### **File 1: LinearCombinationSpan.pdf**

#### **1. Linear Combinations**

* A vector **v** is a linear combination of **v1, v2, ..., vn** if:
  + **v = c1*v1 + c2*v2 + ... + cn\*vn***(Equation in Green)*: **v = c1\*v1 + c2\*v2 + ... + cn\*vn**
* Scalars **c1, c2, ..., cn** determine how much each vector contributes to the combination.

#### **2. Span of Vectors**

* The span of a set of vectors **{v1, v2, ..., vn}** is the collection of all linear combinations of these vectors:
  + **Span({v1, v2, ..., vn}) = {c1*v1 + c2*v2 + ... + cn\*vn | ci ∈ R}**
  + *(Equation in Green)*:

**Span({v1, v2, ..., vn}) = {c1\*v1 + c2\*v2 + ... + cn\*vn | ci ∈ R}**

#### **3. Linear Dependence and Independence**

* Vectors **{v1, v2}** are linearly dependent if one vector is a scalar multiple of the other.
* A set of vectors is linearly independent if no vector in the set can be expressed as a linear combination of the others.

#### **4. Geometric Meaning**

* The span of two independent vectors in **R²** forms a plane through the origin.
* In **R³**, the span of three independent vectors fills the entire space.

#### **5. Relation to Subspaces**

* The column space of a matrix is the span of its column vectors.
* Subspaces, such as null space or row space, are defined in terms of spans.

### 

### 

### 

### 

### **File 2: LeastSquare.pdf**

#### **1. Least Squares Problem**

* The least-squares method minimizes the squared difference between observed data (**b**) and the predicted data (**Ax**):
  + **||Ax - b||²***(Equation in Green)*: **||Ax - b||²**
* It is commonly used when the system **Ax = b** has no exact solution (overdetermined systems).

#### **2. Normal Equations**

* The solution to the least-squares problem satisfies the normal equations:
  + **AᵀA\*x = Aᵀb***(Equation in Green)*: **AᵀA\*x = Aᵀb**
* These equations ensure that the residual **r = b - Ax** is orthogonal to the column space of **A**.

#### **3. Projection Interpretation**

* The solution **x** projects **b** onto the column space of **A**:
  + **b\_proj = Ax***(Equation in Green)*: **b\_proj = Ax**

#### **4. Key Properties of the Residual**

* The residual **r** satisfies:
  + **Aᵀr = 0***(Equation in Green)*: **Aᵀr = 0**
* This implies that **r** is orthogonal to all columns of **A**.

#### **5. Application Contexts**

* Least squares are used in **linear regression** to fit a line or hyperplane to data points by minimizing prediction error.

### **File 3: EigenvectorGeo.pdf**

#### **1. Definition of Eigenvectors and Eigenvalues**

* An eigenvector **v** of a square matrix **A** satisfies:
  + **A*v = λ*v***(Equation in Green)*: **A\*v = λ\*v**
* Here, **λ** is the eigenvalue associated with the eigenvector **v**.

#### **2. Geometric Interpretation of Eigenvectors**

* Eigenvectors represent directions that remain unchanged under the transformation defined by **A**.
* The eigenvalue **λ** scales the eigenvector along its direction.

#### **3. Eigenvectors in 2D and 3D**

* In **R²**, eigenvectors indicate directions in which stretching or compressing occurs under the matrix transformation.
* In **R³**, eigenvectors represent invariant lines for scaling transformations.

#### **4. Characteristic Equation**

* Eigenvalues are solutions to the characteristic equation:
  + **det(A - λI) = 0***(Equation in Green)*: **det(A - λI) = 0**

#### **5. Applications**

* Eigenvectors are used to identify principal directions in data (e.g., in PCA).
* They describe geometric transformations such as rotations, scaling, and shearing.

### 

### 

### 

### 

### 

### 

### **File 4: ParametricEigenvec.pdf**

#### **1. Parameterized Representation of Eigenvectors**

* Eigenvectors corresponding to a specific eigenvalue can form a vector space (eigenspace).
* These eigenvectors can be expressed parametrically, especially when multiple solutions exist for the equation:
  + **(A - λI)v = 0***(Equation in Green)*: **(A - λI)v = 0**
* For example, if the eigenspace has two basis vectors **v1** and **v2**, any eigenvector can be expressed as:
  + **v = c1*v1 + c2*v2, c1, c2 ∈ R***(Equation in Green)*: **v = c1\*v1 + c2\*v2, c1, c2 ∈ R**

#### **2. Geometric Interpretation**

* For a given eigenvalue **λ**, the eigenvectors lie along a line (in 2D) or a plane (in 3D).
* If **λ** is repeated, the eigenspace's dimension increases, often forming a subspace.

#### **3. Eigenspace Construction**

* The eigenspace for eigenvalue **λ** is the null space of the matrix **(A - λI)**.
* It can be parameterized using free variables after solving the system of linear equations.

#### **4. Applications of Parameterized Eigenvectors**

* Used in physics for systems with symmetry, where parameterized eigenvectors describe invariant modes.
* Applied in PCA for reconstructing data along principal directions.

### 

### 

### 

### 

### 

### 

### **File 5: IntroToVectors.pdf**

#### **1. Definition of a Vector**

* A vector **v** is an ordered collection of numbers representing magnitude and direction. It is commonly written as:
  + **v = [v1 v2 ... vn]***(Equation in Green)*: **v = [v1 v2 ... vn]**
* The elements **vi** are the components of the vector.

