CS57300
PURDUE UNIVERSITY

JANUARY 15, 2019

DATA MINING

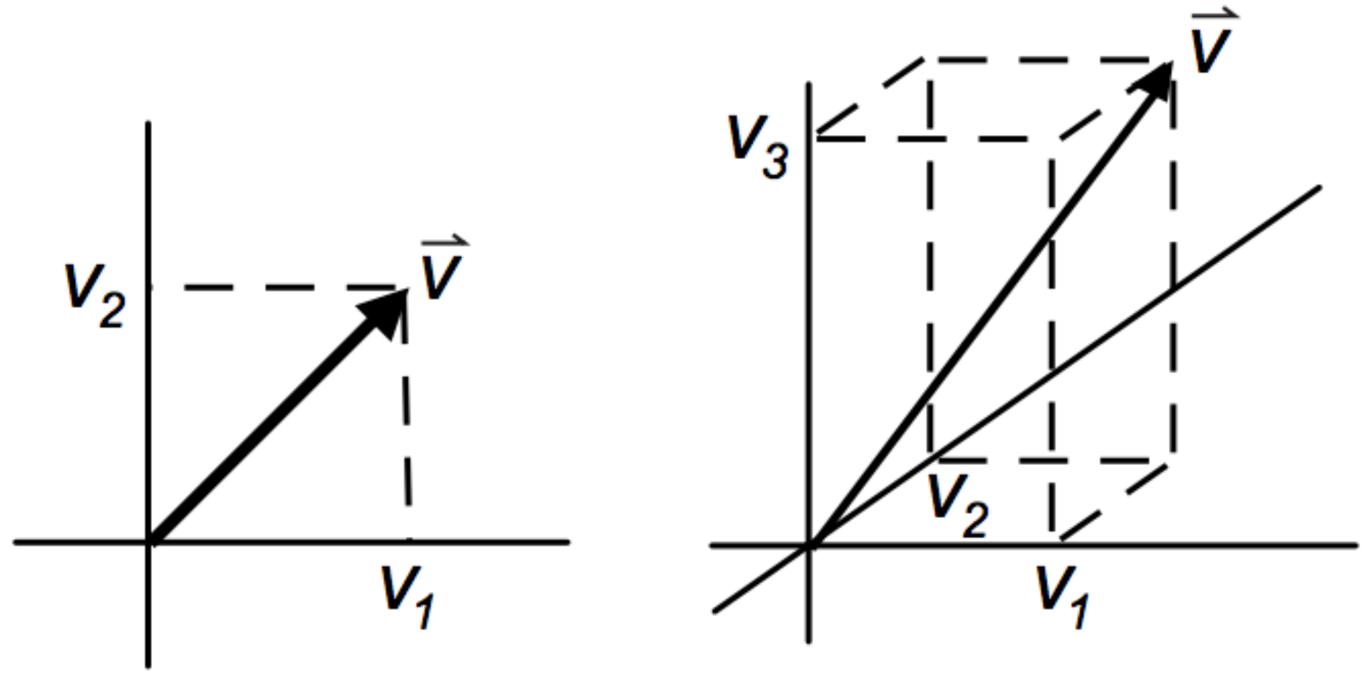
LINEAR ALGEBRA

VECTORS

- A **vector** is a 1D array of values

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- We use the notation x_i to denote the i th entry of x
- Vectors can be graphically depicted as arrows in n-dimensional space

