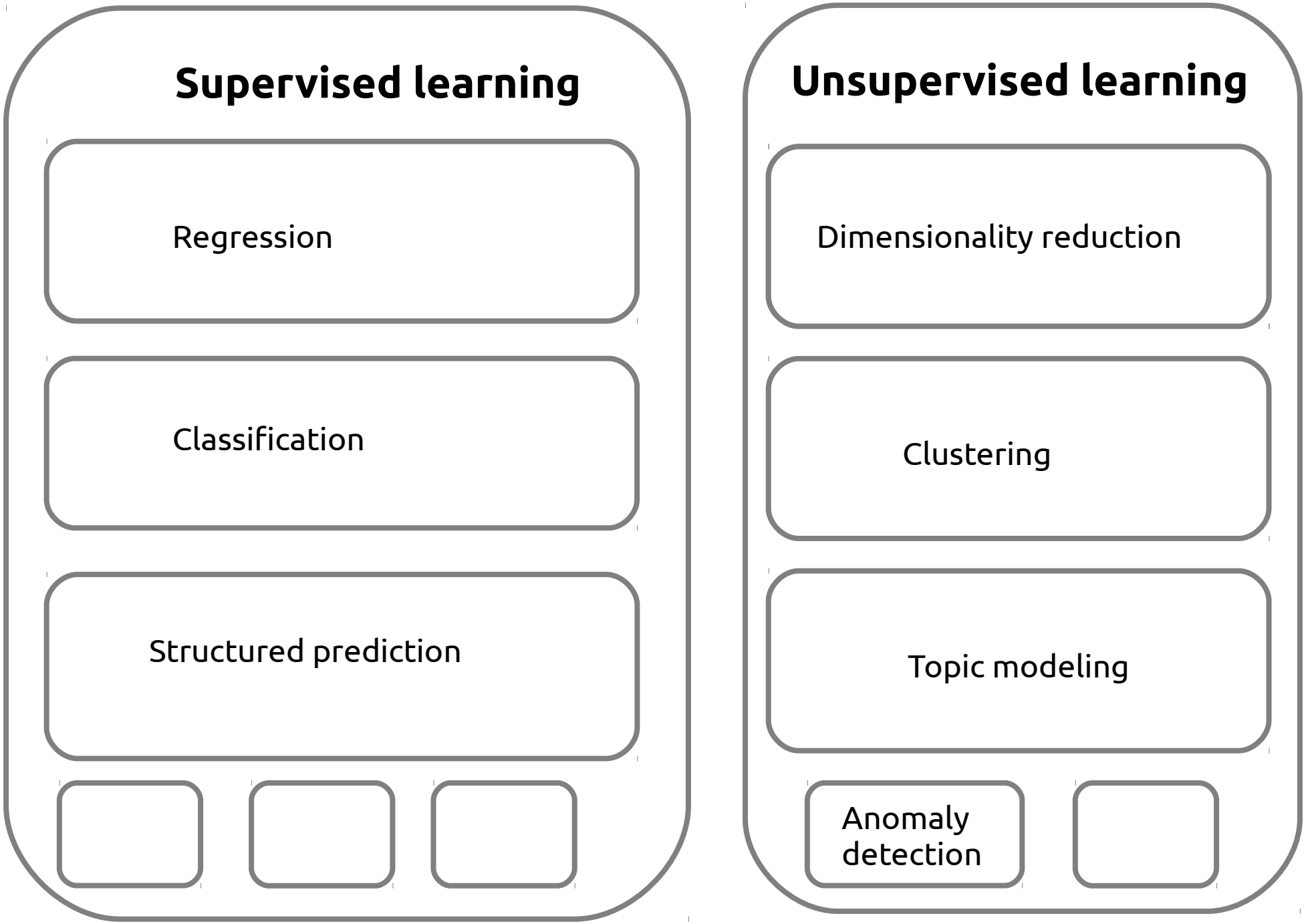
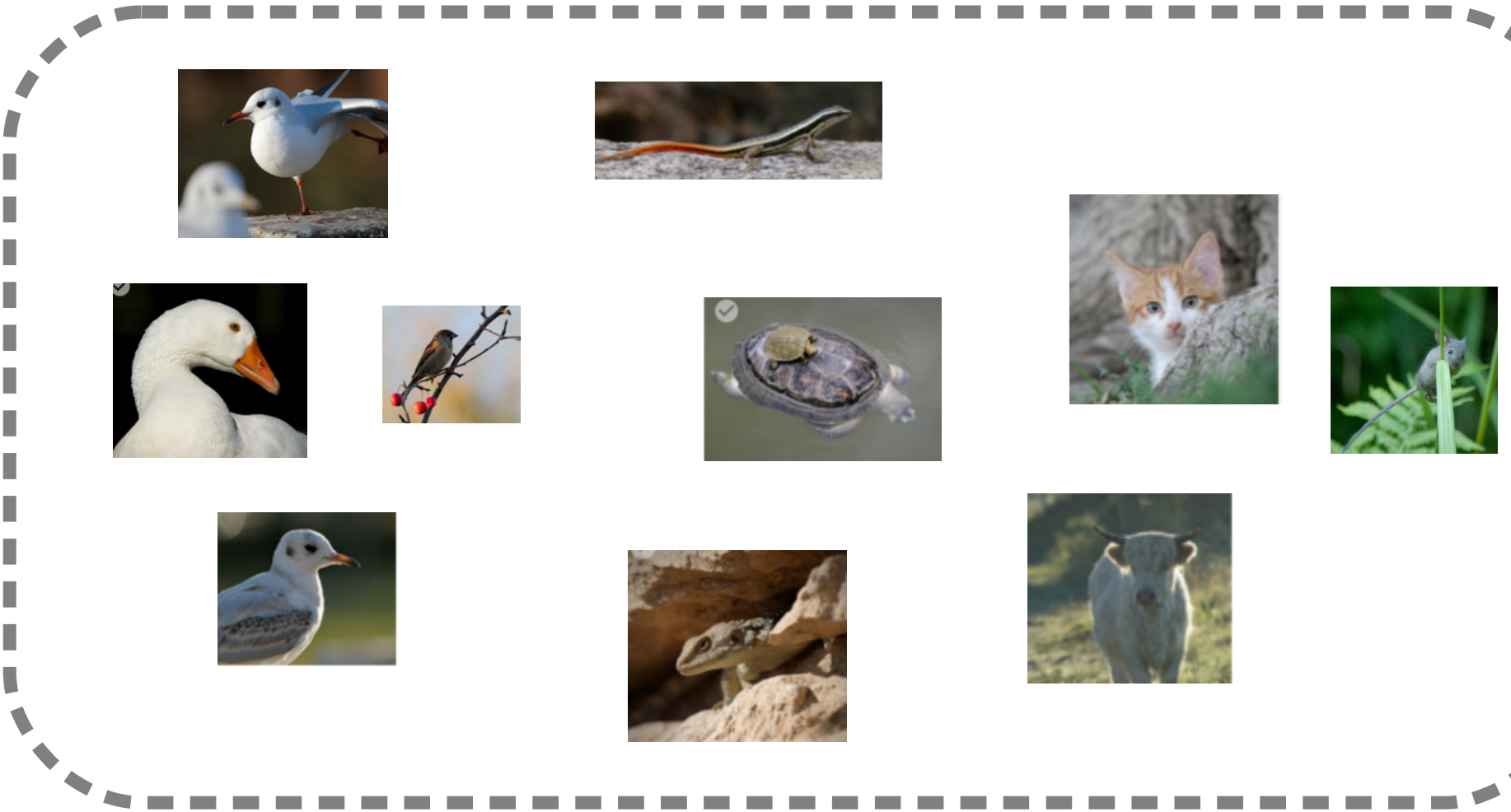
Dimensionality reduction