#### **2. Vector Operations**

* **Addition:** The sum of two vectors **u** and **v** is:
  + **u + v = [u1 + v1 u2 + v2 ... un + vn]***(Equation in Green)*: **u + v = [u1 + v1 u2 + v2 ... un + vn]**
* **Scalar Multiplication:** For a scalar **c**, multiplying **c** with **v** results in:
  + **c*v = [c*v1 c*v2 ... c*vn]***(Equation in Green)*: **c\*v = [c\*v1 c\*v2 ... c\*vn]**

#### **3. Dot Product**

* The dot product of two vectors **u** and **v** is:
  + **u ⋅ v = u1*v1 + u2*v2 + ... + un\*vn***(Equation in Green)*: **u ⋅ v = u1\*v1 + u2\*v2 + ... + un\*vn**
* It provides a measure of similarity between **u** and **v**. Orthogonal vectors have a dot product of **0**.

#### **4. Magnitude of a Vector**

* The length (magnitude) of **v** is calculated using the Euclidean norm:
  + **||v|| = √(v1² + v2² + ... + vn²)***(Equation in Green)*: **||v|| = √(v1² + v2² + ... + vn²)**

#### **5. Unit Vectors**

* A unit vector is a vector with magnitude **1**. Any vector **v** can be converted into a unit vector **u** by normalizing it:
  + **u = v / ||v||***(Equation in Green)*: **u = v / ||v||**

#### **6. Geometric Interpretation**

* Vectors represent directions and magnitudes in space.
* Operations such as addition and scalar multiplication can be visualized geometrically as shifting or scaling vectors.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 6: MatrixMultiplication.pdf**

#### **1. Definition of Matrix Multiplication**

* The product of two matrices **A** (of size **m × n**) and **B** (of size **n × p**) is a matrix **C** (of size **m × p**), where:
  + **Cᵢⱼ = Σ (Aᵢₖ \* Bₖⱼ)** (sum over k from 1 to n)  
    *(Equation in Green)*: **Cᵢⱼ = Σ (Aᵢₖ \* Bₖⱼ)**
* This involves taking the dot product of the **i**-th row of **A** with the **j**-th column of **B**.

#### **2. Conditions for Multiplication**

* Matrix multiplication is only defined if the number of columns in **A** equals the number of rows in **B**.

#### **3. Properties of Matrix Multiplication**

* **Associativity:** *A(B*C) = (A*B)C  
  (Equation in Green)*: **A\*(B\*C) = (A\*B)\*C**
* **Distributivity:** *A(B + C) = A*B + A\*C\*\*  
  *(Equation in Green)*: **A\*(B + C) = A\*B + A\*C**
* **Non-commutativity:** **A*B ≠ B*A** in general.

#### **4. Transpose Rule for Products**

* The transpose of a product of matrices satisfies:
  + **(A\*B)ᵀ = Bᵀ \* Aᵀ***(Equation in Green)*: **(A\*B)ᵀ = Bᵀ \* Aᵀ**

#### **5. Geometric Interpretation**

* Matrix multiplication can be viewed as applying a linear transformation (encoded by **B**) to each column of **A**.

#### **6. Applications of Matrix Multiplication**

* Used in transforming coordinates in 2D/3D graphics.
* Forms the foundation for solving systems of linear equations, machine learning algorithms, and quantum mechanics.

### 

### 

### **File 7: DotProd.pdf & DotProd2.pdf**

#### **1. Definition of the Dot Product**

* The dot product of two vectors **u** and **v** is defined as:
  + **u ⋅ v = Σ (uᵢ \* vᵢ)** (sum over i from 1 to n)  
    *(Equation in Green)*: **u ⋅ v = Σ (uᵢ \* vᵢ)**
* Alternatively, in terms of magnitude and angle:
  + **u ⋅ v = ||u|| \* ||v|| \* cos(θ)***(Equation in Green)*: **u ⋅ v = ||u|| \* ||v|| \* cos(θ)**
* where **θ** is the angle between **u** and **v**.

#### **2. Geometric Interpretation**

* The dot product quantifies the projection of one vector onto another.
* If **u ⋅ v = 0**, the vectors **u** and **v** are orthogonal.

#### **3. Properties of the Dot Product**

* **Commutativity:** **u ⋅ v = v ⋅ u***(Equation in Green)*: **u ⋅ v = v ⋅ u**
* **Distributivity:** **u ⋅ (v + w) = u ⋅ v + u ⋅ w***(Equation in Green)*: **u ⋅ (v + w) = u ⋅ v + u ⋅ w**
* **Scaling:** **(c \* u) ⋅ v = c \* (u ⋅ v)***(Equation in Green)*: **(c \* u) ⋅ v = c \* (u ⋅ v)**

#### **4. Angle Between Vectors**

* The cosine of the angle **θ** between two vectors can be found using:
  + **cos(θ) = (u ⋅ v) / (||u|| \* ||v||)***(Equation in Green)*: **cos(θ) = (u ⋅ v) / (||u|| \* ||v||)**

#### **5. Applications**

* Computing similarity between vectors (e.g., in information retrieval or machine learning).
* Detecting perpendicularity or alignment between vectors in physics and engineering problems.