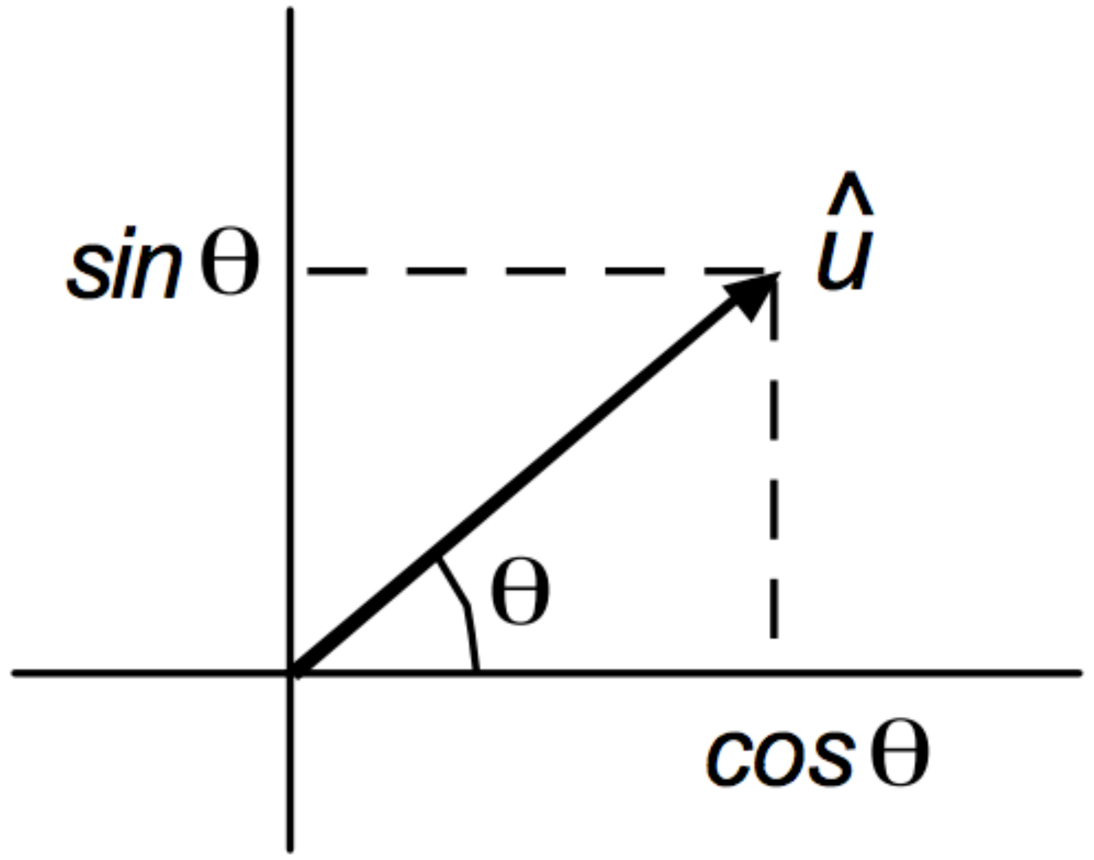


- The ***norm*** (length) of a vector is defined as $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$

MORE ON VECTORS

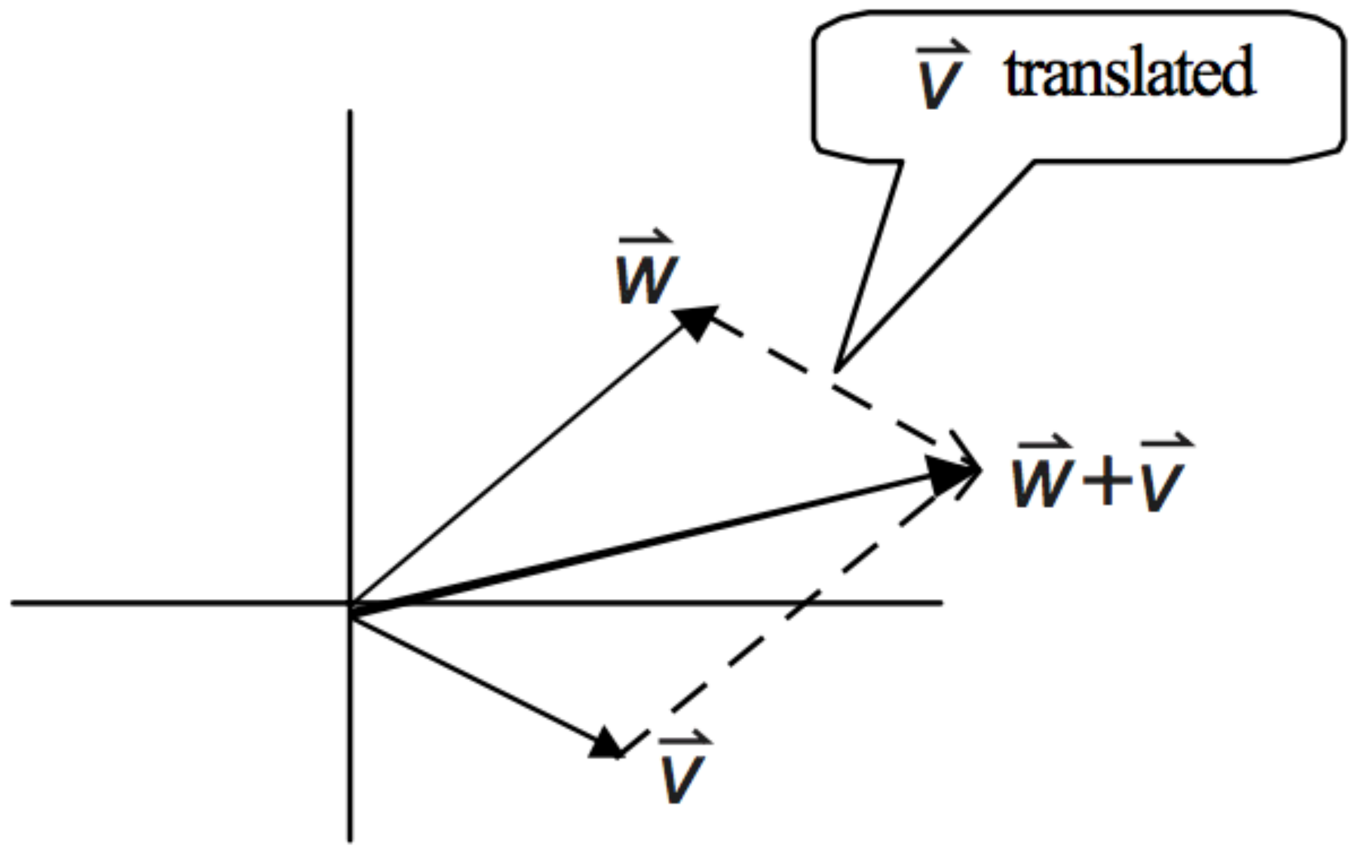
- A ***unit vector*** is a vector of length 1. A 2-D unit vector can be parameterized as:

$$\hat{u}(\theta) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$$



- Multiplying a vector by a scalar simply changes the length of the vector by that factor $\|a\boldsymbol{x}\| = |a|\|\boldsymbol{x}\|$ (when *a* is negative, the direction of the vector is reversed)