Grzegorz Chrupała
@gchrupala

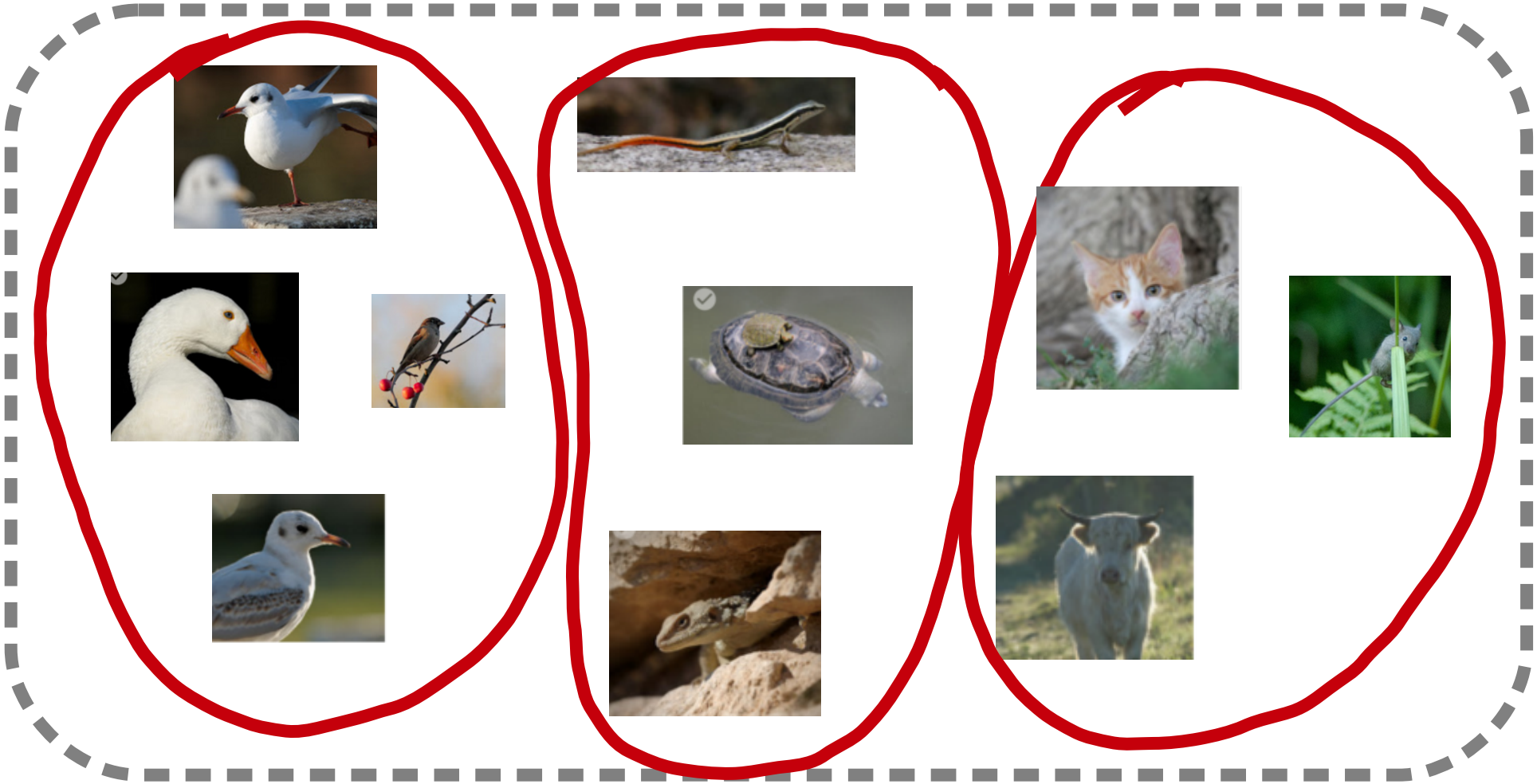

Supervised learning
Regression
Classification
Structured prediction
Unsupervised learning
Dimensionality reduction
Clustering
Topic modeling
Anomaly detection

- Do clustering and dimensionality reduction have anything in common?
- They are both useful for **simplifying** and *visualizing* data