### 

### 

### 

### **File 8: L2NormUnitVector.pdf**

#### **1. Definition of the L₂ Norm**

* The **L₂** (Euclidean) norm of a vector **v** is defined as:
  + **||v|| = √(Σ vᵢ²)** (sum over i from 1 to n)  
    *(Equation in Green)*: **||v|| = √(Σ vᵢ²)**
* It measures the length (magnitude) of the vector in Euclidean space.

#### **2. Unit Vector**

* A unit vector is a vector with a magnitude of **1**:
  + **||u|| = 1***(Equation in Green)*: **||u|| = 1**
* To convert a vector **v** into a unit vector **u**, normalize **v** by dividing it by its **L₂** norm:
  + **u = v / ||v||***(Equation in Green)*: **u = v / ||v||**

#### **3. Geometric Interpretation**

* The unit vector retains the direction of the original vector but scales it to have a magnitude of **1**.

#### **4. Applications of the L₂ Norm**

* In machine learning, **L₂** norm is used for regularization (Ridge Regression) to penalize large weights.
* Unit vectors are widely used in defining coordinate axes, representing directions, and calculating projections.

### 

### 

### 

### 

### 

### 

### **File 9: MatrixEquation.pdf**

#### **1. Matrix Equation Representation**

* A system of linear equations can be written in matrix form as:
  + **A\*x = b***(Equation in Green)*: **A\*x = b**
* where:
  + **A**: Coefficient matrix
  + **x**: Vector of unknowns
  + **b**: Right-hand side vector

#### **2. Solution Types**

* **Unique Solution:** Occurs when **A** is square and invertible (**det(A) ≠ 0**).
* **No Solution:** Occurs when **b** lies outside the column space of **A**.
* **Infinite Solutions:** Occurs when **A** is not full rank (linearly dependent columns).

#### **3. Solving the Equation**

* If **A** is invertible, the solution is:
  + **x = A⁻¹ \* b***(Equation in Green)*: **x = A⁻¹ \* b**
* If **A** is not invertible, least-squares methods or row reduction are used.

#### **4. Geometric Interpretation**

* **A\*x** represents a linear transformation applied to **x**.
* The solution **x** corresponds to the point where the transformation **A** maps **x** to **b**.

#### **5. Applications**

* Matrix equations are used to model systems of equations in physics, engineering, and computer graphics.
* They form the foundation for linear regression and other optimization techniques.

### **File 10: HowToDiag.pdf**

#### **1. Matrix Diagonalization**

* A matrix **A** is diagonalizable if it can be written as:
  + **A = P*D*P⁻¹***(Equation in Green)*: **A = P\*D\*P⁻¹**
* where:
  + **P** is the matrix of eigenvectors.
  + **D** is a diagonal matrix with eigenvalues of **A** on the diagonal.

#### **2. Steps to Diagonalize a Matrix**

* **Step 1:** Compute the eigenvalues **λ** by solving:
  + **det(A - λI) = 0***(Equation in Green)*: **det(A - λI) = 0**
* **Step 2:** Find the eigenvectors corresponding to each eigenvalue by solving:
  + **(A - λI)v = 0***(Equation in Green)*: **(A - λI)v = 0**
* **Step 3:** Form **P** using the eigenvectors as columns.
* **Step 4:** Form **D** by placing the eigenvalues along its diagonal.

#### **3. Conditions for Diagonalizability**

* **A** is diagonalizable if and only if it has **n** linearly independent eigenvectors (for an **n × n** matrix).
* If eigenvalues are distinct, **A** is guaranteed to be diagonalizable.

#### **4. Geometric Interpretation**

* Diagonalization simplifies linear transformations by aligning them with the eigenvectors' directions.
* Each eigenvector corresponds to an axis of scaling defined by its eigenvalue.

#### **5. Applications of Diagonalization**

* Simplifies matrix exponentiation and computation of powers **Aᵏ** by:
  + **Aᵏ = P*Dᵏ*P⁻¹***(Equation in Green)*: **Aᵏ = P\*Dᵏ\*P⁻¹**
* Used in differential equations, Markov chains, and Principal Component Analysis (PCA).

### 

### **File 11: WhyPCA.pdf**

#### **1. Purpose of PCA (Principal Component Analysis)**

* PCA reduces the dimensionality of data while preserving as much variance as possible.
* It identifies new axes (principal components) aligned with directions of maximum variance in the data.

#### **2. Reasons for Using PCA**

* **Eliminating Redundancy:** Reduces highly correlated features into fewer independent components.
* **Data Compression:** Maintains key patterns in the data while reducing storage and computational requirements.
* **Noise Reduction:** Filters out noise by retaining only the components with significant variance.

#### **3. Steps in PCA**

* **Step 1:** Standardize the data (zero mean and unit variance).
* **Step 2:** Compute the covariance matrix to measure the relationships between features.
* **Step 3:** Calculate the eigenvalues and eigenvectors of the covariance matrix.
* **Step 4:** Project the data onto the principal components (eigenvectors corresponding to the largest eigenvalues).

#### **4. Interpretation of Principal Components**

* The first principal component captures the maximum variance in the data.
* Subsequent principal components capture the remaining variance and are orthogonal to each other.

#### **5. Applications of PCA**

* **Image Compression:** Reduces image data dimensions while maintaining most of the visual content.
* **Feature Selection:** Helps select key features for machine learning models.
* **Data Visualization:** Projects high-dimensional data into 2D or 3D for visualization.

