machine learning

Feature Selection and Dimensionality Reduction

Lecture VII

פיתוח:
ד"ר יהונתן שלר
משה פרידמן

תודות לד"ר יונתן רובין שעזר בהכנת המצגת

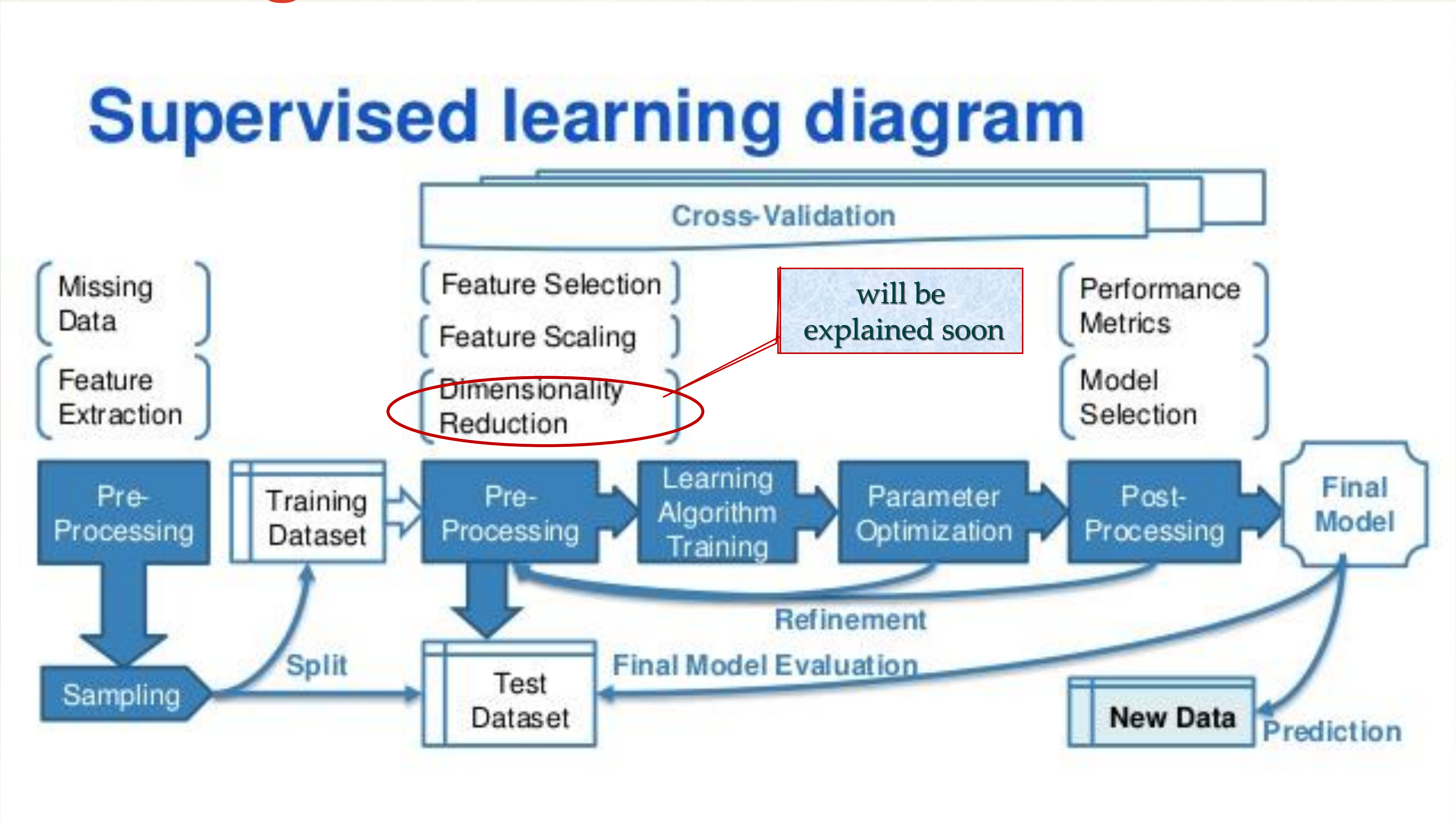
סוגי בעיות בלמידה לא מונחית - חזרה

Clustering: represent each input case using a prototype example (e.g., k-means, mixture models)

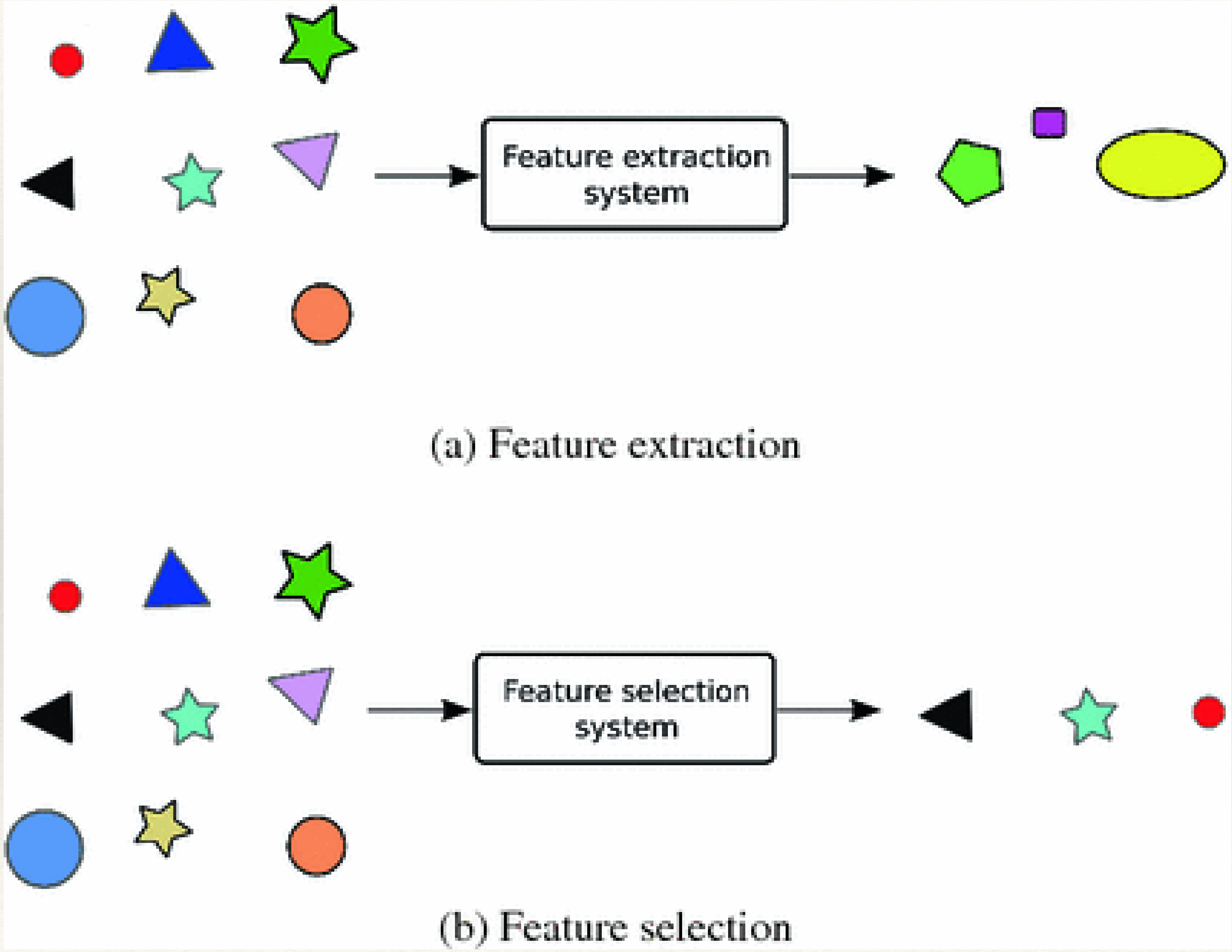
Dimensionality reduction: represent each input case using a small number of variables (e.g., principal components analysis, factor analysis, independent components analysis)

