Dimension Reduction

Data Mining for Business and Governance*
19/09/2017

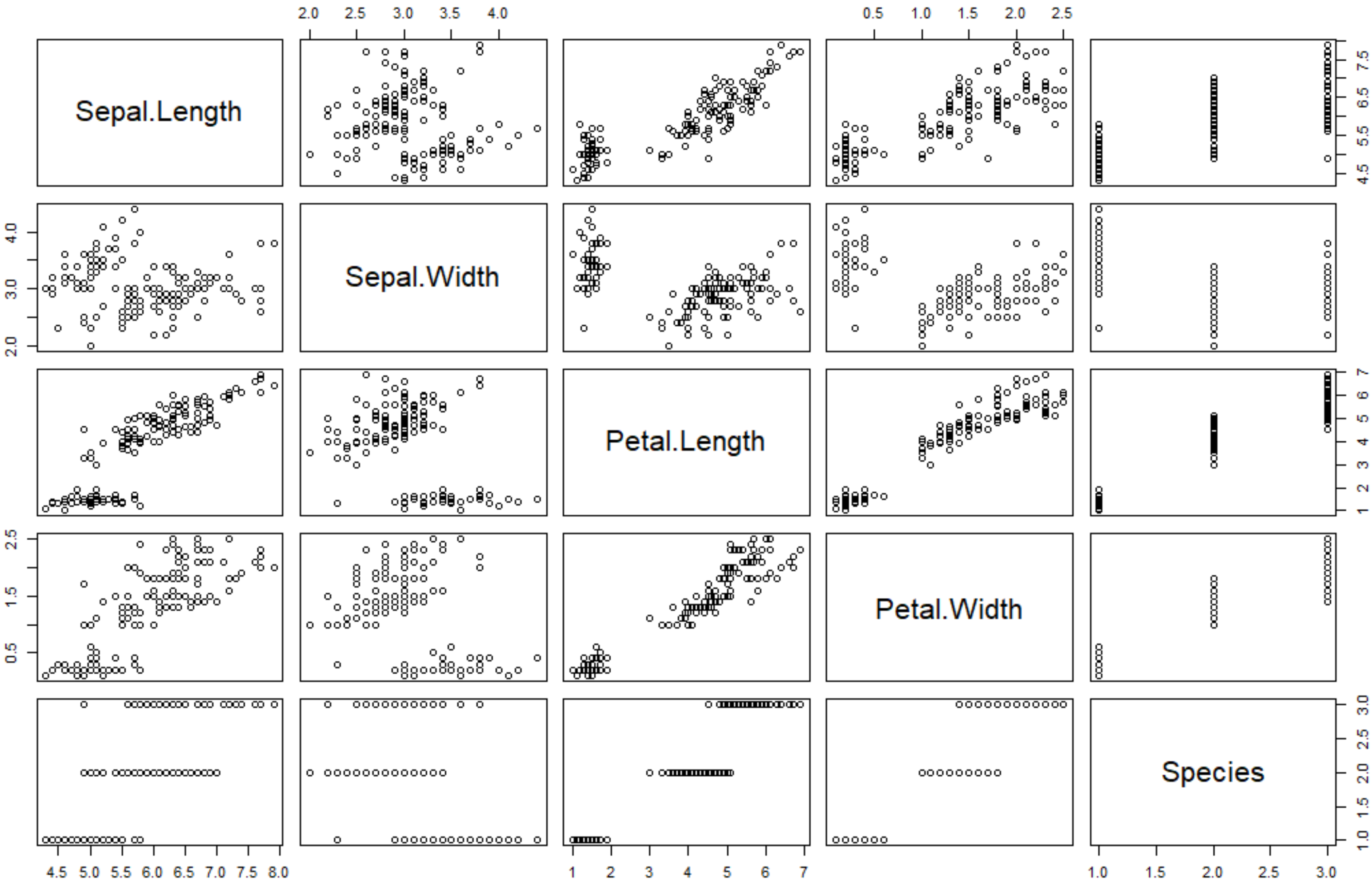
Tilburg University