### 

### 

### 

### **File 12: WhyGreatestVariability.pdf**

#### **1. PCA Focuses on Greatest Variability**

* Principal Component Analysis (PCA) identifies the directions (principal components) where the data varies the most.
* These directions correspond to the eigenvectors of the covariance matrix, ordered by the magnitude of their eigenvalues.

#### **2. Why Variability Matters**

* Variability reflects the spread or dispersion of the data. Focusing on the greatest variability ensures that key patterns in the data are preserved.
* Lower variability directions often correspond to noise or redundant features.

#### **3. Role of Covariance Matrix in PCA**

* The covariance matrix summarizes the relationships (correlations) between features:
  + **Cov(X) = (1/(n - 1)) \* Xᵀ\*X***(Equation in Green)*: **Cov(X) = (1/(n - 1)) \* Xᵀ\*X**
* Eigenvalues of the covariance matrix indicate the amount of variance explained by each principal component.

#### **4. Reduction Without Significant Loss**

* Retaining only the top **k** principal components (with the largest eigenvalues) captures the majority of the total variance while discarding less informative directions.

#### **5. Applications of Variability Focus**

* **Data Compression:** Reduces dimensionality by keeping only the most significant components.
* **Noise Filtering:** Low-variance components are often attributed to noise and can be discarded without substantial loss of information.

#### **6. Example of Variability Analysis**

* In a dataset with correlated features, PCA rotates the axes to align with directions of maximum spread, decorrelating the data.

### 

### 

### **File 13: HowToPCA.pdf**

#### **1. Step-by-Step Process for PCA**

* **Step 1:** Standardize the Data  
  Ensure that each feature has zero mean and unit variance to avoid biasing principal components toward features with larger magnitudes.
* **Step 2:** Compute the Covariance Matrix  
  Calculate the covariance matrix to capture relationships between features:
  + **Cov(X) = (1/(n - 1)) \* Xᵀ\*X***(Equation in Green)*: **Cov(X) = (1/(n - 1)) \* Xᵀ\*X**
* **Step 3:** Eigenvalue Decomposition  
  Perform eigenvalue decomposition of the covariance matrix:
  + **Cov(X)v = λv***(Equation in Green)*: **Cov(X)v = λv**
  + Eigenvalues (**λ**) represent the variance explained by the principal components.
  + Eigenvectors (**v**) represent the principal components.
* **Step 4:** Select Top Components  
  Rank the eigenvalues and select the top **k** components with the largest eigenvalues.
* **Step 5:** Project Data  
  Transform the data into the new basis (principal components):
  + **X\_projected = X \* W***(Equation in Green)*: **X\_projected = X \* W**
* where **W** is a matrix of the top **k** eigenvectors.

#### **2. Interpretation of PCA Output**

* The principal components are linear combinations of the original features that capture the greatest variance.
* The variance explained by each component is proportional to its eigenvalue.

#### **3. Practical Considerations**

* **Standardization is Key:** Features must be standardized if they are on different scales.
* **Cumulative Variance Explained:** Decide the number of components to retain based on the cumulative percentage of variance explained.

#### **4. Applications**

* **Dimensionality Reduction:** Retain only components contributing to a specified percentage of variance.
* **Data Visualization:** Project high-dimensional data onto 2D or 3D for easier interpretation.

### **File 14: HowToScorePCA.pdf**

#### **1. Scoring in PCA**

* PCA Scores refer to the transformed data obtained after projecting the original dataset onto the principal components.
* For a data matrix **X**, the scores are computed as:
  + **X\_scores = X \* W***(Equation in Green)*: **X\_scores = X \* W**
* where **W** contains the eigenvectors (principal components).

#### **2. Significance of Scores**

* Each row in the PCA scores matrix represents the projection of a data point onto the principal component axes.
* Scores indicate how much a data point contributes to each principal component.

#### **3. Interpreting Scores**

* Higher absolute values in the scores indicate stronger alignment of a data point with the corresponding principal component.
* The first principal component typically explains the largest portion of variance, so its scores often dominate.

#### **4. Scaling and Reconstruction**

* PCA scores can be scaled back to the original feature space to approximate the original data:
  + **X\_approx = X\_scores \* Wᵀ***(Equation in Green)*: **X\_approx = X\_scores \* Wᵀ**
* This approximation is useful for compression and noise reduction.

#### **5. Applications of PCA Scoring**

* **Clustering:** Group data points based on PCA scores to identify patterns.
* **Outlier Detection:** Data points with extreme PCA scores can be flagged as outliers.
* **Feature Analysis:** Scores help evaluate the relative importance of each component for individual data points.

#### **6. Practical Considerations**

* The choice of **k**, the number of components retained, significantly impacts the interpretability and accuracy of the scores.
* Cumulative variance explained by retained components helps guide the selection of **k**.

### **File 15: LandsatPCA (2).pdf**

#### **1. PCA for Landsat Imagery**

* PCA is applied to multispectral Landsat imagery to reduce the number of bands while preserving key information.
* Original Landsat images consist of multiple correlated spectral bands that can be compressed into fewer principal components.