Density estimation: estimating the probability distribution over the data space

a typical supervised learning flow - diving in



Feature selection



(a) Feature extraction

(b) Feature selection

Feature selection – techniques - reminder

1. Low Variance
2. Remove highly correlated features
3. Select features with high correlation to target

Feature selection – techniques
4. Recursive feature elimination

Feature selection using classification errors

Wrapper approach:

- The feature selection is driven by the prediction accuracy of the classifier (regressor) actually used

How to find the appropriate feature set?

- **Idea: Greedy search in the space of classifiers**
  - Gradually add features improving most the quality score
  - Score should reflect the accuracy of the classifier (error) and also prevent overfit
- **Two ways to measure overfit**
  - Regularization: penalize explicitly for each feature parameter
  - Cross-validation (m-fold cross validation)

Dimensionality Reduction

What is the difference between "simple" feature selection and dimensionality reduction?

- The difference is that the set of features made by feature selection must be a subset of the original set of features, and the set made by dimensionality reduction doesn't have to

Dimensionality Reduction

What is the difference between "simple" feature selection and dimensionality reduction?

- Feature selection: Choosing $k<d$ important features, ignoring the remaining $d - k$
  - Subset selection algorithms
- dimensionality reduction project the original x_i , i =1,...,d dimensions to new $k<d$ dimensions, z_j , j =1,...,k
  - Principal Components Analysis (PCA) – explained later

Dimensionality Reduction – example

Classification problem example:

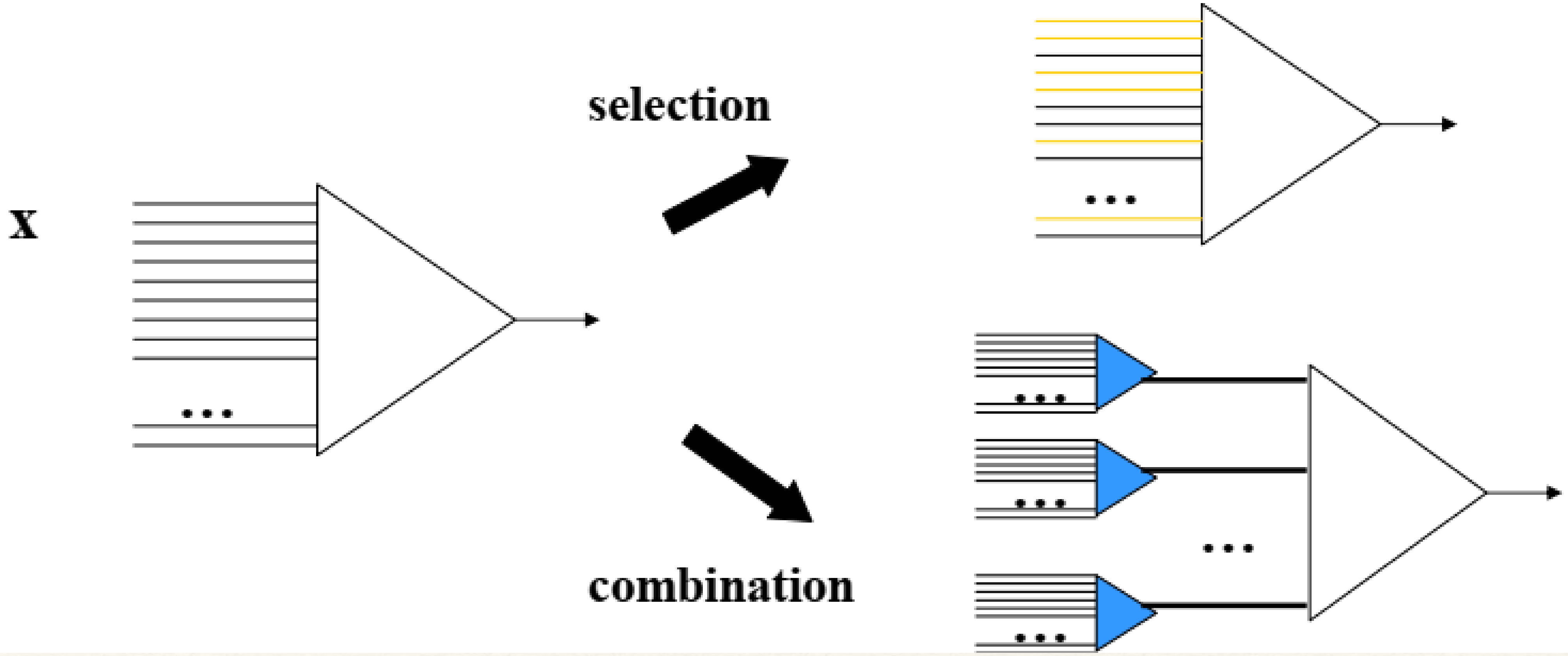
- We have an input data $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ such that

$$\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^d)$$

  and a set of corresponding output labels $\{y_1, y_2, ..., y_N\}$
- Assume the dimension d of the data point $\boldsymbol{x}$ is very large
- We want to classify $\boldsymbol{x}$

Dimensionality Reduction – example

- **Solutions**:
  - **Selection of a smaller subset** of inputs (features) from a large set of inputs; train classifier on the reduced input set
  - **Combination of high dimensional inputs** to a smaller set of features $\phi_k(\mathbf{x})$; train classifier on new features



הורדת מימדים (dimension reduction) – מוטיבציה - Data Compression

Motivation I: Data Compression

- We may want to reduce the dimension of our features if we have a lot of redundant data.
- Dimensionality reduction will reduce the total data we have to store in computer memory and will speed up our learning algorithm.