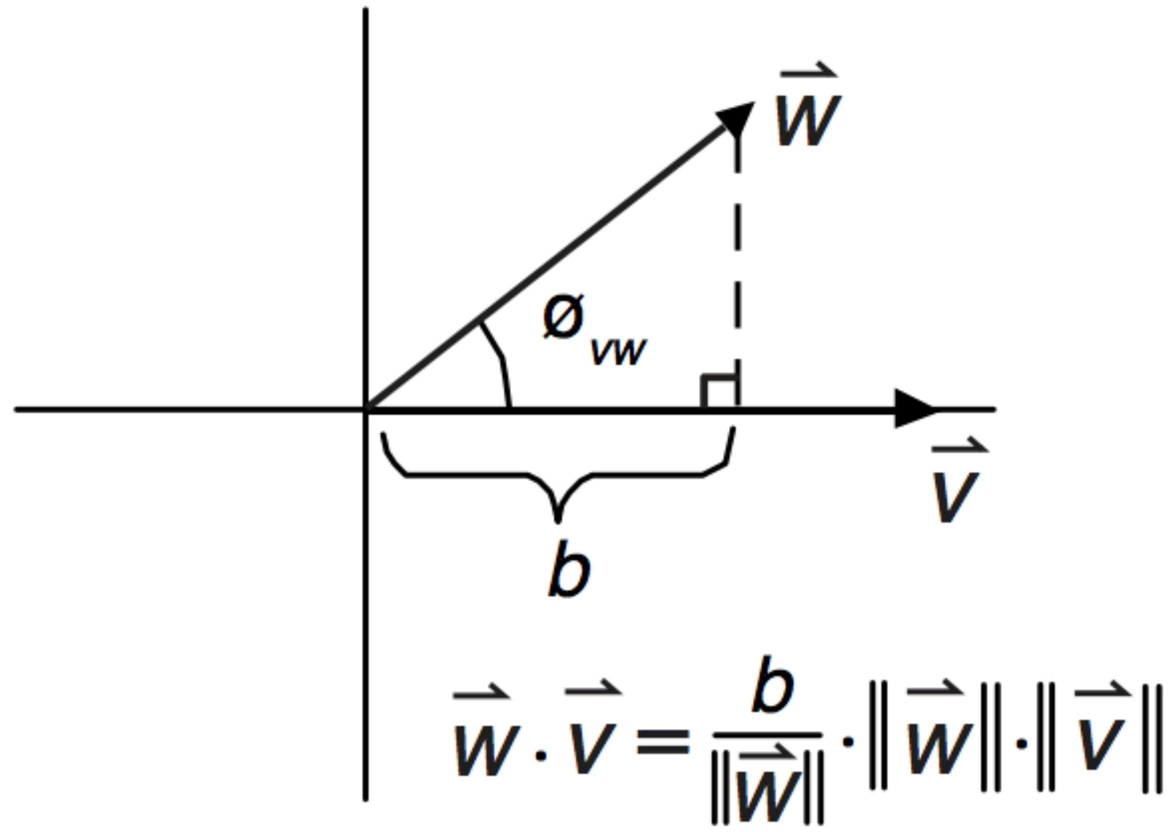
- Vector addition: $z = w + v \Leftrightarrow z_i = w_i + v_i$



INNER PRODUCT

- The ***inner product*** of two vectors is the sum of pairwise product of components

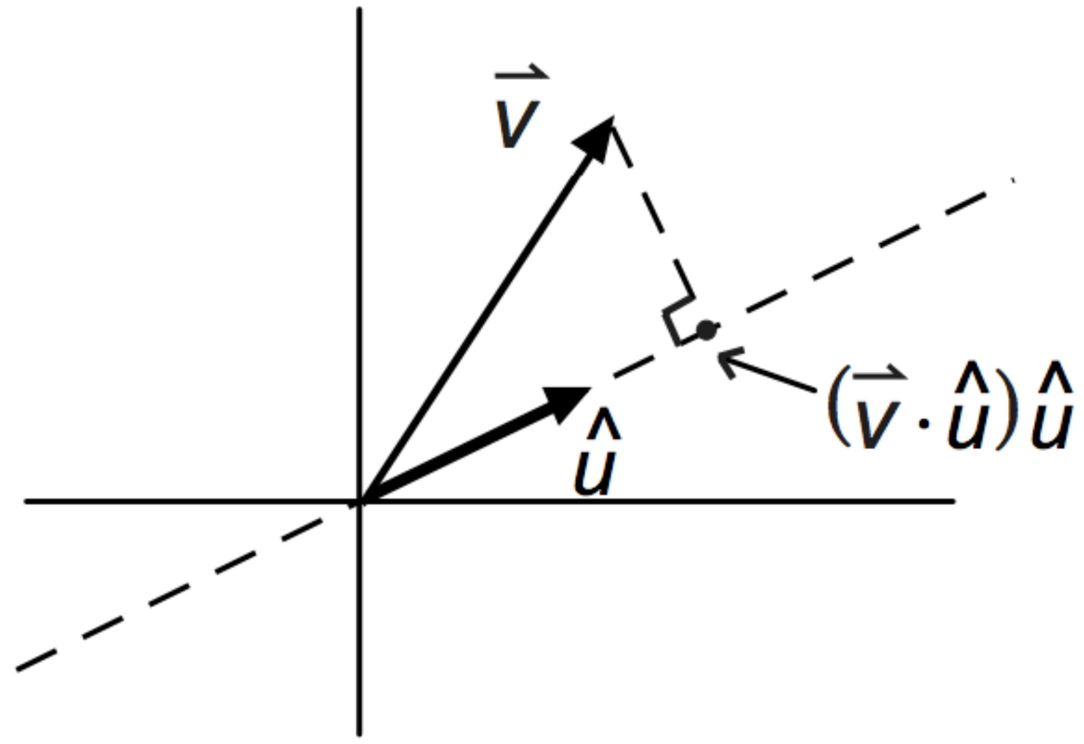
$$w \cdot v = \sum_{i=1}^{n} w_i v_i$$



- Its equivalent geometric definition is:

$$w \cdot v = \|w\| \|v\| \cos(\emptyset_{vw})$$

- The inner product of a vector *v* with a unit vector *u* is the length of *v*'s projection on *u*.



- Two vectors are *orthogonal* to each other if their inner product is 0.

VECTOR SPACE

- A vector space can be **spanned** by a set of vectors iff one can write any vector in the vector space as a linear combination of the set
  - Can the 3D vector space be spanned by (1, 1, 0) and (0, 2, 3)?
- A set of vectors $\{v_1, v_2, \ldots, v_n\}$ is linearly independent iff the only solution to the following equation is $\alpha_k = 0$ (for all k)

$$\sum_{k=1}^{n} \alpha_k v_k = 0$$

BASIS

- A **basis** for a vector space is a linearly independent spanning set.
  - Is (1, 1, 0), (0, 2, 3), (0, 1, 0), (2, 5, 3) a basis for the 3D vector space?
- The **standard basis** of a vector space is the set of unit vectors that lie along the axes of the space
  - $e_1=(1, 0, \ldots, 0)$, $e_2=(0, 1, \ldots, 0)$, …, $e_n=(0, 0, \ldots, 1)$

MATRICES

- A **matrix** is a 2D array of values
- We use A_{ij} to denote the entry in row i and column j
- Higher dimensional matrices are called tensors

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

BASIC MATRIX OPERATIONS

For $A, B \in \mathbb{R}^{m \times n}$, matrix addition/subtraction is just the elementwise addition or subtraction of entries

$$C \in \mathbb{R}^{m \times n} = A + B \iff C_{ij} = A_{ij} + B_{ij}$$

For $A \in \mathbb{R}^{m \times n}$, transpose is an operator that "flips" rows and columns

$$C \in \mathbb{R}^{n \times m} = A^T \iff C_{ji} = A_{ij}$$

For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$ matrix multiplication is defined as

$$C \in \mathbb{R}^{m \times p} = AB \iff C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

Note: Matrix multiplication is associative ($A(BC) = (AB)C$), distributive ($A(B+C) = AB + AC$), *not commutative* ($AB \neq BA$)

SPECIAL TYPES OF MATRICES

- A **square matrix** is a matrix with the same number of rows and columns
- A **diagonal matrix** is a matrix for which all entries outside the main diagonal are zero

IDENTITY AND INVERSE MATRIX

The identity matrix $I \in \mathbb{R}^{n \times n}$ is a square matrix with ones on diagonal and zeros elsewhere, has property that for $A \in \mathbb{R}^{m \times n}$

$$AI = IA = A \text{ (for different sized } I\text{)}$$

ORTHOGONAL MATRIX

- An **orthogonal matrix** is a square matrix for which every column is a unit vector, and every pair of columns is orthogonal.
  - If A is an orthogonal matrix, then

$$A^T A = I \quad \text{and} \quad A^{-1} = A^T \quad \text{and} \quad AA^T = I$$

  - So A^T is also an orthogonal matrix, which means that every row of A is a unit vector and every pair of rows of *A* is orthogonal.

OTHER DEFINITIONS/PROPERTIES

Transpose of matrix multiplication, $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$

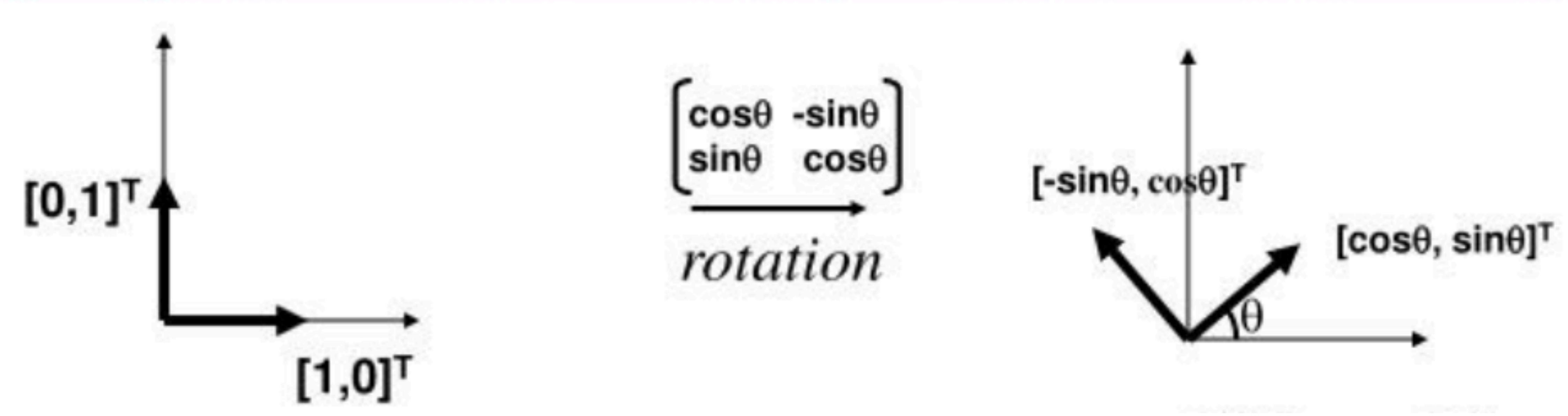
$$(AB)^T = B^T A^T$$

Inverse of product, $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times n}$ *both square and invertible*