#### **2. Steps for PCA on Landsat Data**

* **Step 1:** Standardization  
  Standardize each band to have zero mean and unit variance, ensuring that bands with higher magnitude values do not dominate the principal components.
* **Step 2:** Compute Covariance Matrix  
  The covariance matrix captures the relationships between pixel intensities across different bands.
* **Step 3:** Eigenvalue Decomposition  
  Decompose the covariance matrix into eigenvalues and eigenvectors to identify principal components.
* **Step 4:** Projection  
  Project the original data onto the eigenvectors corresponding to the largest eigenvalues.

#### **3. Benefits of PCA for Landsat Data**

* **Data Compression:** Reduces the number of bands without significant loss of information.
* **Noise Reduction:** Low-variance bands (often associated with noise) are discarded.
* **Improved Visualization:** PCA combines the most informative features into a smaller number of components, which are easier to visualize and interpret.

#### **4. Interpretation of PCA Output for Landsat**

* The first few principal components typically capture surface reflectance information that is crucial for vegetation, water bodies, and urban features.
* Higher-order components often correspond to noise or less relevant spectral information.

#### **5. Applications in Remote Sensing**

* **Land Cover Classification:** PCA simplifies data, making it easier to classify land types.
* **Change Detection:** Principal components highlight regions with significant spectral variations, useful for monitoring environmental changes.
* **Feature Extraction:** PCA-derived components are used as input features for further analysis in machine learning models.

#### **6. Practical Example in Landsat Analysis**

* Applying PCA to a 7-band Landsat image may yield 2-3 principal components that capture most of the variance, significantly reducing the dimensionality.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 16: SVDRecommenderSystems.pdf**

#### **1. Role of SVD in Recommender Systems**

* Singular Value Decomposition (SVD) is used to decompose user-item interaction matrices into latent factors, enabling personalized recommendations.
* A matrix **A** (user-item ratings) is decomposed as:
  + **A = U \* Σ \* Vᵀ***(Equation in Green)*: **A = U \* Σ \* Vᵀ**
* where:
  + **U**: User latent factors matrix.
  + **Σ**: Diagonal matrix of singular values.
  + **V**: Item latent factors matrix.

#### **2. Steps in SVD for Recommendations**

* **Step 1:** Construct the user-item matrix with rows representing users and columns representing items.
* **Step 2:** Apply SVD to factorize the matrix into **U**, **Σ**, and **V**.
* **Step 3:** Use truncated SVD by retaining only the largest **k** singular values and corresponding singular vectors:
  + **Aₖ = Uₖ \* Σₖ \* Vₖᵀ***(Equation in Green)*: **Aₖ = Uₖ \* Σₖ \* Vₖᵀ**
* **Step 4:** Predict missing entries in the user-item matrix by reconstructing it from **Aₖ**.

#### **3. Benefits of Using SVD**

* **Dimensionality Reduction:** SVD reduces the size of the user-item matrix, capturing only the most relevant latent factors.
* **Noise Filtering:** Smaller singular values often represent noise, which can be discarded to improve predictions.
* **Latent Space Modeling:** Projects users and items into a shared latent space where similarities can be measured.

#### **4. Recommender System Predictions**

* Predict a user's rating for an item by computing the dot product of the user’s latent vector and the item’s latent vector:
  + **r̂ᵤᵢ = uᵤᵀ \* vᵢ***(Equation in Green)*: **r̂ᵤᵢ = uᵤᵀ \* vᵢ**
* where **uᵤ** is the user latent vector and **vᵢ** is the item latent vector.

#### **5. Applications of SVD in Recommender Systems**

* **Collaborative Filtering:** Recommends items based on shared preferences of similar users or items.
* **Cold Start Problem Mitigation:** SVD can handle sparse matrices by inferring latent factors from available data.
* **Improved Scalability:** Truncated SVD reduces computational costs, making it feasible for large datasets.

#### **6. Limitations**

* SVD assumes that the user-item matrix is dense, which can pose challenges for highly sparse matrices.
* Requires retraining the model when new users or items are introduced.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 17: PCA\_SVD.pdf**

#### **1. Relationship Between PCA and SVD**

* PCA and SVD are mathematically related and often used interchangeably for dimensionality reduction:
  + In PCA, the principal components are the eigenvectors of the covariance matrix.
  + In SVD, the right singular vectors (**V**) of the data matrix correspond to the principal components.

#### **2. SVD and Data Matrix Decomposition**

* For a data matrix **X**, SVD decomposes it as:
  + **X = U \* Σ \* Vᵀ***(Equation in Green)*: **X = U \* Σ \* Vᵀ**
  + **U**: Left singular vectors, representing the row space.
  + **Σ**: Singular values, representing the magnitude of variance.
  + **V**: Right singular vectors, aligned with the principal components.

#### **3. PCA Using SVD**

* Instead of directly computing the covariance matrix, PCA can be performed using SVD:
  + The eigenvalues of the covariance matrix are the squares of the singular values from **Σ**.
  + The principal components are given by the columns of **V**.

#### **4. Dimensionality Reduction via SVD**

* By truncating **Σ** and retaining only the top **k** singular values, the data can be approximated as:
  + **Xₖ = Uₖ \* Σₖ \* Vₖᵀ***(Equation in Green)*: **Xₖ = Uₖ \* Σₖ \* Vₖᵀ**
* This retains the most significant directions of variance while reducing noise.

#### **5. Key Advantages of Using SVD for PCA**

* **Numerical Stability:** SVD avoids issues with large covariance matrices by directly operating on the data matrix.
* **Efficient for Sparse Data:** Works well with datasets where the covariance matrix is computationally expensive to calculate.