הורדת מימדים (dimension reduction) – מוטיבציה - Data Compression

- If number of observables is increased
  - More time to compute
  - More memory to store inputs and intermediate results
  - More complicated explanations (knowledge from learning)
    - Regression from 100 vs. 2 parameters
  - No simple visualization
    - 2D vs. 10D graph
  - **Need much more data (curse of dimensionality)**
    - 1M of 1-d inputs is not equal to 1 input of dimension 1M

הורדת מימדים (dimension reduction) – מוטיבציה - Data Compression - הסבר

- Some features (dimensions) bear little or nor useful information (e.g. color of hair for a car selection)
  - Can drop some features
  - Have to estimate which features can be dropped from data

- Several features can be combined together without loss or even with gain of information (e.g. income of all family members for loan application)
  - Some features can be combined together
  - Have to estimate which features to combine from data

הורדת מימדים (dimension reduction) – מוטיבציה - <u>Data Compression</u> - שימוש

- Have data of dimension d
- Reduce dimensionality to k<d
  - Discard unimportant features (we saw this also before)
  - Combine several features in one
- Use resulting k-dimensional data set for
  - Learning for classification problem (e.g. parameters of probabilities P(x|C)
  - Learning for regression problem (e.g. parameters for model y=g(x|Thetha)

הורדת מימדים (dimension reduction) –דוגמה

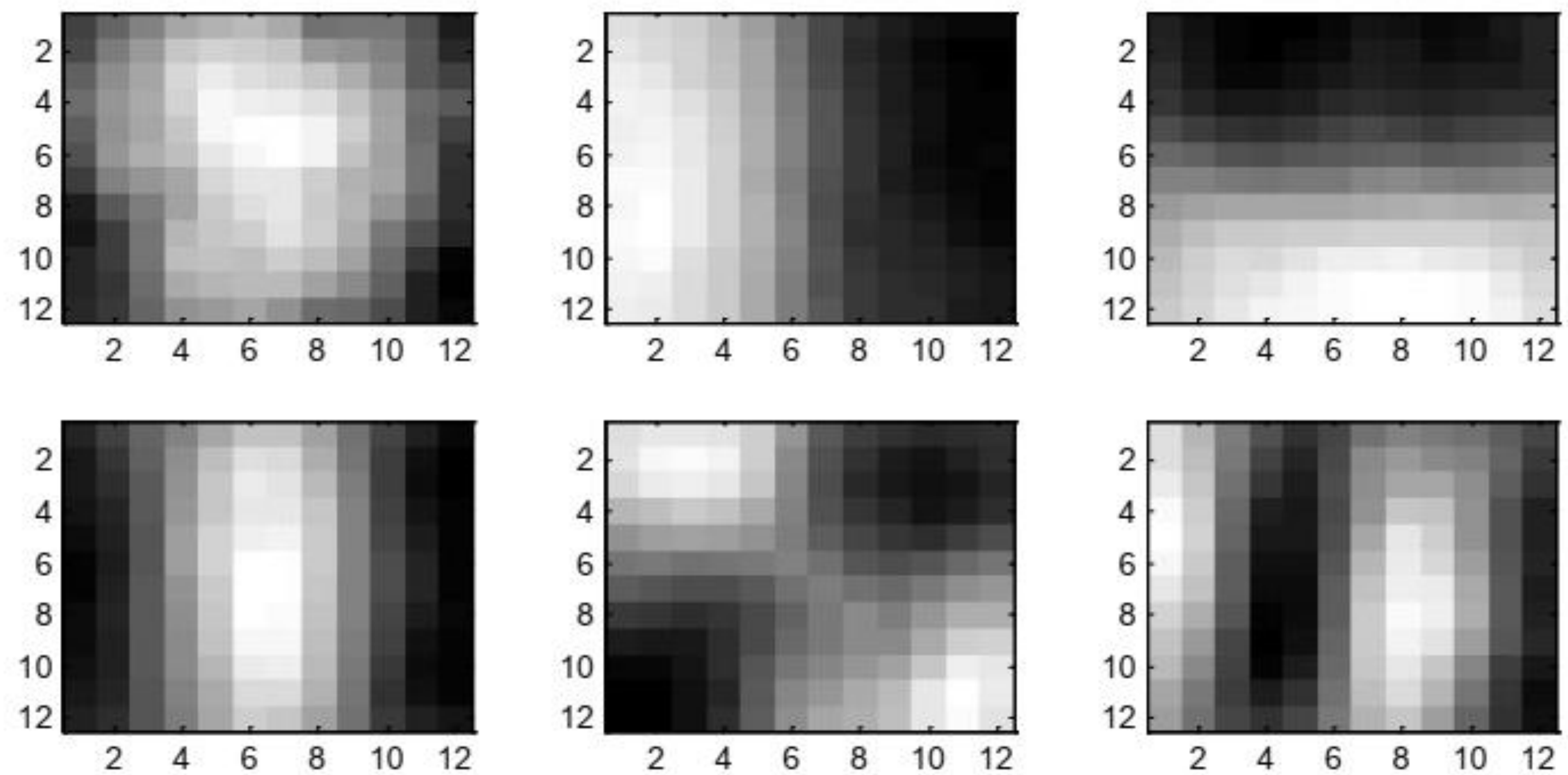
- ❑ Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
- ❑ Consider each as a 144-D vector

הורדת מימדים (dimension reduction) –דוגמה

הורדת המימדים D144 --> D6

הורדת מימדים (dimension reduction) –דוגמה

6 most important eigenvectors:

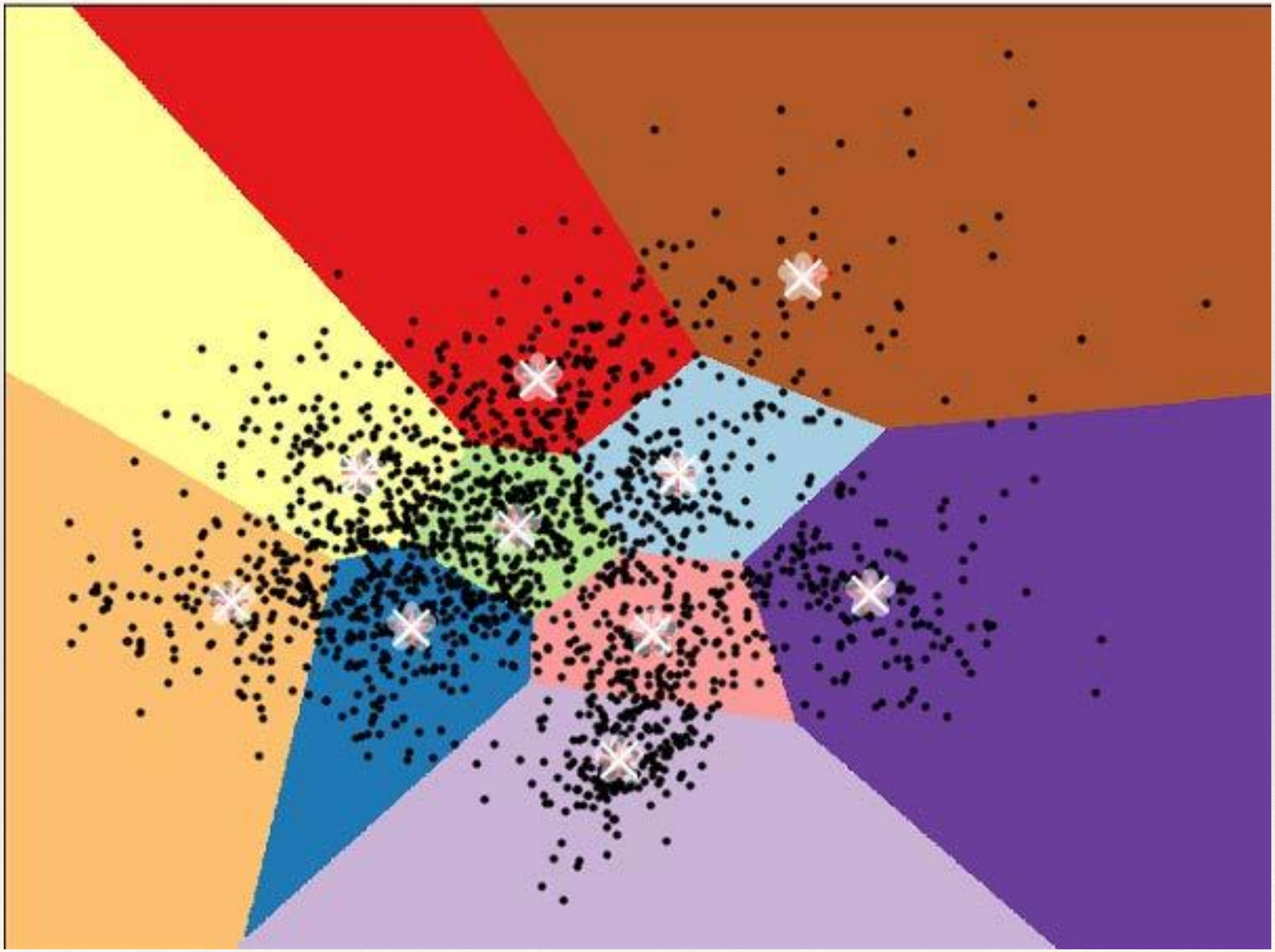


הורדת מימדים (dimension reduction) – מוטיבציה - Visualization

Motivation II: Visualization

- It is not easy to visualize data that is more than three dimensions. We can reduce the dimensions of our data to 3 or less in order to plot it
- We need that can deficiently summarize all the other features.
- Data exploration - The right visualization method may reveal problems with the experimental data.

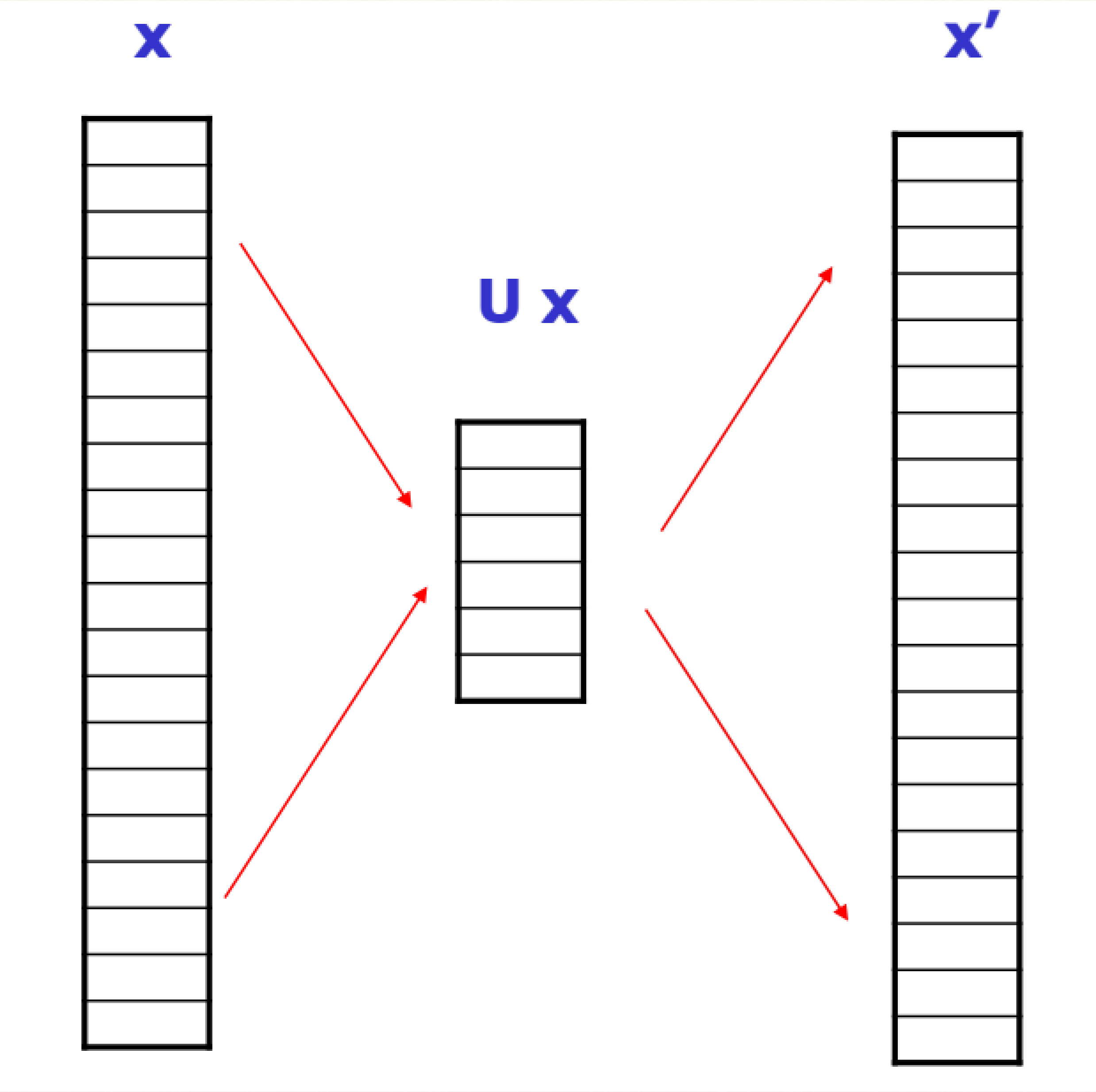
הורדת מימדים (dimension reduction) – Visualization – דו' עבור k-mean (k=10)