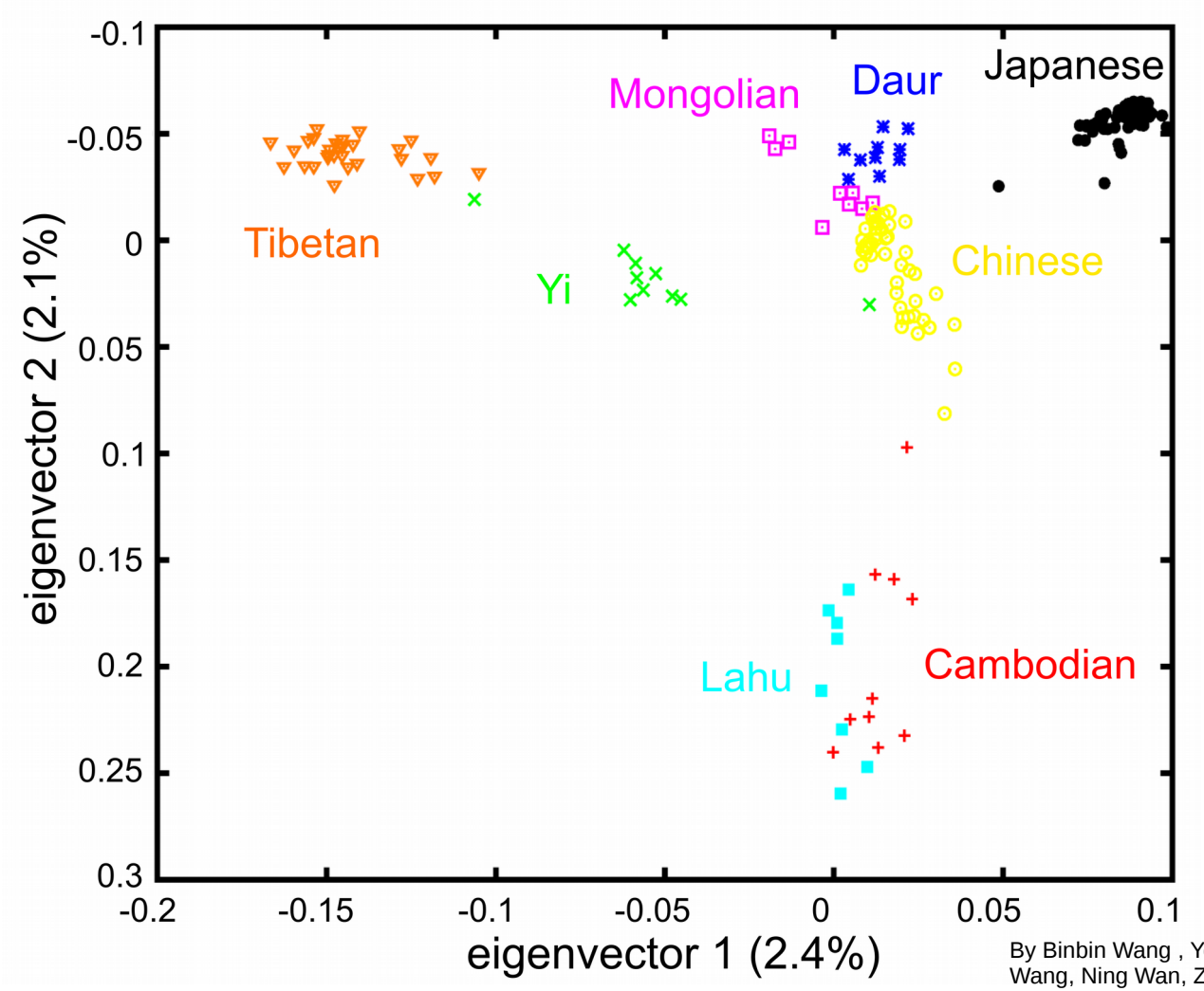
Clustering



Clustering



Dimensionality reduction



Rotate data in feature space

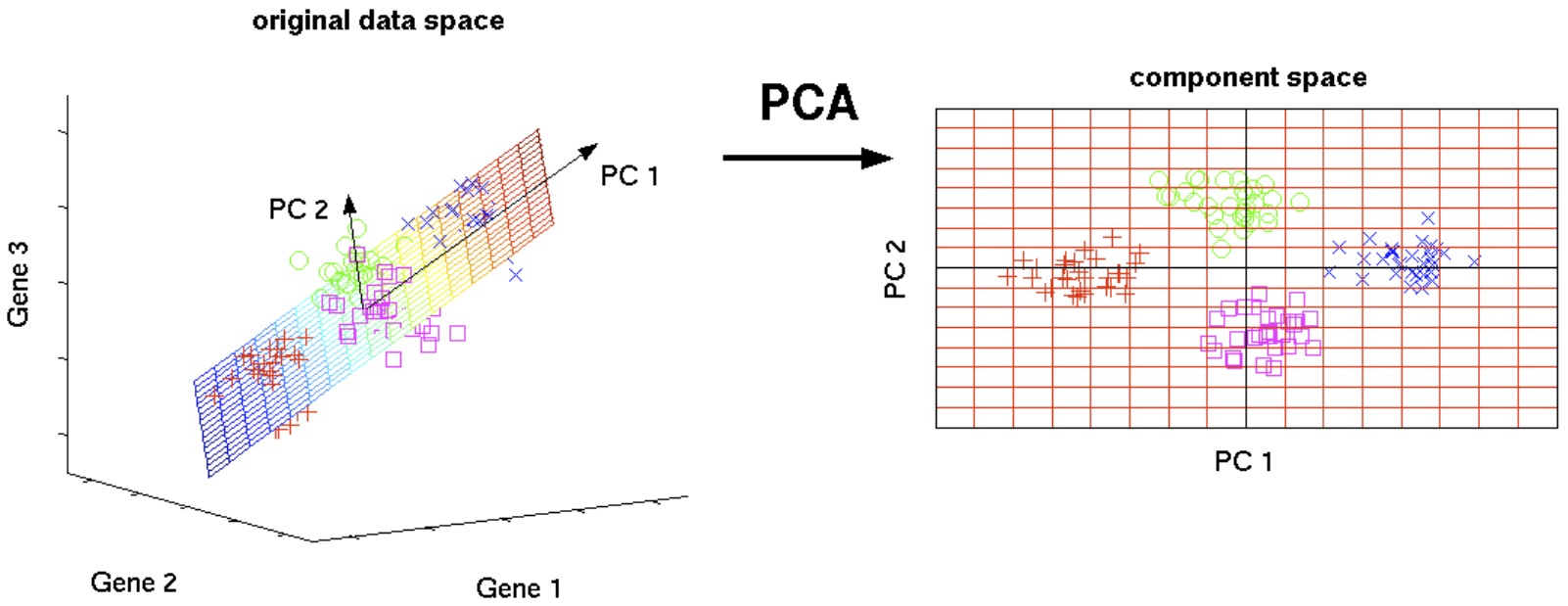
- Greatest variance → along 1st component
- Next greatest variance → along next component
- …

- Finally, keep only *N* components

Principal Component Analysis

- Simple technique of dimensionality reduction
- Technically, **orthogonal transform** (think rotation)
  - Orthogonal transformations preserve lengths of vectors and angles between them

Projection



http://phdthesis-bioinformatics-maxplanckinstitute-molecularplantphys.matthias-scholz.de/

- See interactive demo:

http://setosa.io/ev/principal-component-analysis/

Mathematically

- **X** – data matrix with *K* examples and *J* features
- **W** – *J* x *N* matrix which projects X to *N*-dimensional space:

$$\mathrm{T} = \mathrm{XW}$$

- The *i*th column of **W** contains the weights for the *i*th component

Projection

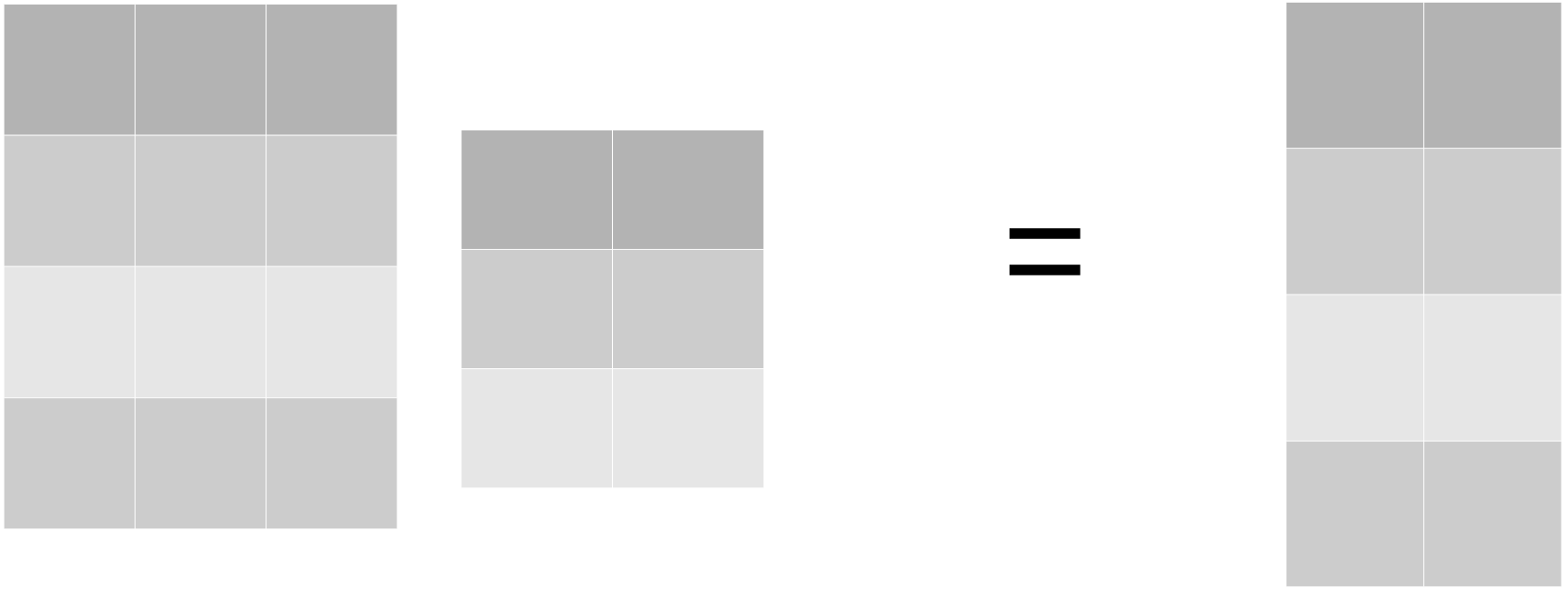
- In order to project example *k* from matrix **X**:

$$T_{k,i} = \sum_{j=1}^{J} X_{k,j} W_{j,i}$$

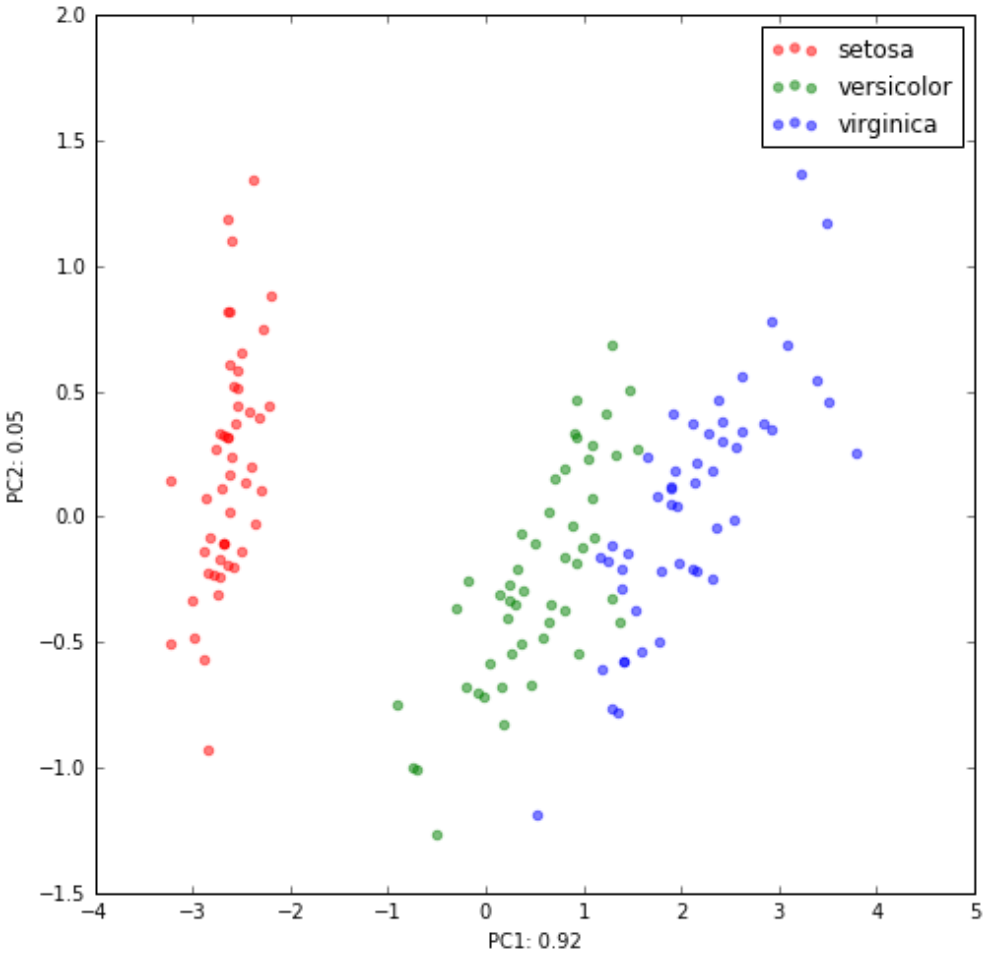
- Each component is a weighted sum of the original features

