based learners
- Boosting: building an ensemble by adding new ensemble members to correct past mistakes. Fit slowly for 'gradient descent' on functions

Distributions:

- Bernoulli Distribution: $X \sim \text{Bernoulli}(p) \in \{0,1\} \implies \mathbb{P}(X = x) = p^x(1-p)^{1-x}$
- Binomial Distribution: $X \sim \text{Binomial}(n,p) \in \{0,\dots,n\} \implies \mathbb{P}(X = x) = \binom{n}{x} p^x(1-p)^{n-x}$
- Poisson Distribution: $X \sim \text{Poisson}(\lambda) \in \{0,1,\dots\} \implies \mathbb{P}(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$
- Standard normal distribution. $Z \sim \mathcal{N}(0,1)$. Mean Zero + Variance 1
- Standard normal PDF - $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$. Standard normal CDF $\Phi(z) = \int_{-\infty}^{z} \phi(x)\,\mathrm{d}x$ - no closed form.
- General normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ - generated by scale+shift of standard normal $X \overset{d}{=} \mu + \sigma Z$.
- Normal PDF via standardization (z-score): $f_X(x) = \phi(\frac{x-\mu}{\sigma}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$. CDF: $\Phi(\frac{x-\mu}{\sigma})$.
- Multivariate normal parameterized by mean vector and (co)variance matrix: $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Standard multi-normal: $\boldsymbol{Z} \sim \mathcal{N}_n(\mathbf{0}_n, \boldsymbol{I}_n)$. PDF $f_{\boldsymbol{Z}}(\boldsymbol{z}) = (2\pi)^{-n/2} e^{-\|\boldsymbol{z}\|^2/2}$.
- General multi-normal $\boldsymbol{X} \overset{d}{=} \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}$ where $\boldsymbol{\Sigma}^{1/2}$ is a matrix square root (Cholesky or symmetric).
- Bivariate normal PDF

$$f_{(X,Y)}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2\left[1-\rho^2\right]}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right)$$

- Multivariate normal: any linear combination (weighted sum) of X_i is normal.
- If $\mathbb{C}[X_i, X_j] = 0$, then $X_i \perp\!\!\!\perp X_j$ (for multi-normal, uncorrelated implies independent)
- If $\boldsymbol{Z}$ is a standard normal n-vector, $\|\boldsymbol{Z}\|^2 = \sum_{i=1}^{n} Z_i^2$ has a χ^2 distribution with n degrees of freedom

STA9890 - Test 3 - Formula Sheet

Clustering Scores:

- Silhouette Score (point):

$$s_i = \frac{b_i - a_i}{\max\{b_i, a_i\}}$$

  where a_i is average distance to points in the same cluster as i and b_i is average distance to points in the nearest cluster to i

- Silhouette Score (Entire Data Set):

$$S = \max_k \frac{1}{|C_k|} \sum_{k' \in C_k} s_{k'}$$

  That is, the maximum average silhouette score across all clusters.

- Within-Cluster Sum of Squares:

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{i:\boldsymbol{x}_i \in C_k} \|\boldsymbol{x}_i - \mu_k\|^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \mu_{C_i}\|^2$$

Clustering Linkages:

- Single Linkage: $d(A,B) = \min_{a\in A, b\in B} d(a,b)$
- Complete Linkage: $d(A,B) = \max_{a\in A, b\in B} d(a,b)$
- Average Linkage: $d(A,B) = \frac{1}{|A||B|} \sum_{a\in A, b\in B} d(a,b)$
- Centroid Linkage: $d(A,B) = d(\overline{a}, \overline{b}) = d\left(|A|^{-1} \sum_{a\in A} a, |B|^{-1} \sum_{b\in B} b\right)$

where the pointwise distances $d(a,b)$ are typically based on ℓ_p norms: $d(a,b) = \|a-b\|_p$

Convex Clustering:

$$\arg\min_{\boldsymbol{u}_i \in \mathbb{R}^p (i\in 1,\ldots,n)} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{u}_i\| + \lambda \sum_{i,j} \|\boldsymbol{u}_i - \boldsymbol{u}_j\|$$

PCA:

- Variance explained by PC-k: d_k^2
- Percent variance explained (marginally) by PC-k: $d_k^2 / \sum_i d_i^2$
- Cumulative percent variance explained up to (and including) PC-k: $(\sum_{i=1}^{k} d_i^2)/(\sum_{j=1}^{\text{rank}(\boldsymbol{X})} d_j^2)$
- Built upon SVD (see above):

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{j=1}^{\text{rank}(\boldsymbol{X})} d_j \, \boldsymbol{u}_j \boldsymbol{v}_j^\top$$