הורדת מימדים (dimension reduction) – מוטיבציה - Noise reduction

Motivation III: Noise reduction

- By selecting most significant eigenvectors and reproducing



הורדת מימדים (dimension reduction) – מוטיבציה - Noise reduction - דוגמה

Noisy image

הורדת מימדים (dimension reduction) – מוטיבציה - Noise reduction - דוגמה

De-noised image

הורדת מימדים (dimension reduction) – מוטיבציה – Deriving new data

Motivation IV: Deriving new data

- Here the goal is opposite from feature selection, the goal here is to find correlation within the features in order to find new knowledge

הורדת מימדים (dimension reduction) – מוטיבציה – Deriving new data - דוגמה

- Vector Representation

We can define a word by a vector of counts over contexts, For Example:

	song	cucumber	meal	black
tomato	**0**	**6**	**5**	**0**
book	2	0	2	3
pizza	**0**	**2**	**4**	**1**

- Each word is associated with a vector of dimension |V| (the size of the vocabulary)
- We expect similar words to have similar vectors
- Given the vectors of two words, we can determine their similarity (more about that later)

These vectors are:

- huge – each of dimension |V| (the size of the vocabulary ~ $100K +$)

הורדת מימדים (dimension reduction) – <u>הגדרה</u>

<u>הגדרת הורדת המימדים</u>:

- נתונות לנו n דוגמאות במימד d
- נרצה למצוא יצוג לכל הדוגמאות במימד d נמוך יותר $(k<d)$

<u>איך עושים זאת</u>?

הורדת מימדים (dimension reduction) – הגדרה

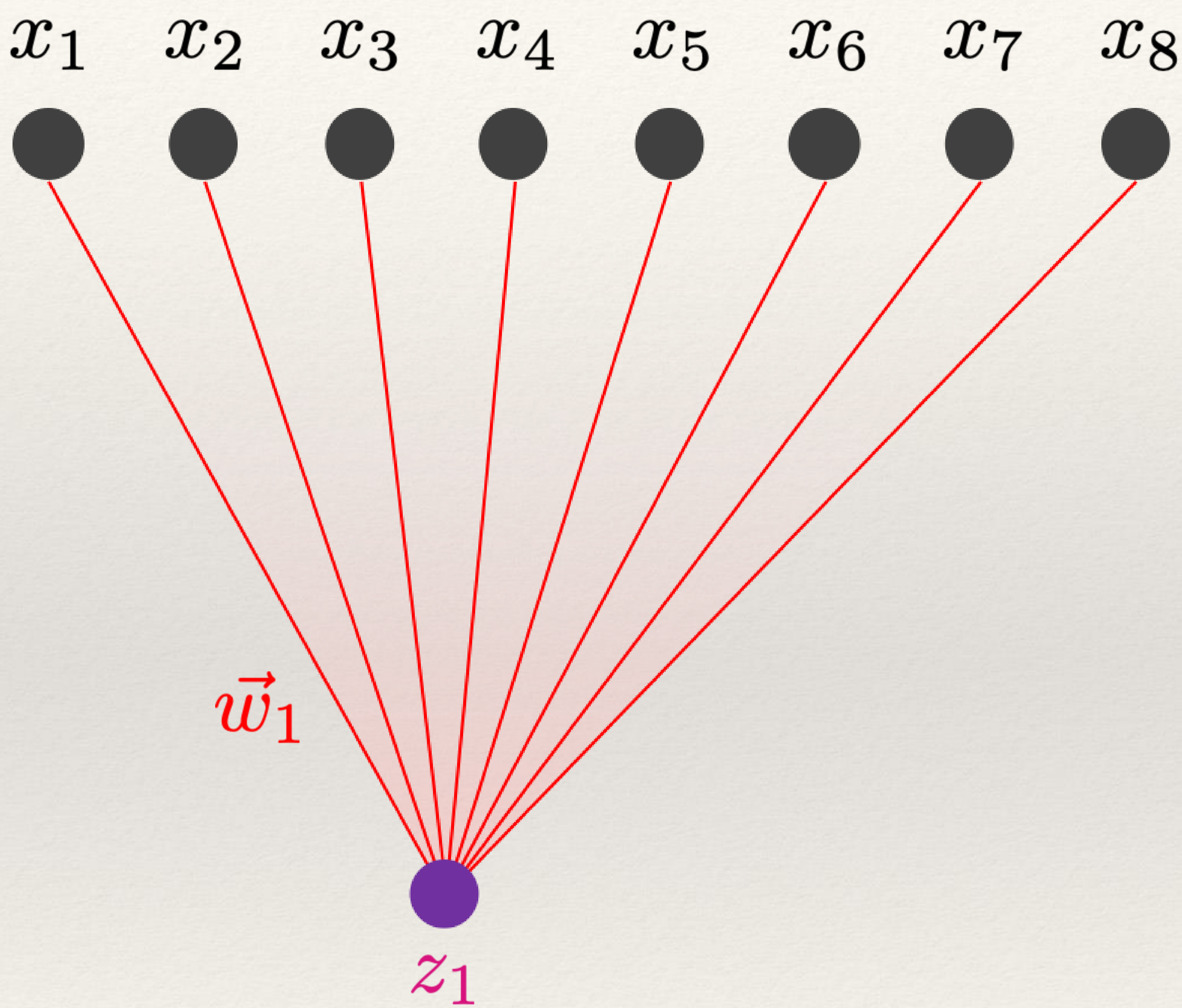
הגדרת הורדת המימדים:

- נתונות לנו n דוגמאות במימד d
- נרצה למצוא יצוג לכל הדוגמאות במימד d נמוך יותר ($k<d$)

איך עושים זאת? הטלה ממימד d למימד k

- PCA – דו' בה ההיטל מורכב מקומבינציות לינאריות של המאפיינים
- tSNE – דו' בה ההיטל מורכב קומבינציות לא לינאריות של המאפיינים