Example

X W = T



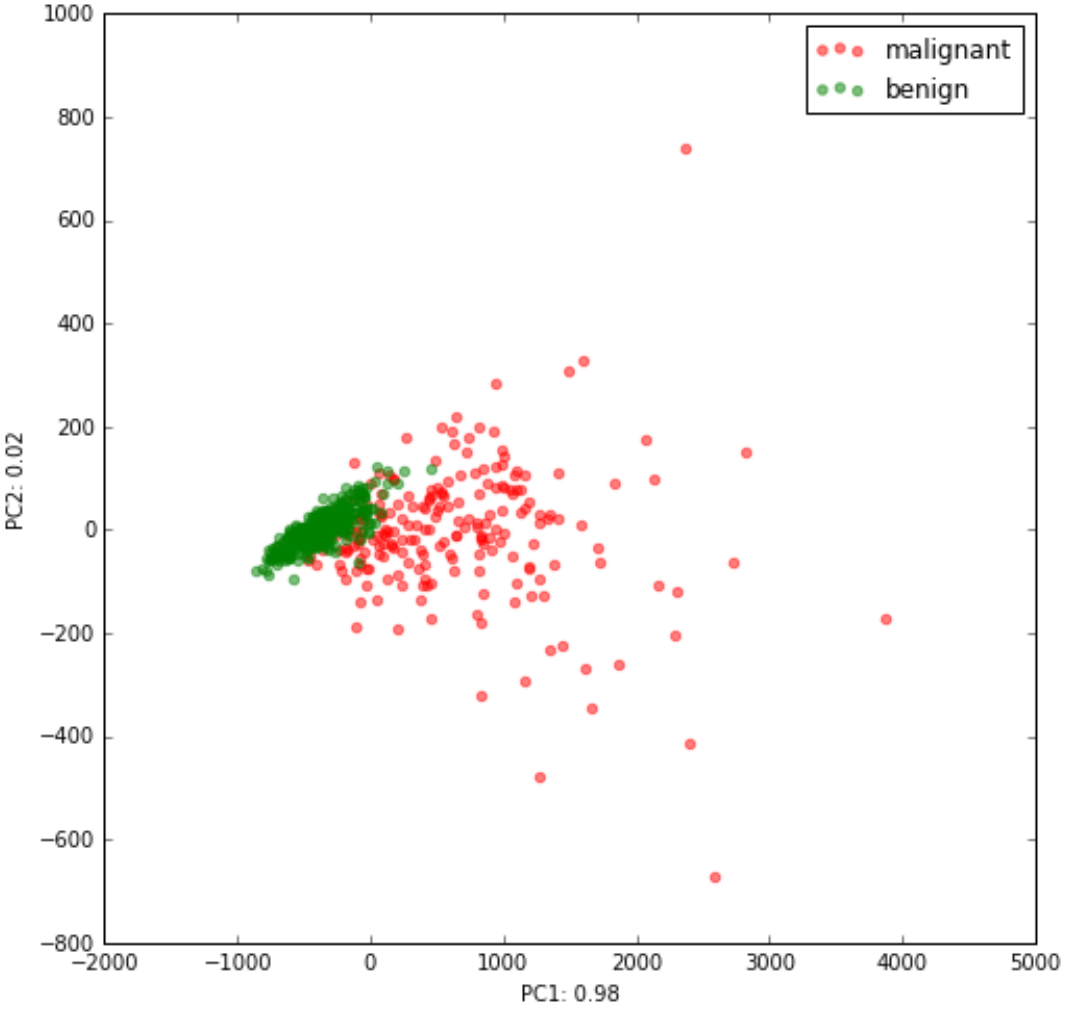
Iris dataset projected to 2D via PCA



Breast cancer diagnostic dataset

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

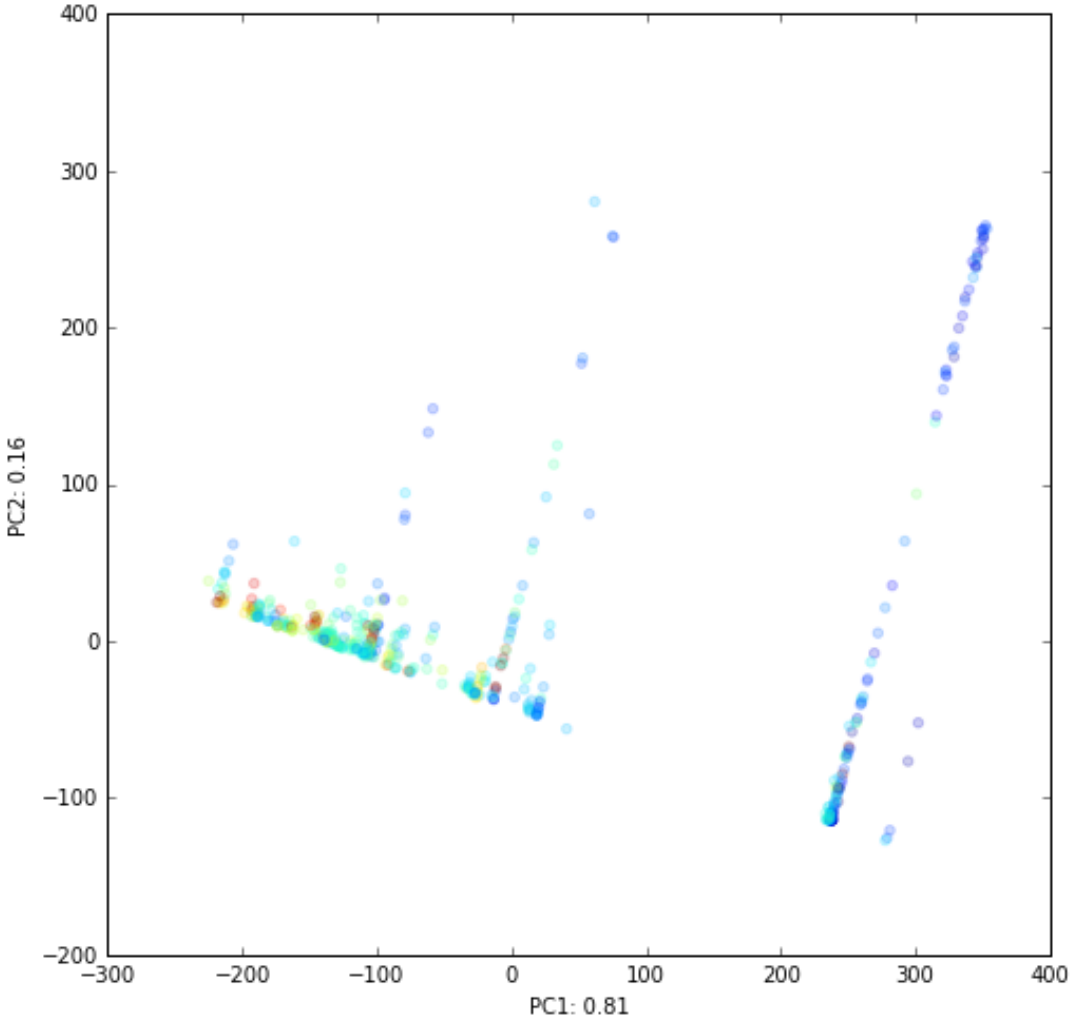
Projection of cancer dataset to 2D via PCA



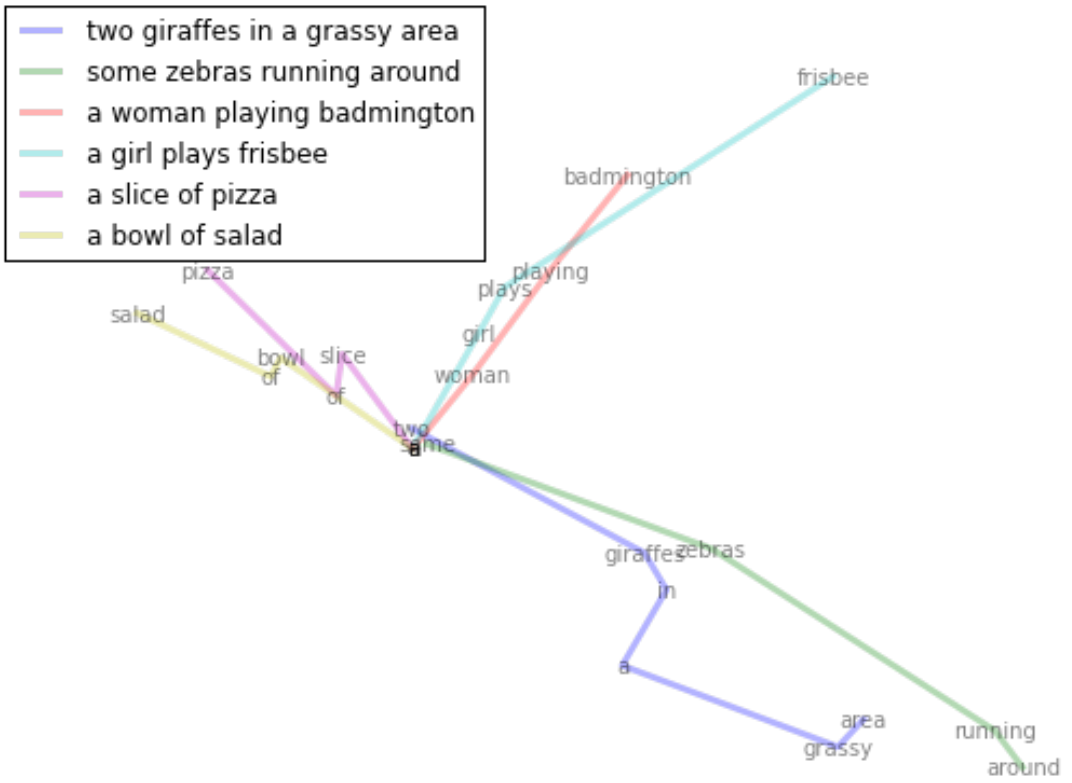
Boston house prices

- CRIM   per capita crime rate by town
- ZN   proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS   proportion of non-retail business acres per town
- CHAS   Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX   nitric oxides concentration (parts per 10 million)
- RM   average number of rooms per dwelling
- AGE   proportion of owner-occupied units built prior to 1940
- DIS   weighted distances to five Boston employment centres
- RAD   index of accessibility to radial highways
- TAX   full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B   1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT   % lower status of the population
- MEDV   Median value of owner-occupied homes in $1000's

Projection of Boston dataset to 2D via PCA



1024D sentence representations projected to 2D via PCA

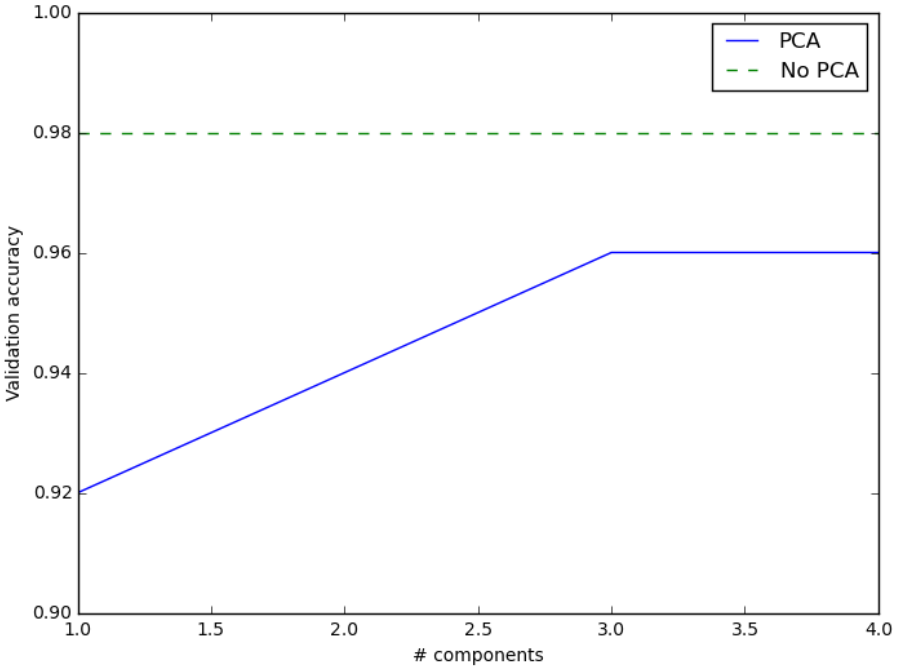


Beyond visualization

- What else can PCA be useful for?
- Reducing runtime of supervised learning algorithms
- Sometimes, improving accuracy of supervised learning