$$(AB)^{-1} = B^{-1} A^{-1}$$

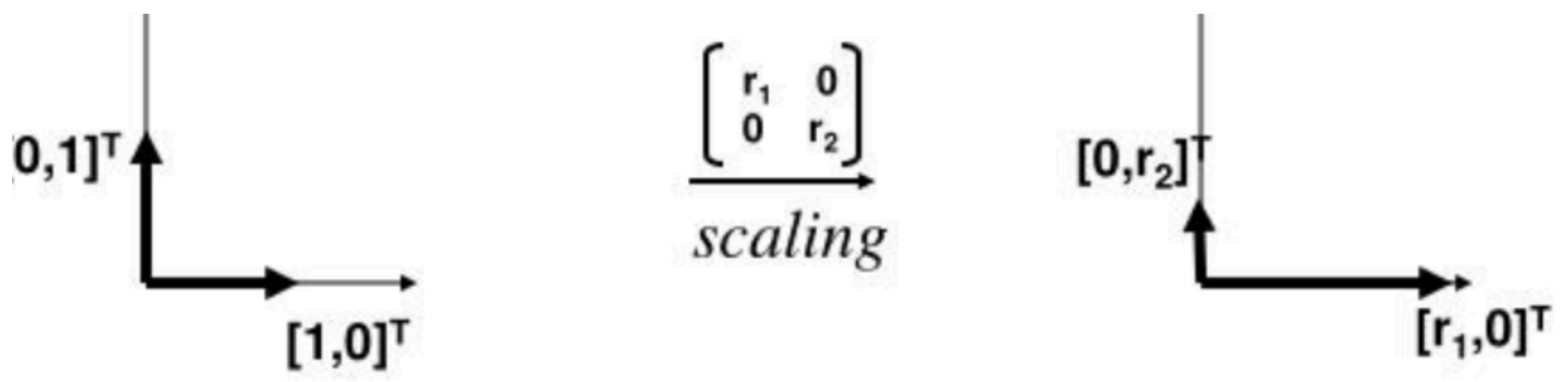
REPRESENTING LINEAR TRANSFORMATION USING MATRICES

- When A is an orthogonal matrix, Ax rotates x



Can also be interpreted as change of basis

- When A is an diagonal matrix, Ax stretch or squeeze the axes



- More general square matrix involves both rotation and scaling

EIGENVALUES AND EIGENVECTORS

- An **eigenvector** is a non-zero vector that changes by only a scalar factor when a particular linear transformation is applied to it, and the scalar is **eigenvalue**.

$$Ax = \lambda x$$

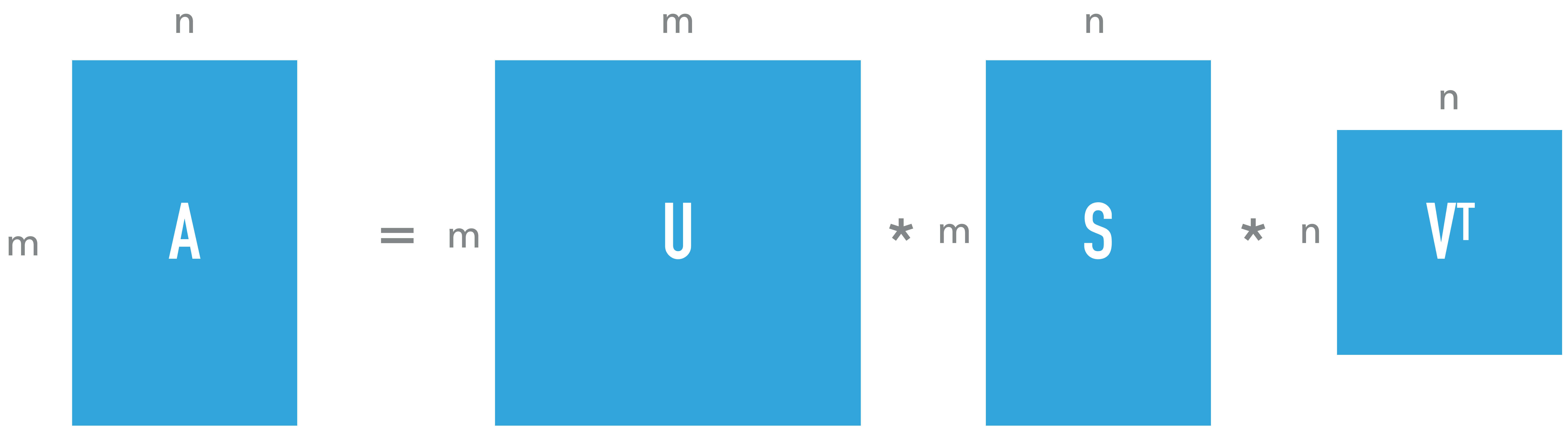
- How to calculate eigenvalues and eigenvectors?
  - $(A - \lambda I)x = 0$. Let the determinant of $A - \lambda I$ be 0.