הורדת מימדים - PCA



$$z_1 = \vec{w}_1 \cdot \vec{x}$$

הורדת מימדים - PCA

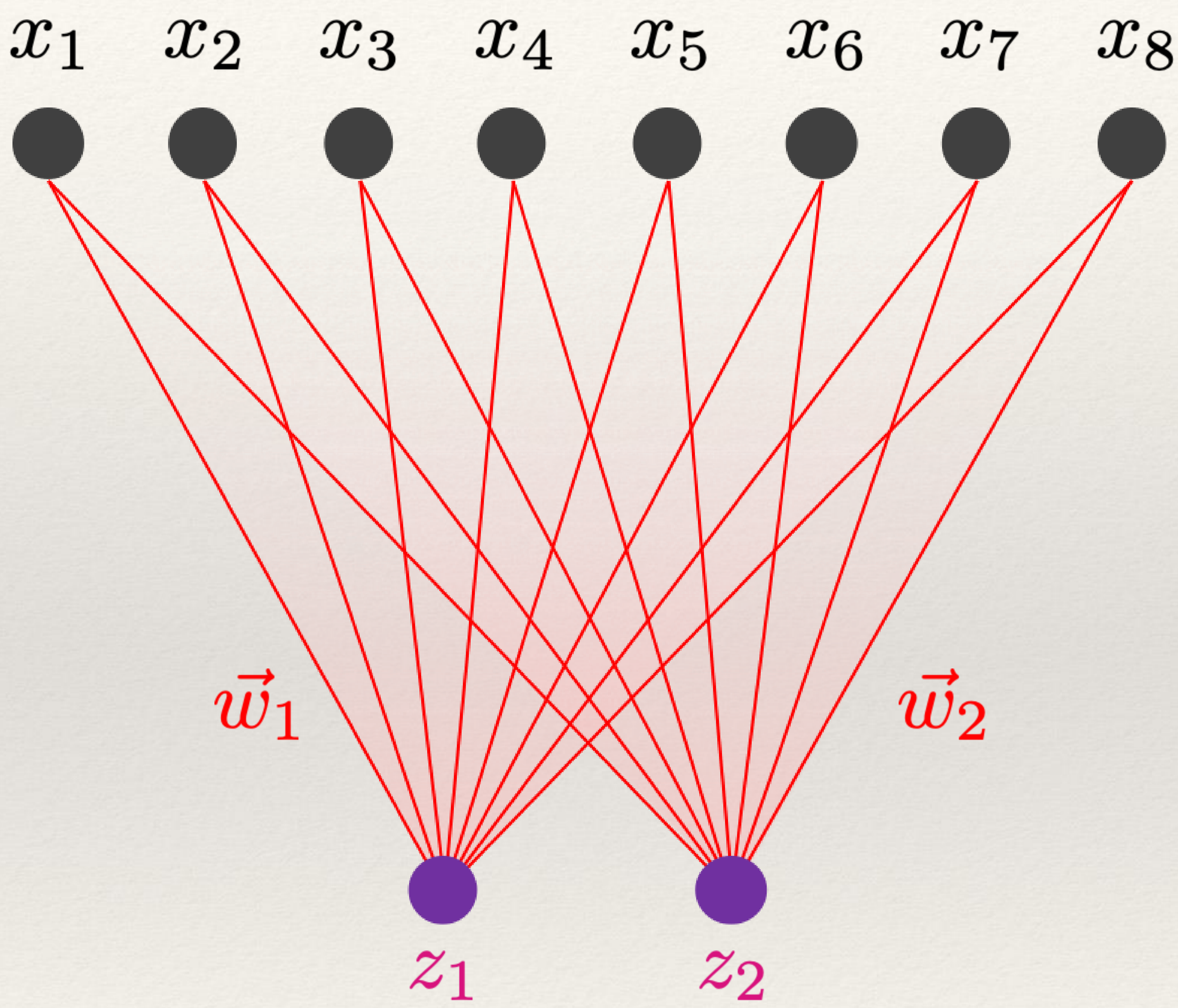
- PCA – Principal component analysis
- נתונות לנו n דוגמאות במימד d
- נרצה למצוא יצוג לכל הדוגמאות במימד נמוך יותר $(k<d)$
- האמצעי: קומבינציות לינאריות של המאפיינים
  כלומר: הטלה ממימד d למימד k

$$\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$$

$$z_{i,j} = \vec{w}_j \cdot \vec{x}_i$$

$$\vec{z}_i = \left(\vec{w}_1 \cdot \vec{x}_i \, , \, \vec{w}_2 \cdot \vec{x}_i \, , \, \dots \, , \, \vec{w}_k \cdot \vec{x}_i\right) = (z_{i,1}, z_{i,2}, \dots, z_{i,k})$$