Classification accuracy on Iris via PCA

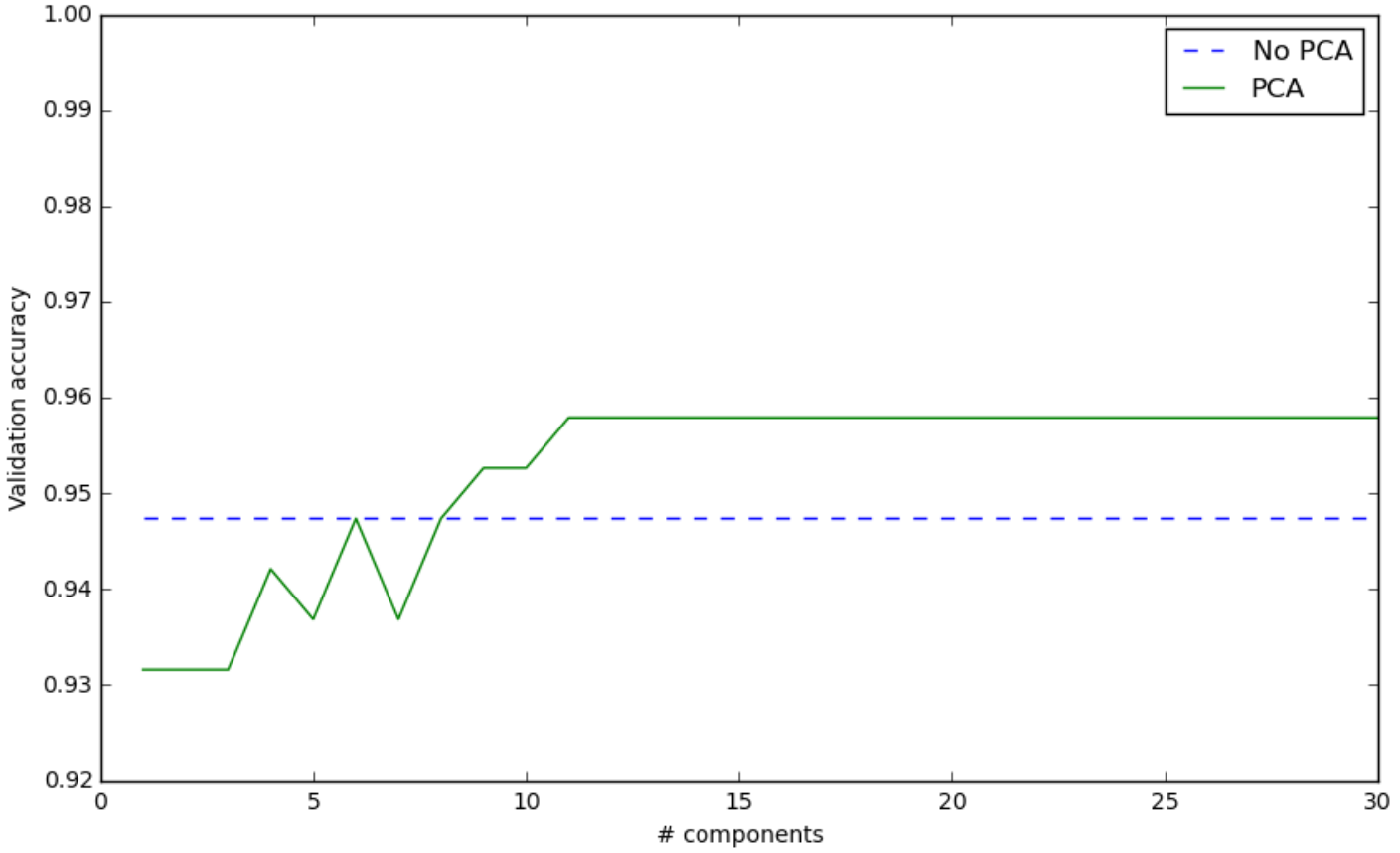
- Model with PCA doesn't reach the original model accuracy
- The last components does not improve



PCA for supervised learning

- Build **W** on training data
- Project training, validation and test data using **W**
- Fit classifier/regressor on training data, varying *N*
  - (possibly together with other hyperparams)
  - Choose best combination on validation data
- Test on test data