EIGENDECOMPOSITION

- Let A be a square matrix with N linearly independent eigenvectors, q_i (i=1, …, N). Then A can be factorized as:
  - $A = Q\Lambda Q^{-1}$
  - Q is the square matrix whose i-th column is the eigenvector q_i of A, Λ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e., $\Lambda_{ii} = \lambda_i$
  - For a symmetric matrix A, Q is an orthogonal matrix, that is, $A = Q\Lambda Q^T$

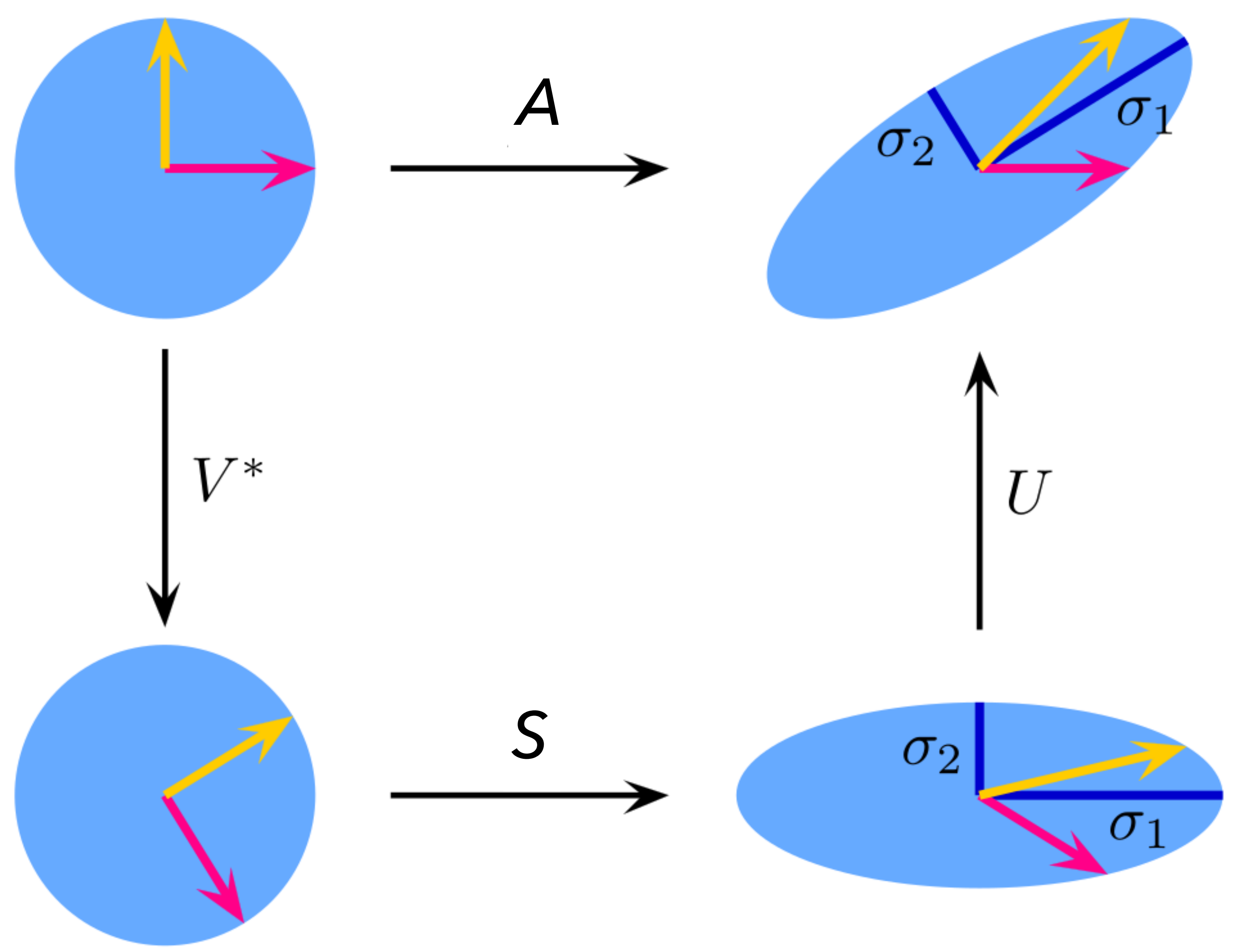
SIGULAR VALUE DECOMPOSITION (SVD)

- A rectangular matrix *A* can be broken down into the product of three matrices: an orthogonal matrix *U*, a diagonal matrix *S*, and the transpose of an orthogonal matrix *V*.