הורדת מימדים - PCA



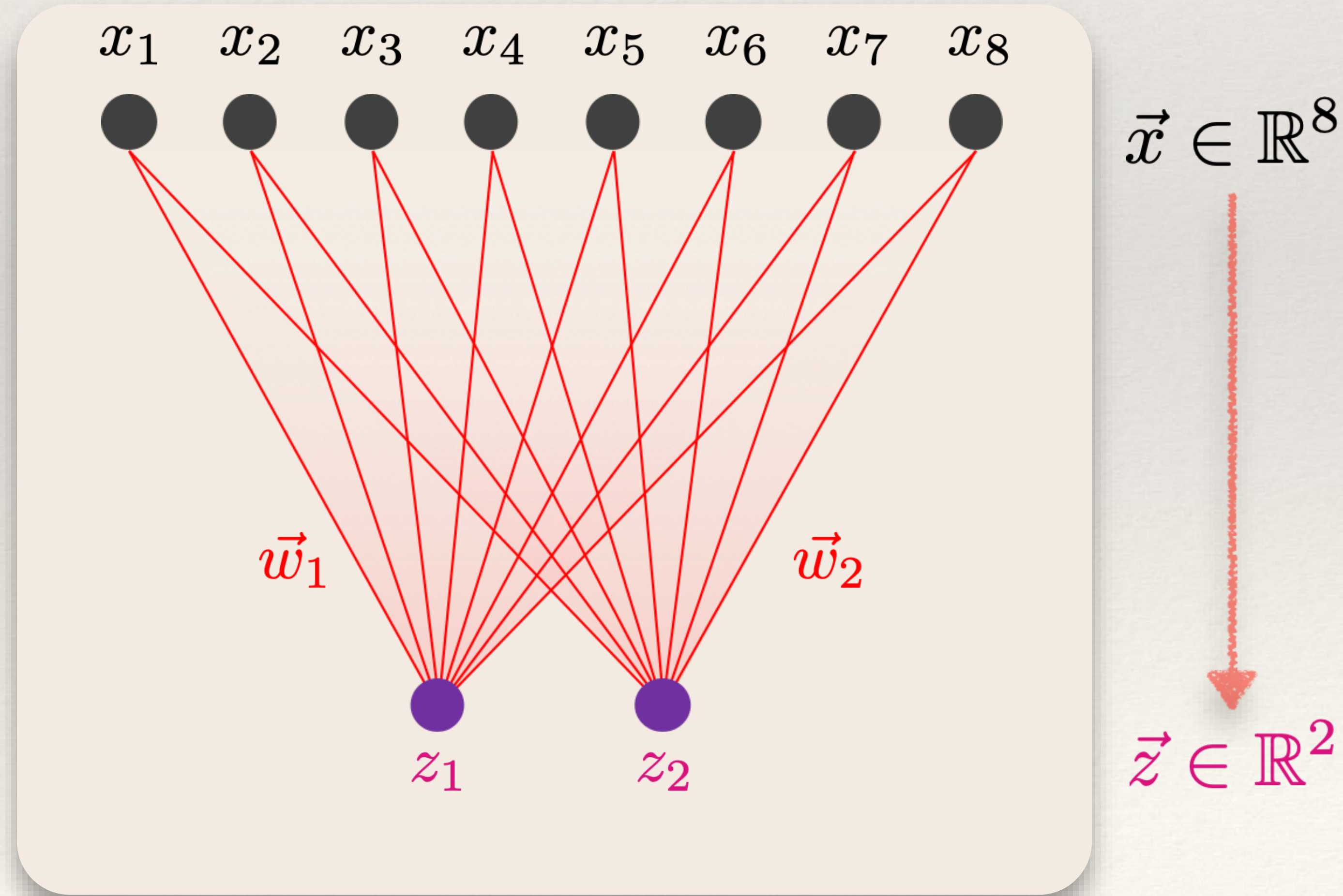
$z_1 = \vec{w}_1 \cdot \vec{x}$ $z_2 = \vec{w}_2 \cdot \vec{x}$

הורדת מימדים - PCA

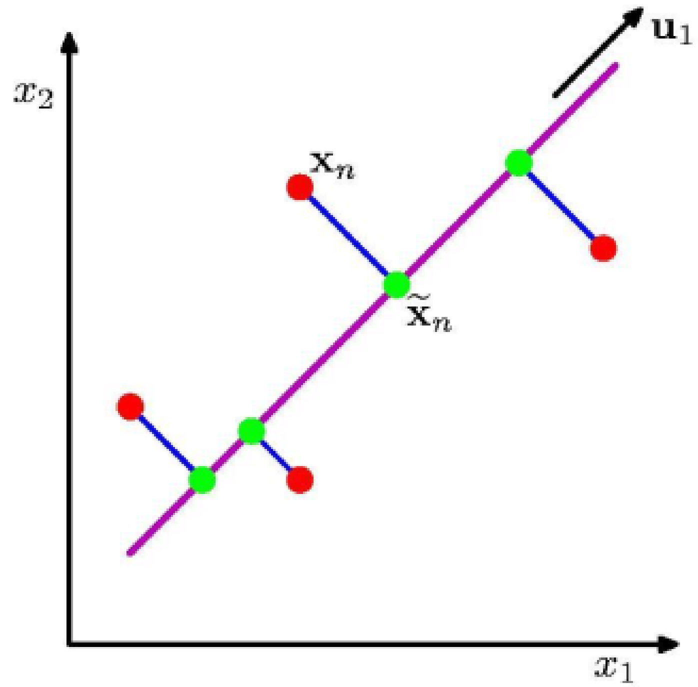
- מחפשים ליצג את $\vec{x} \in \mathbb{R}^d$ באמצעות $\vec{z} \in \mathbb{R}^k$
- ע״י שימוש בקומבינציות לינאריות $\vec{w}_1, \dots, \vec{w}_k$ של המאפיינים.

ש: איך נבחר את $\vec{w}_1, \dots, \vec{w}_k$?

ת: שגיאת שחזור מינימלית.



PCA: Motivation



PCA:

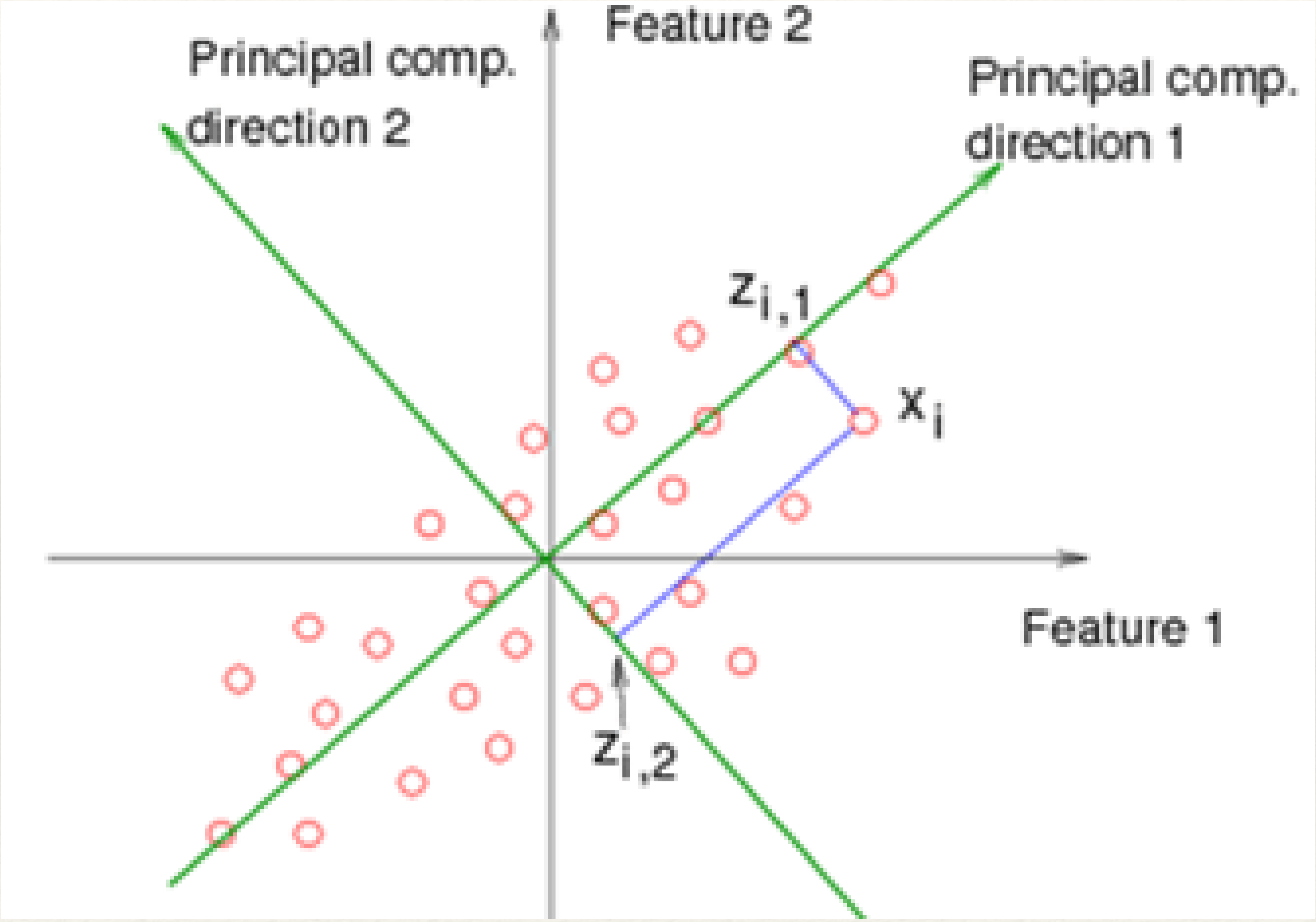
Orthogonal projection of the data onto a lower-dimension linear space that...