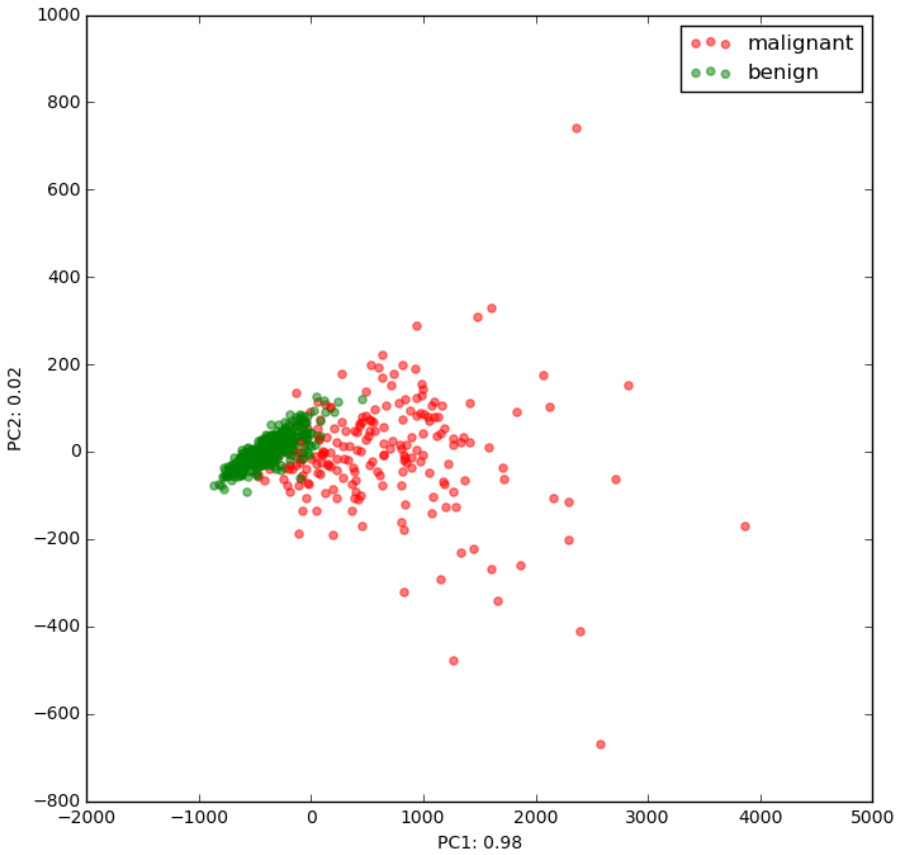
Classification of cancer dataset

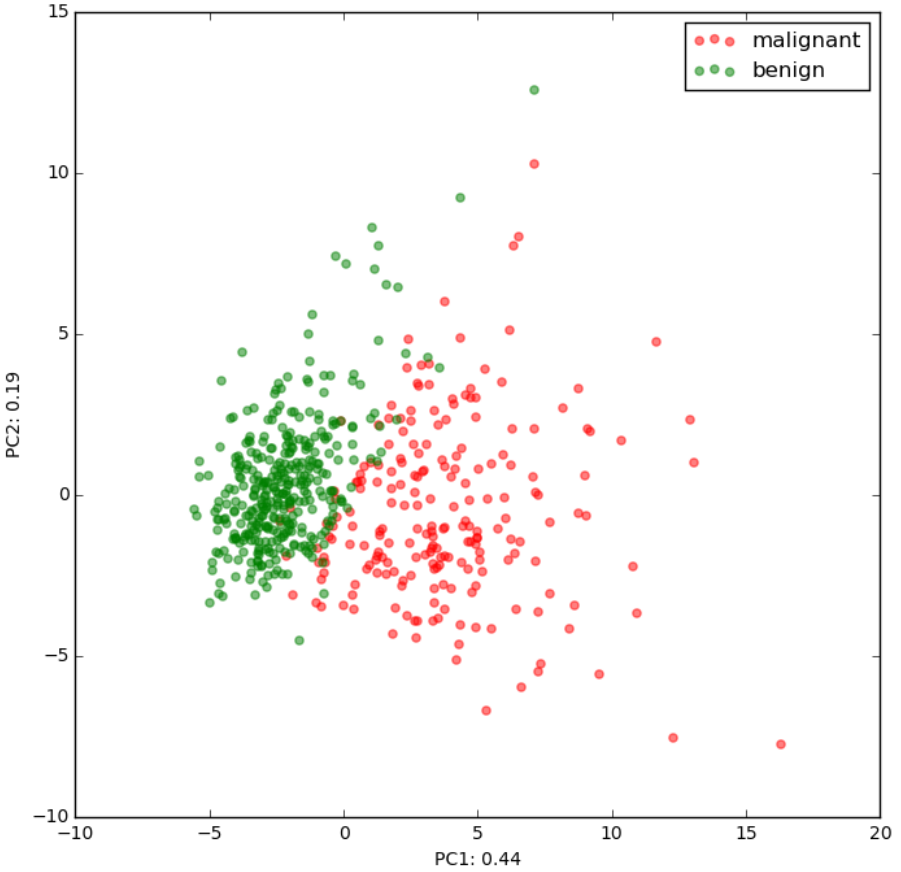


Scale of variables

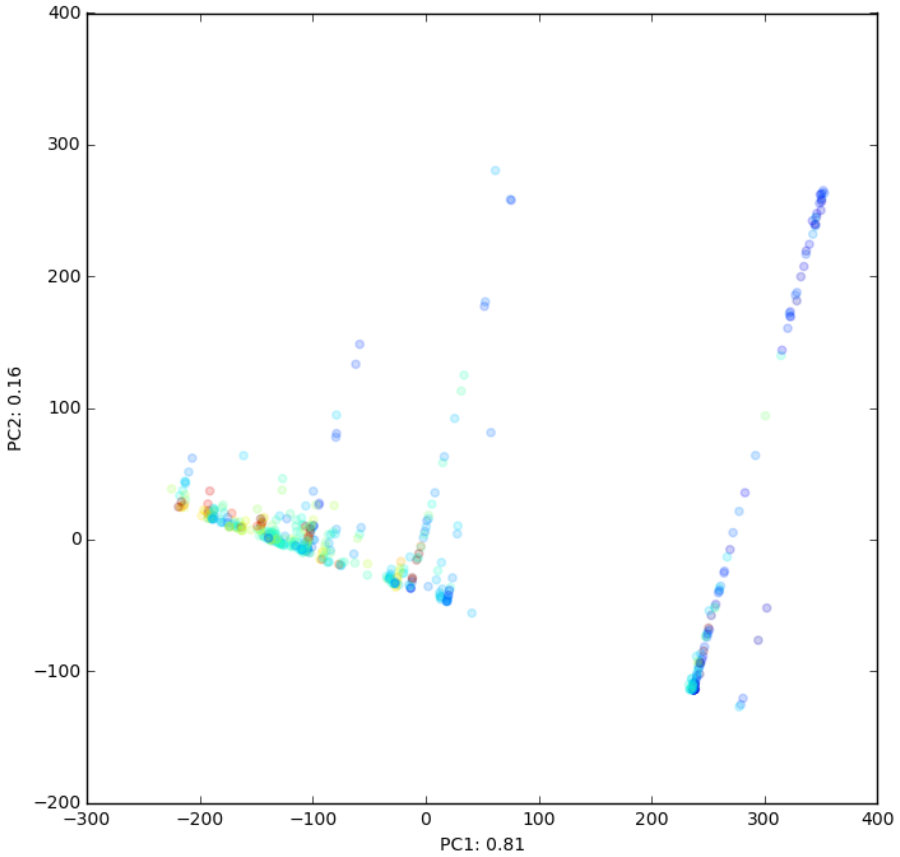
- Remember that magnitude of variance depends on scale/units of variable
- Therefore PCA weights are influenced by scale!
- Try standardizing data before PCA