SINGULAR VALUE DECOMPOSITION (SVD)

- Columns of U are eigenvectors of AA^T
- Columns of V are eigenvectors of A^TA
- Diagonal entries of S are the square roots of the non-zero eigenvalues of AA^T (as well as A^TA)
- Geometric interpretation:



DISTANCE MEASURES

REPRESENTING DATA IN EUCLIDEAN SPACE

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute

- Many data mining techniques then use similarity/dissimilarity measures to characterize relationships between the instances

Height	Weight	Heart Rate	BP (Diastolic)	BP (Systolic)
1.79	80	70	73	112
1.60	51	73	69	105

DISTANCE MEASURES

- Many data mining techniques utilize similarity/dissimilarity measures to characterize relationships between instances
    - Nearest-neighbor classification
    - Cluster analysis
- **Proximity**: general term to indicate similarity and dissimilarity
- **Distance**: dissimilarity only

METRIC PROPERTIES

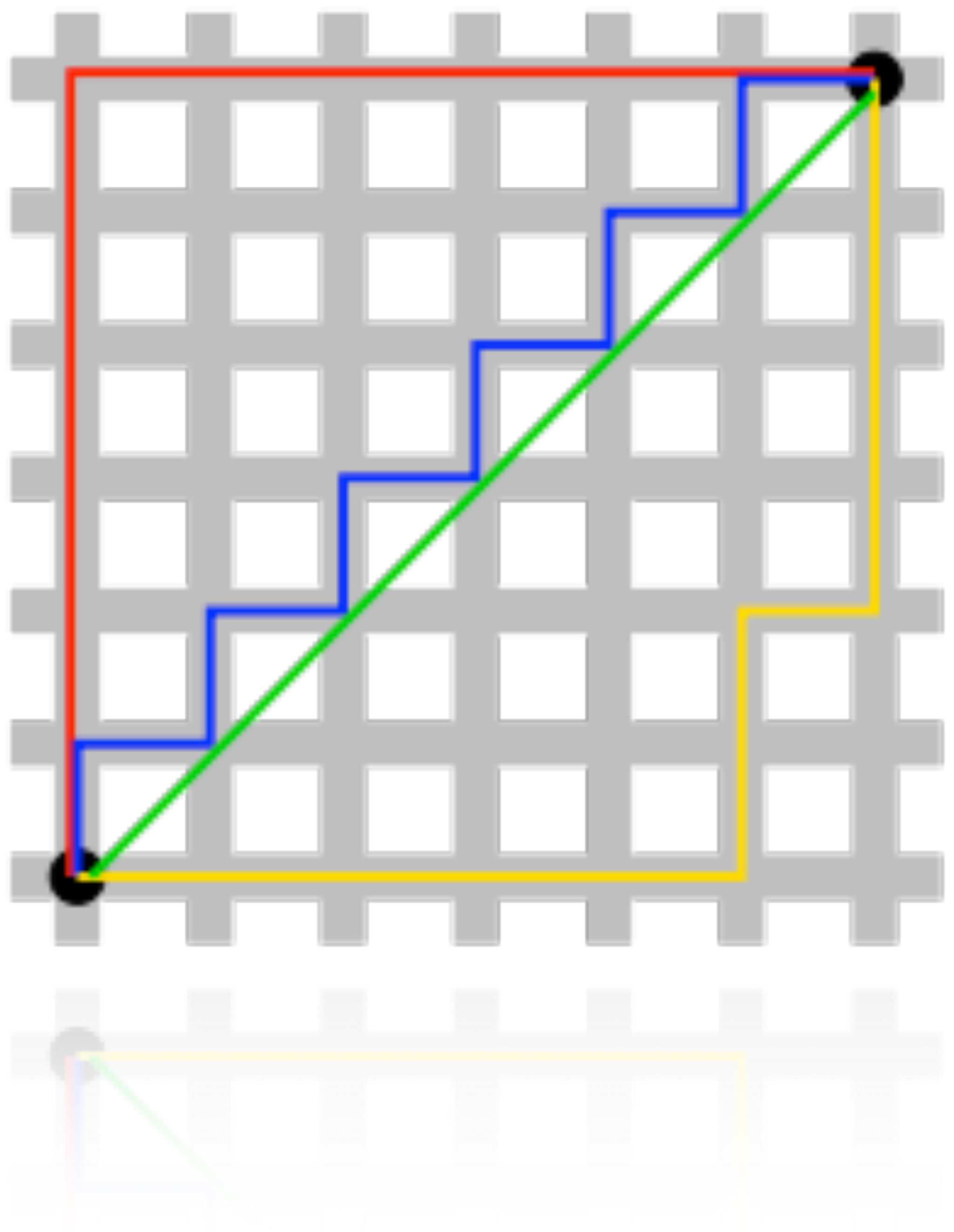
- A **metric** $d(x,y)$ (or a distance function) is a function that satisfies the following properties:
  - $d(x,y) \geq 0$ for all x,y and $d(x,y)=0$ iff $x=y$ **Positivity**
  - $d(x,y) = d(y,x)$ for all x,y **Symmetry**
  - $d(x,y) \leq d(x,k)+d(k,y)$ for all x,y,k **Triangle inequality**