#### **6. Applications of PCA and SVD**

* **PCA:** Used for feature extraction, noise reduction, and data visualization.
* **SVD:** Applied in collaborative filtering, image compression, and text analysis (e.g., Latent Semantic Analysis).

#### **7. Interpretation of Singular Values**

* Singular values represent the amount of variance explained by each component. Larger singular values correspond to more significant components.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 18: CharacteristicEq.pdf**

#### **1. Characteristic Equation**

* The characteristic equation of a square matrix **A** is used to find its eigenvalues. It is given by:
  + **det(A - λI) = 0***(Equation in Green)*: **det(A - λI) = 0**
* where:
  + **A**: The matrix.
  + **λ**: Eigenvalue.
  + **I**: Identity matrix of the same size as **A**.

#### **2. Steps to Solve the Characteristic Equation**

* **Step 1:** Construct **A - λI**, which involves subtracting **λ** from the diagonal elements of **A**.
* **Step 2:** Calculate the determinant of **A - λI**.
* **Step 3:** Solve the resulting polynomial equation (characteristic polynomial) for **λ**.

#### **3. Properties of Eigenvalues**

* For an **n × n** matrix, the characteristic polynomial is a degree-**n** polynomial, yielding up to **n** eigenvalues (including multiplicities).
* Eigenvalues can be real or complex, depending on the matrix.

#### **4. Eigenvalues of Special Matrices**

* **Diagonal Matrices:** Eigenvalues are the diagonal entries.
* **Triangular Matrices:** Eigenvalues are also the diagonal entries.
* **Symmetric Matrices:** All eigenvalues are real.

#### **5. Geometric Interpretation of Eigenvalues**

* Eigenvalues indicate the factor by which an eigenvector is stretched or compressed under the linear transformation represented by **A**.

#### **6. Applications of the Characteristic Equation**

* Used to determine stability in dynamic systems (e.g., in control theory).
* Essential in Principal Component Analysis (PCA) for determining the variance captured by each principal component.
* Forms the basis for diagonalization, which simplifies many matrix operations.

### 

### **File 19: WhyDiag.pdf**

#### **1. Purpose of Diagonalization**

* Diagonalization simplifies the representation of a square matrix **A** by expressing it in terms of its eigenvalues and eigenvectors:
  + **A = P \* D \* P⁻¹***(Equation in Green)*: **A = P \* D \* P⁻¹**
* where:
  + **P**: Matrix of eigenvectors of **A**.
  + **D**: Diagonal matrix of eigenvalues of **A**.

#### **2. Benefits of Diagonalization**

* **Matrix Powers:** Simplifies computation of powers of **A**:
  + **Aᵏ = P \* Dᵏ \* P⁻¹***(Equation in Green)*: **Aᵏ = P \* Dᵏ \* P⁻¹**
* where **Dᵏ** is easy to compute since it is diagonal.
* **Simplifies Linear Transformations:** Enables better understanding of how a matrix scales and rotates vectors.
* **Efficient Calculations:** Simplifies matrix exponentiation, logarithms, and inversion.

#### **3. Conditions for Diagonalization**

* A matrix **A** is diagonalizable if:
  + **A** has **n** linearly independent eigenvectors (for an **n × n** matrix).
  + Matrices with distinct eigenvalues are always diagonalizable.

#### **4. Applications of Diagonalization**

* **Differential Equations:** Used to solve systems of linear differential equations.
* **Quantum Mechanics:** Diagonalization simplifies Hamiltonians to find eigenstates and eigenenergies.
* **Markov Chains:** Transition matrices in Markov processes are diagonalized to compute steady-state probabilities.

#### **5. Geometric Interpretation**

* The columns of **P** align with the eigenvectors, and the diagonal entries of **D** scale along those directions.
* Diagonalization transforms a matrix into a form that reveals its scaling and rotation properties explicitly.

### 

### **File 20: NullAltHyp.pdf**

#### **1. Definition of Null and Alternative Hypotheses**

* **Null Hypothesis (H₀):** Represents the default assumption that there is no effect, no difference, or no relationship in the data.
  + Example: **H₀: μ = 50***(Equation in Green)*: **H₀: μ = 50***(The population mean is 50.)*
* **Alternative Hypothesis (Hₐ):** Represents the opposing claim that there is an effect, a difference, or a relationship.
  + Example: **Hₐ: μ > 50***(Equation in Green)*: **Hₐ: μ > 50***(The population mean is greater than 50.)*

#### **2. Types of Alternative Hypotheses**

* **One-Sided (Directional):** Specifies a direction of the effect (e.g., **Hₐ: μ > 50** or **Hₐ: μ < 50**).
* **Two-Sided (Non-Directional):** Tests for any difference (e.g., **Hₐ: μ ≠ 50**).
  + Example: **Hₐ: μ ≠ 50***(Equation in Green)*: **Hₐ: μ ≠ 50**

#### **3. Hypothesis Testing Procedure**

* Formulate **H₀** and **Hₐ**.
* Choose a significance level (**α**), often set at 0.05.
* Compute the test statistic based on sample data.
* Compare the test statistic to the critical value or compute the **p-value** to make a decision:
  + Reject **H₀** if the **p-value < α**.