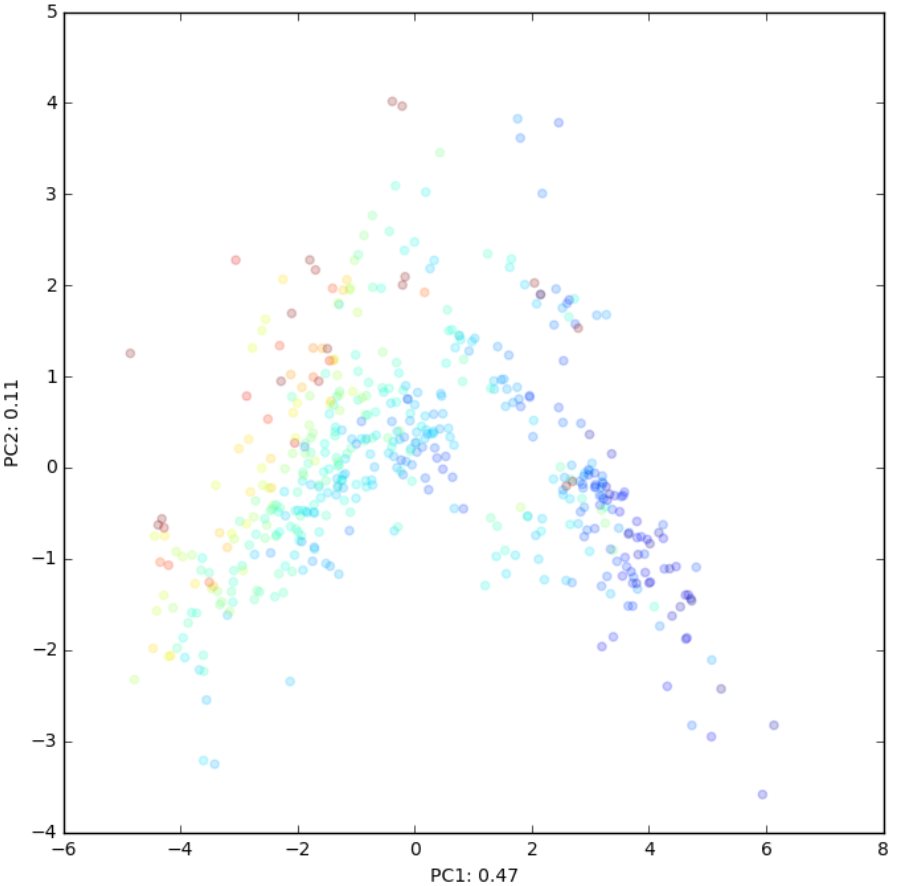
Cancer original vs z-scored



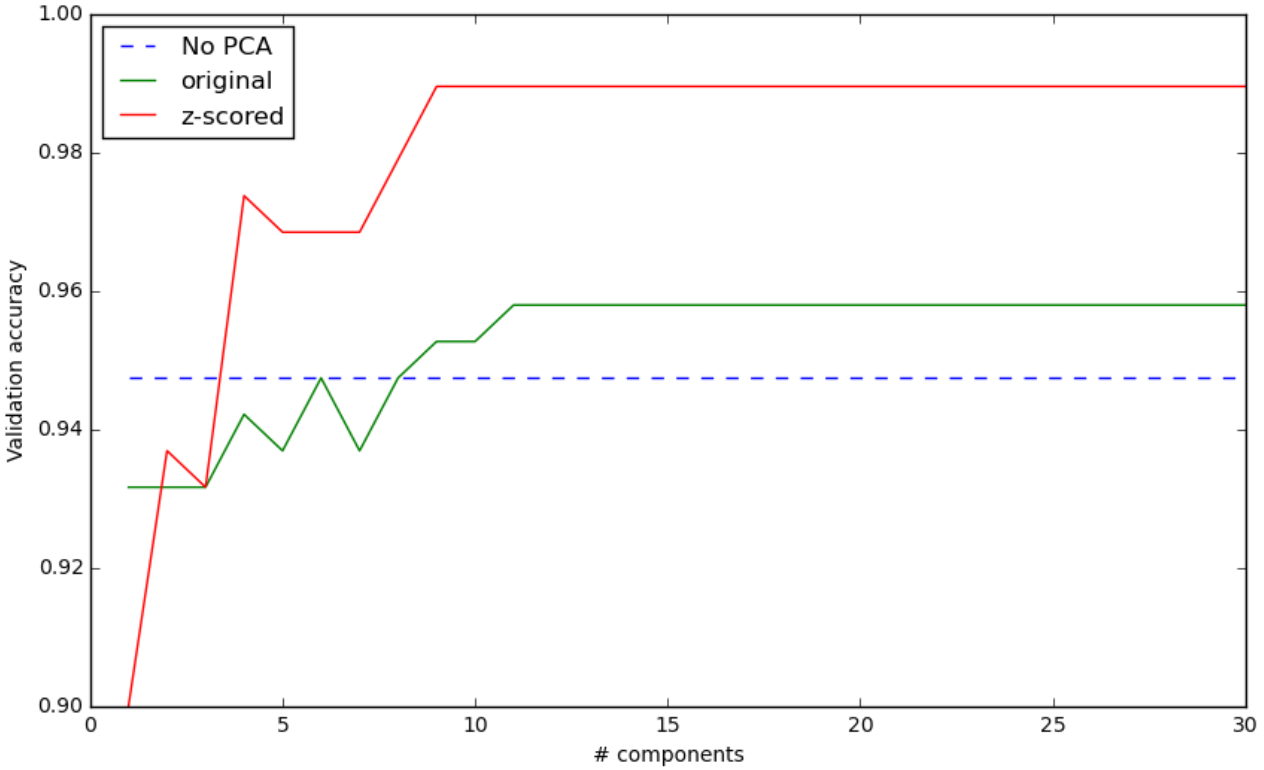


Boston
original vs z-scored





Cancer classification

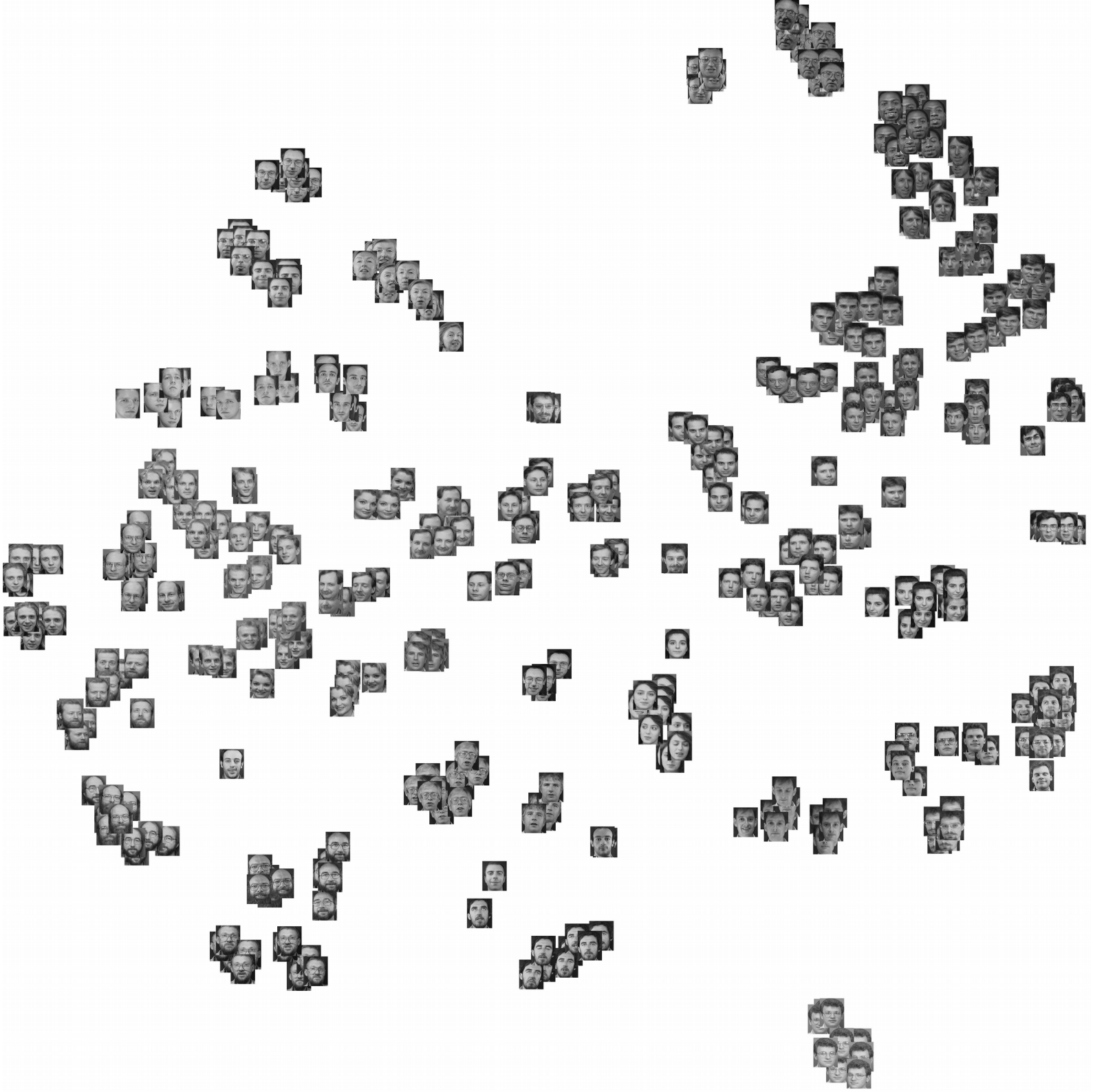


Dimensionality reduction beyond PCA

t-SNE

Tilburg alumn **Laurens van der Maaten**

- t-Distributed Stochastic Neighbor Embedding: http://lvdmaaten.github.io/tsne/
- *"T-SNE represents each object by a point in a two-dimensional scatter plot, and arranges the points in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points"*
- L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.



Many other techniques

See implementations in **scikit-learn**

- http://scikit-learn.org/stable/modules/manifold
- http://scikit-learn.org/stable/modules/decomposition