- ❑maximizes variance of projected data (purple line)
- ❑minimizes the mean squared distance between
  - data point and
  - projections (sum of blue lines)

PCA: Motivation

- Choose directions such that a total variance of data will be maximum
  - Maximize Total Variance
- Choose directions that are orthogonal
  - Minimize correlation
- Choose k<d orthogonal directions which maximize total variance

PCA: Motivation



PCA – פעולות מרכזיות

PCA does the following:

- finds orthonormal basis for data
- Sorts dimensions in order of “importance”
- Discard low significance dimensions

Explanations:

- Principal components – the W_i vectors
- Singular values – the coefficients of the principal components
  - higher coefficients mean more important principal components
- λ_i - eigenvalues – square of singular values

PCA - How to choose k ?

Principal components – the W_i vectors

Singular values – the coefficients of the principal components

- λ_i - eigenvalues – square of singular values

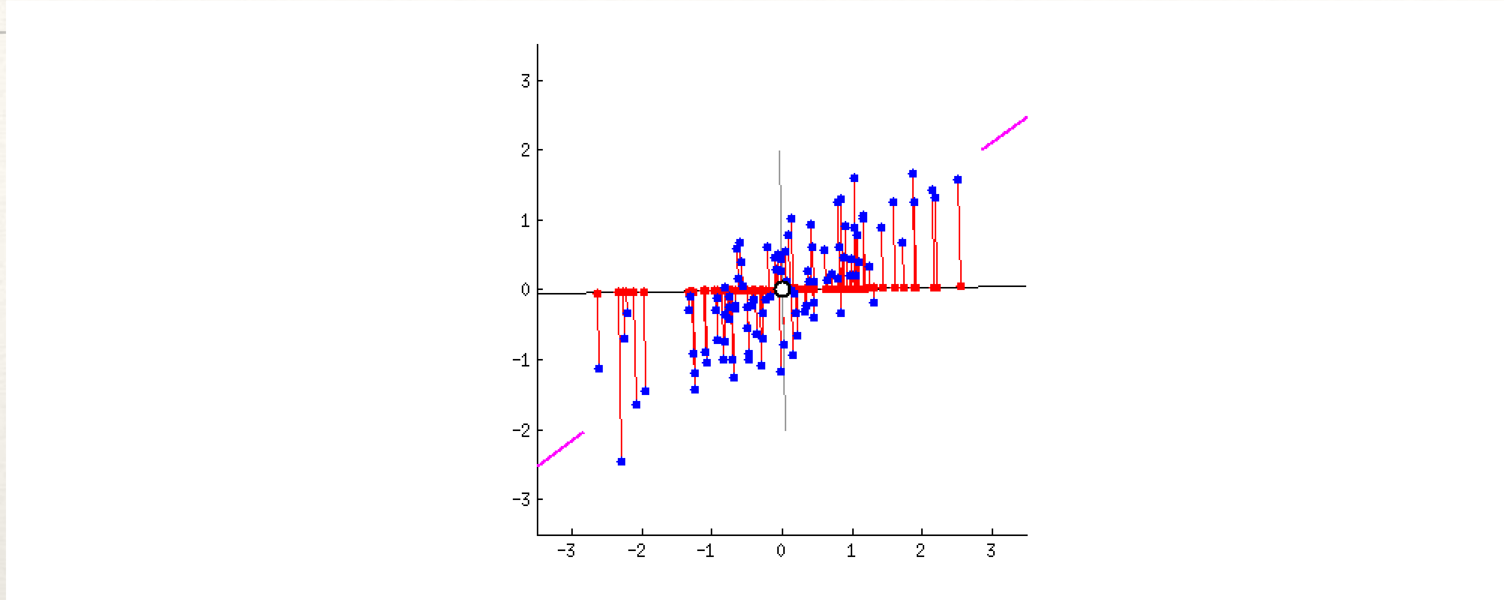
How do we choose k?

Use the following proportion: $$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when λ_i are sorted in descending order

- Typically, stop when proportion>0.9
- K could be also predefined

Using PCA



Notations

- Reduced dataset – Z
- W – principal components
- $X^{scaled} = standartized\ original\ dataset$

PCA Flow

- Find principal components
- Sort principal components, by the singular values/eigen values
- Select the most significant principal components

Transfer dataset in the following way:

- $Z = W^T * {X^{scaled}}^T$

PCA – pros and cons

Pros

- Reflect intuition of the data
- Dramatic reduce in size of data
  - Improve performance, reduce overfitting
- Interested in resulting uncorrelated variables which explain large portion of **total** sample variance
- Sometimes interested in explained shared variance (common factors) that affect data

Cons

- **PCA** is limited to linear dimensionality reduction
- Doesn’t know class labels
- PCA Does not try to explain noise
  - Large noise can become new dimension/largest PC
- Too expensive for some applications
- In cases of sparse data, there are better ways to deal with the dimensionality

PCA vs Feature selection

- Feature selection
  - Supervised: drop features which don't introduce large errors (validation set)
  - Unsupervised: keep only uncorrelated features (drop features that don't add much information)
- Dimensionality Reduction
  - PCA - Linearly combine feature into smaller set of features
  - PCA – Supervised data - explain most of the total variability