#### **4. Key Errors in Hypothesis Testing**

* **Type I Error:** Rejecting **H₀** when it is true (false positive).
* **Type II Error:** Failing to reject **H₀** when **Hₐ** is true (false negative).

#### **5. Applications of Hypothesis Testing**

* Comparing group means in experiments (e.g., treatment vs. control).
* Testing the efficacy of new drugs or treatments.
* Validating relationships in regression models.

#### **6. Practical Example**

* If a study tests whether a new drug increases recovery rates, the hypotheses might be:
  + **H₀:** The drug has no effect on recovery rates.
  + **Hₐ:** The drug increases recovery rates.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 21: OneSampleTtest.pdf**

#### **1. Purpose of the One-Sample t-Test**

* The one-sample t-test determines whether the mean of a single sample differs significantly from a known or hypothesized population mean (**μ₀**).

#### **2. Hypotheses in a One-Sample t-Test**

* **Null Hypothesis (H₀):** The sample mean equals the population mean:
  + **H₀: μ = μ₀***(Equation in Green)*: **H₀: μ = μ₀**
* **Alternative Hypothesis (Hₐ):** The sample mean differs from the population mean. Examples include:
  + **Two-sided:** **Hₐ: μ ≠ μ₀***(Equation in Green)*: **Hₐ: μ ≠ μ₀**
  + **One-sided:** **Hₐ: μ > μ₀** or **Hₐ: μ < μ₀**

#### **3. Test Statistic Formula**

* The t-test statistic is calculated as:
  + **t = (x̄ - μ₀) / (s / √n)***(Equation in Green)*: **t = (x̄ - μ₀) / (s / √n)**
* where:
  + **x̄**: Sample mean.
  + **μ₀**: Hypothesized population mean.
  + **s**: Sample standard deviation.
  + **n**: Sample size.

#### **4. Degrees of Freedom**

* The degrees of freedom for the t-test is **n - 1**, where **n** is the sample size.

#### **5. Decision Rule**

* Compare the calculated **t-statistic** with the critical value from the t-distribution table at a given significance level (**α**).
* Alternatively, compute the **p-value**:
  + Reject **H₀** if **p-value < α**.

#### **6. Applications**

* Evaluating whether the average test score in a class differs from a national average.
* Testing whether the mean weight of a product meets a specified target.

#### **7. Example**

* If the sample mean **x̄ = 102**, **μ₀ = 100**, **s = 5**, and **n = 25**:
  + **t = (102 - 100) / (5 / √25) = 2***(Equation in Green)*: **t = (102 - 100) / (5 / √25) = 2**
* The test statistic would then be compared to the critical value or used to calculate the **p-value**.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 22: ThreeStatisticalTests.pdf**

#### **1. Overview of the Three Statistical Tests**

* This document outlines three common statistical tests: t-tests, Chi-square tests, and ANOVA. These tests are designed for different types of data and hypotheses.

#### **2. T-Test**

##### **Purpose of a T-Test**

* Used to compare the means of two groups to determine if they are significantly different.

##### **Types of T-Tests**

* **One-Sample T-Test:** Tests whether the mean of a single group differs from a known value or population mean.
* **Independent T-Test:** Compares the means of two independent groups (e.g., treatment vs. control).
* **Paired T-Test:** Compares the means of two related groups (e.g., pre-treatment vs. post-treatment).

##### **Key Requirements**

* Assumes the data are normally distributed.
* For an independent t-test, assumes equal variance between groups.

##### **T-Test Statistic Formula**

* General formula:
  + **t = (difference in means) / (standard error of the difference)***(Equation in Green)*: **t = (difference in means) / (standard error of the difference)**

#### **3. Chi-Square Test**

##### **Purpose of a Chi-Square Test**

* Tests for independence between categorical variables in a contingency table.

##### **Test Statistic**

* The Chi-square test statistic is calculated as:
  + **χ² = Σ ((O - E)² / E)***(Equation in Green)*: **χ² = Σ ((O - E)² / E)**
* where:
  + **O**: Observed frequency.
  + **E**: Expected frequency.

##### **Applications**

* Testing whether gender and voting preference are independent.
* Evaluating the fit of an observed distribution to an expected distribution.

##### **Key Assumptions**

* Expected frequencies in each cell should be **≥ 5**.

#### **4. ANOVA (Analysis of Variance)**

##### **Purpose of ANOVA**

* Compares the means of three or more groups to determine if at least one group mean is significantly different.

##### **Types of ANOVA**

* **One-Way ANOVA:** Examines one factor with multiple levels (e.g., comparing test scores across three teaching methods).
* **Two-Way ANOVA:** Examines the interaction between two factors (e.g., test scores by teaching method and student gender).

##### **ANOVA Test Statistic**

* ANOVA uses the **F-statistic**:
  + **F = (Between-group variance) / (Within-group variance)***(Equation in Green)*: **F = (Between-group variance) / (Within-group variance)**

##### **Post-Hoc Testing**

* If ANOVA reveals significant differences, post-hoc tests (e.g., Tukey’s HSD) are used to determine which groups differ.

#### **5. Comparison of Tests**

##### **Key Differences**

* **T-tests:** Compare two means.
* **Chi-Square:** Tests independence between categorical variables.
* **ANOVA:** Compares means across multiple groups.