DIFFERENT TYPES OF METRICS

- Manhattan distance (L1) $d_M(x,y) = \sum_{i=1}^{p} |x_i - y_i|$
- Euclidean distance (L2) $d_E(x,y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$
  - Most common metric
  - Assumes dimensions are commensurate
- **Weighted** Euclidean distance

$$d_{WE}(x,y) = \sqrt{\sum_{i=1}^{p} w_i (x_i - y_i)^2}$$

  - Can weight variables by relative importance



STANDARDIZATION

- Normalization
  - Removes effect of scale

$$x'_k = \frac{x_k - \bar{x}_k}{\hat{\sigma}_k}$$

subtract mean

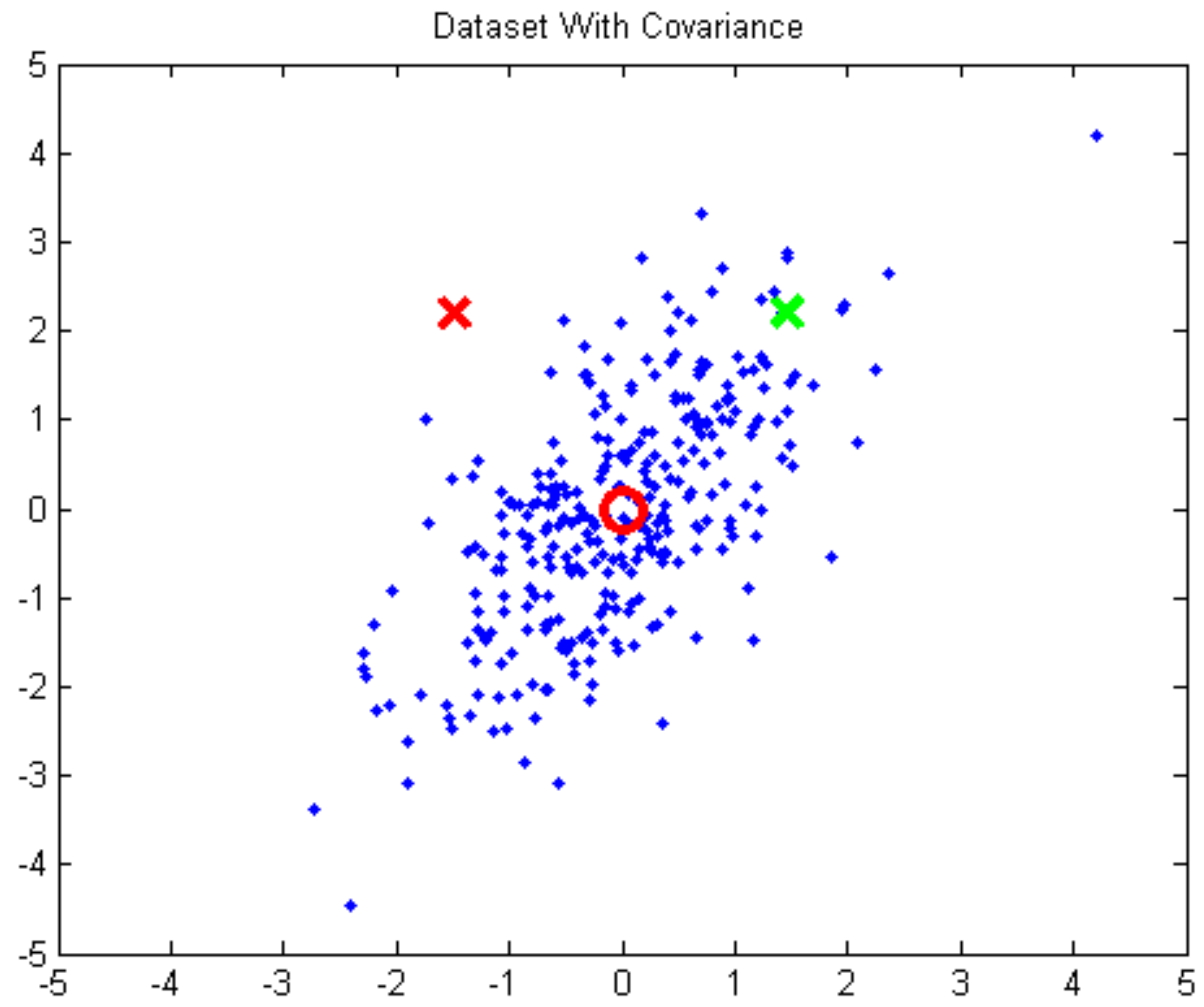
divide by stdev

  - Divide each variable by its standard deviation
  - Weights all variables equally

$$d'_E(x, y) = \sqrt{\sum_{i=1}^{p} (x'_i - y'_i)^2}$$

CORRELATION AMONG VARIABLES

- Variables contribute independently to additive measure of distance
- May not be appropriate if variables are highly correlated
- Can standardize variables in a way that accounts for covariance



MAHALANOBIS DISTANCE

$$d_{MH}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

pxp covariance matrix

- Automatically accounts for scaling
- Corrects for correlation between attributes
- Tradeoff:
  - Covariance matrix can be hard to estimate accurately
  - Memory and time complexity is quadratic rather than linear

DISTANCE MEASURES FOR BINARY DATA

- d(x,y) when items x and y are *p*-dimensional binary vectors
- Let n_{11} be the number of attributes where both items have value 1, etc.

$$n_{11} = \sum_{i}^{p} \mathbb{I}(x_i + y_i = 2)$$

	y=1	y=0
x=1	n_{11}	n_{11}
x=0	n_{10}	n_{00}

- Matching distance
  - Hamming distance normalized by number of bits

$$d_M(x,y) = 1 - \frac{n_{11} + n_{00}}{n_{11} + n_{00} + n_{10} + n_{01}}$$

- Jaccard distance
  - If we don't care about matches on zeros

$$d_M(x,y) = 1 - \frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$

NEXT CLASS

- Review basic knowledge on sampling and statistical inference