##### **Applications**

* T-tests and ANOVA are used for numerical data, while Chi-square is used for categorical data.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 23: SVDDimReduction.pdf**

#### **1. Dimensionality Reduction Using SVD**

* Singular Value Decomposition (SVD) reduces the dimensionality of data by retaining the most significant singular values and corresponding singular vectors.
* For a matrix **A**, SVD is expressed as:
  + **A = U \* Σ \* Vᵀ***(Equation in Green)*: **A = U \* Σ \* Vᵀ**
  + **U**: Left singular vectors (rows' latent space).
  + **Σ**: Diagonal matrix of singular values (variance captured by each dimension).
  + **V**: Right singular vectors (columns' latent space).

#### **2. Steps for Dimensionality Reduction with SVD**

* **Step 1:** Perform SVD on the data matrix **A**.
* **Step 2:** Truncate **Σ** by keeping only the largest **k** singular values, forming **Σₖ**.
* **Step 3:** Retain the corresponding columns of **U** and **V**, forming **Uₖ** and **Vₖ**.
* **Step 4:** Approximate the original matrix **A** as:
  + **Aₖ = Uₖ \* Σₖ \* Vₖᵀ***(Equation in Green)*: **Aₖ = Uₖ \* Σₖ \* Vₖᵀ**

#### **3. Interpretation of Singular Values**

* The magnitude of singular values represents the importance of each corresponding component.
* Larger singular values capture more variance, while smaller ones often correspond to noise.

#### **4. Benefits of SVD for Dimensionality Reduction**

* **Noise Filtering:** Discards less significant singular values and vectors associated with noise or redundancy.
* **Compression:** Reduces storage and computational complexity by approximating **A** with a lower-rank matrix **Aₖ**.
* **Feature Extraction:** Extracts latent features from the data.

#### **5. Applications of SVD in Dimensionality Reduction**

* **Image Compression:** SVD is used to compress images by keeping only the most significant singular values and vectors.
* **Latent Semantic Analysis (LSA):** Reduces the dimensionality of term-document matrices in Natural Language Processing (NLP).
* **Recommender Systems:** Identifies latent user and item features for collaborative filtering.

#### **6. Reconstruction Error**

* The quality of the approximation **Aₖ** is measured by the Frobenius norm of the difference:
  + **||A - Aₖ||ᶠ***(Equation in Green)*: **||A - Aₖ||ᶠ**
* Lower error indicates better reconstruction.

#### **7. Practical Considerations**

* Choosing **k**, the number of singular values to retain, involves a trade-off between information preservation and dimensionality reduction.
* Retain enough singular values to capture a high percentage of the total variance (e.g., 95%).

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **File 24: ThreeStatisticalTests.pdf**

#### **1. Overview of Statistical Tests**

* The document highlights three essential statistical tests: t-tests, Chi-square tests, and ANOVA. These tests are designed for different types of data and hypotheses.

#### **2. T-Test**

##### **Purpose of a T-Test**

* Compares the means of groups to determine if they differ significantly.

##### **Types of T-Tests**

* **One-Sample T-Test:** Compares the sample mean to a known or hypothesized population mean.
* **Independent T-Test:** Compares the means of two independent groups.
* **Paired T-Test:** Compares the means of two related groups (e.g., pre-test vs. post-test).

##### **Formula for the T-Test Statistic**

* General formula:
  + **t = (difference in Means) / (Standard Error of the Difference)***(Equation in Green)*: **t = (difference in Means) / (Standard Error of the Difference)**

##### **Assumptions of the T-Test**

* Data is normally distributed.
* Variances of the two groups are equal (for an independent t-test).

#### **3. Chi-Square Test**

##### **Purpose of the Chi-Square Test**

* Tests for independence between two categorical variables.

##### **Test Statistic Formula**

* The Chi-square test statistic is calculated as:
  + **χ² = Σ ((O - E)² / E)***(Equation in Green)*: **χ² = Σ ((O - E)² / E)**
* where:
  + **O**: Observed frequency.
  + **E**: Expected frequency.

##### **Applications of the Chi-Square Test**

* Determining whether two variables (e.g., gender and voting preference) are independent.
* Testing the goodness-of-fit of observed data to an expected distribution.

##### **Key Assumptions of the Chi-Square Test**

* Expected frequencies in each cell of the contingency table should be **≥ 5**.

#### **4. ANOVA (Analysis of Variance)**

##### **Purpose of ANOVA**

* Compares the means of three or more groups to determine if there are significant differences.

##### **Types of ANOVA**

* **One-Way ANOVA:** Analyzes differences across one factor with multiple levels (e.g., comparing test scores among three teaching methods).
* **Two-Way ANOVA:** Examines interactions between two factors (e.g., test scores by teaching method and student gender).

##### **ANOVA Test Statistic Formula**

* ANOVA uses the **F-statistic**:
  + **F = (Between-Group Variance) / (Within-Group Variance)***(Equation in Green)*: **F = (Between-Group Variance) / (Within-Group Variance)**

##### **Post-Hoc Testing**

* If ANOVA reveals significant differences, post-hoc tests (e.g., Tukey’s HSD) are used to determine which groups differ.

#### **5. Comparison of the Three Tests**

##### **Key Differences**

* **T-Tests:** Compare means of two groups.
* **Chi-Square:** Tests independence between categorical variables.
* **ANOVA:** Compares means of three or more groups.

##### **Applications**

* T-tests and ANOVA are applied to numerical data, while Chi-square tests are used for categorical data.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 