*Formerly known as Social Data Mining

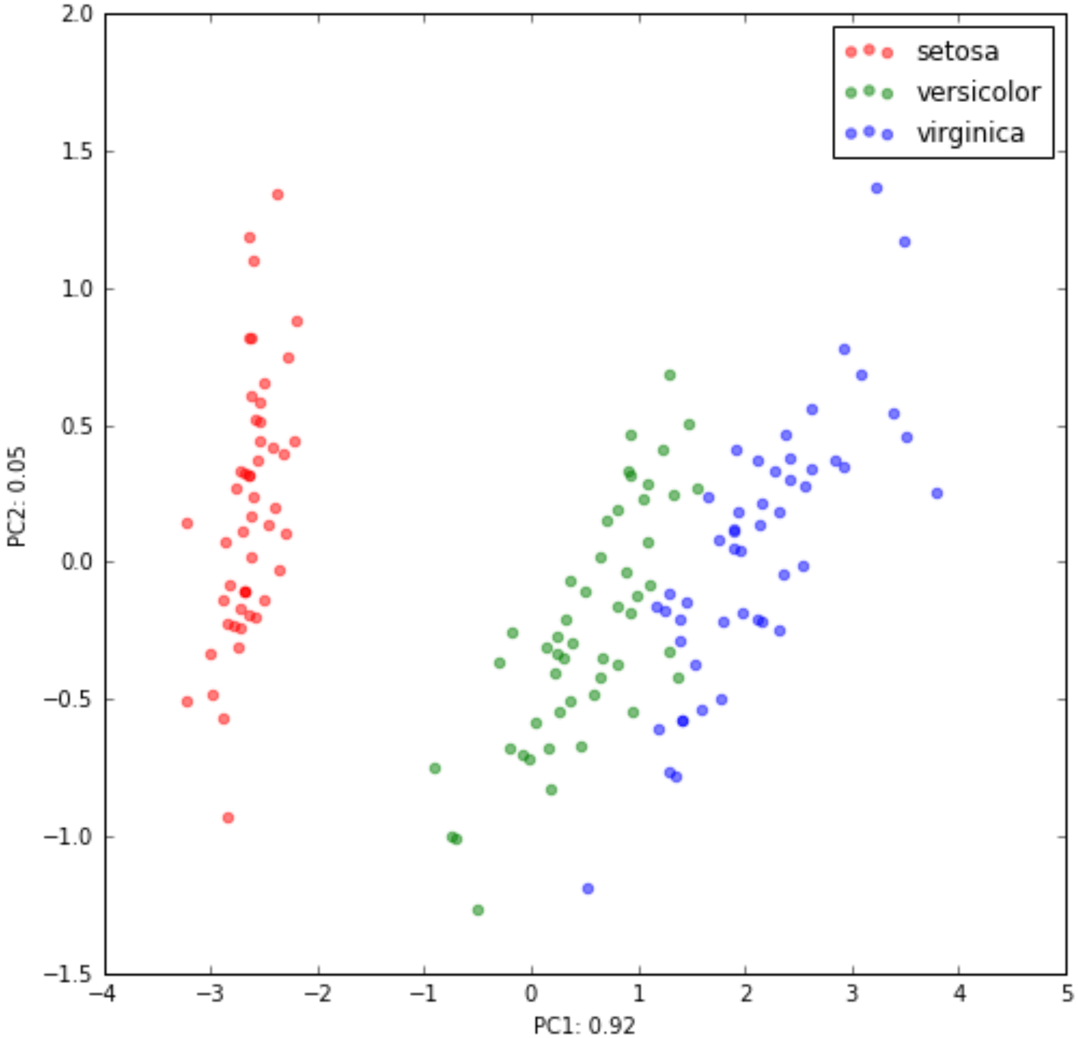
Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds

Visualization



Visualization

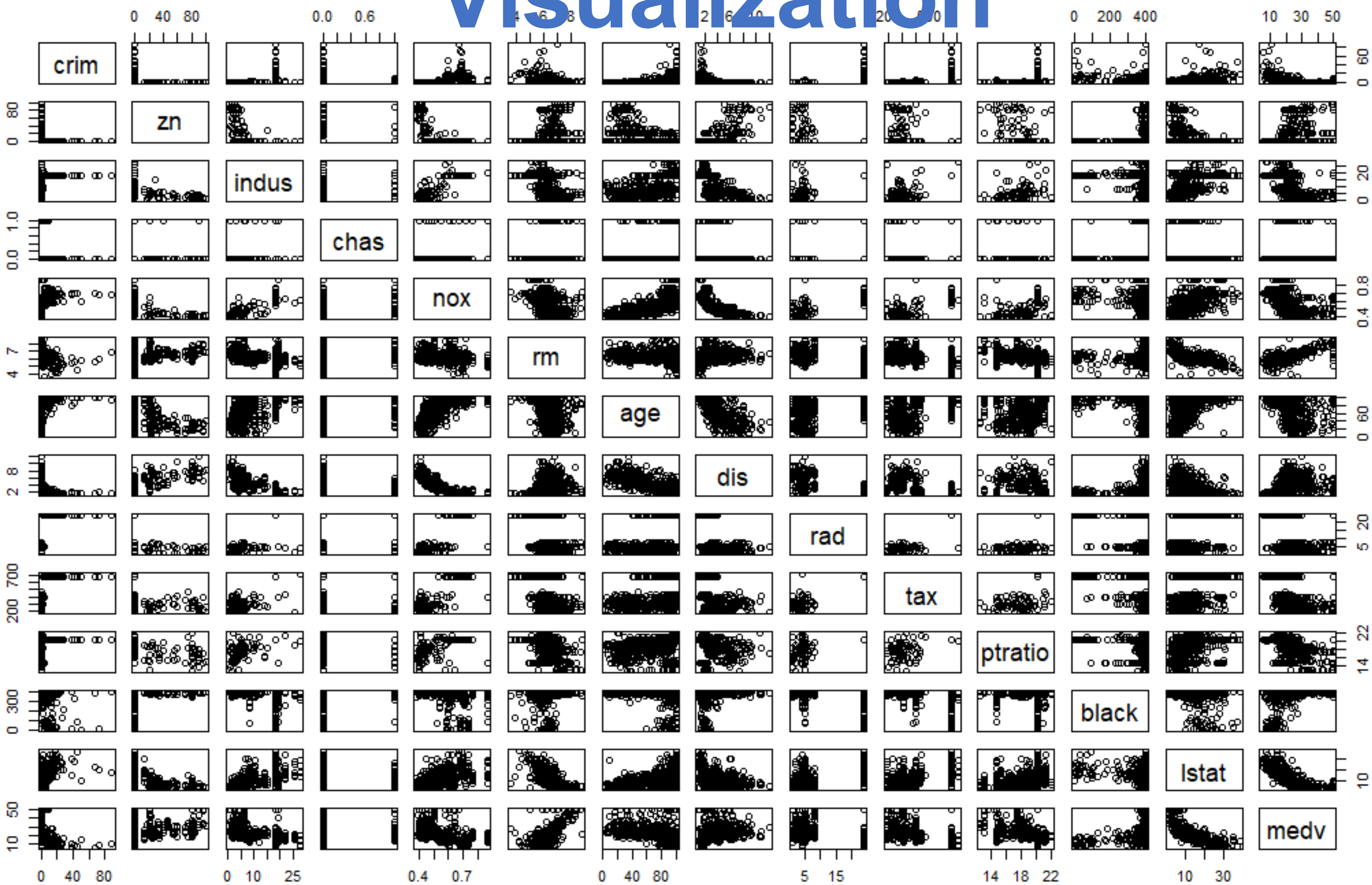


Visualization

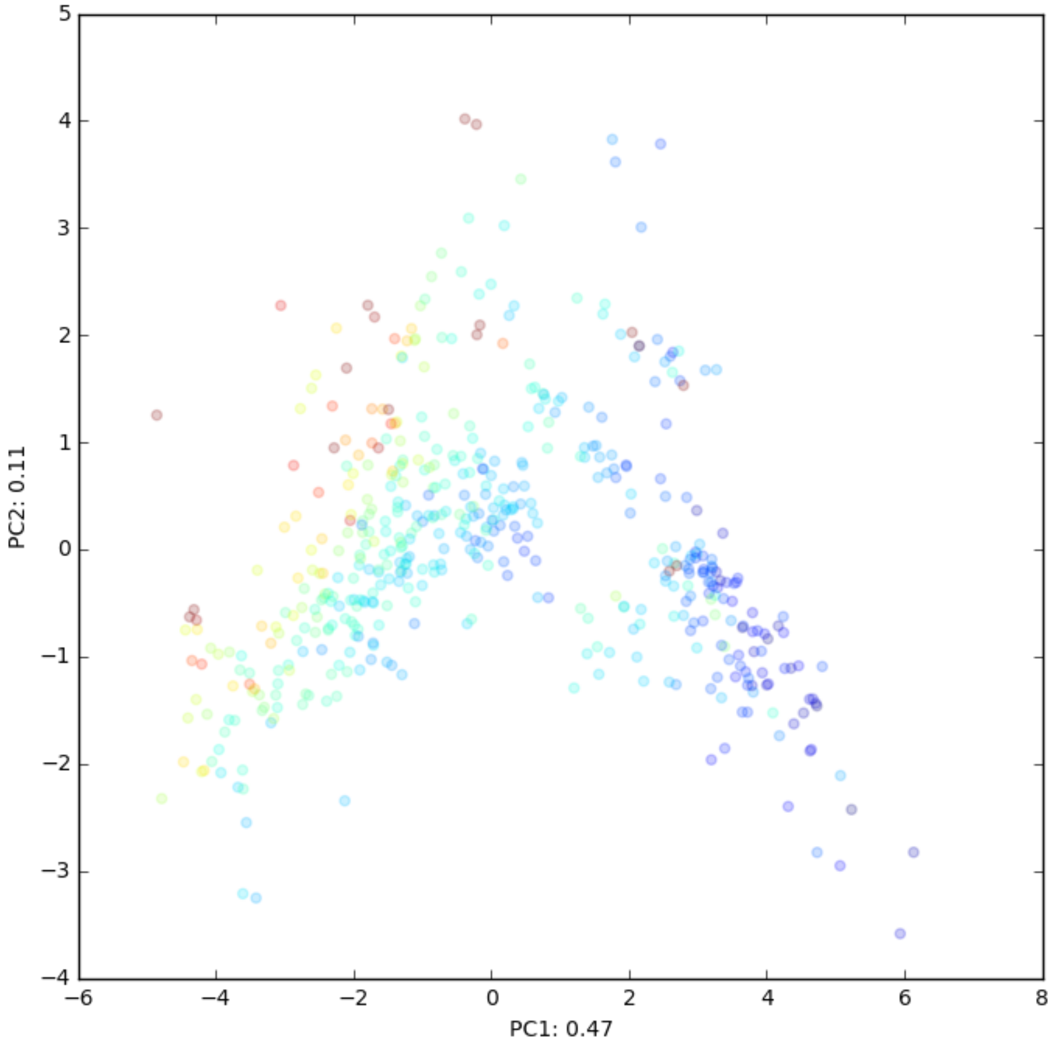
Dimensions of house prices in Boston:

```
- 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)
```

Visualization



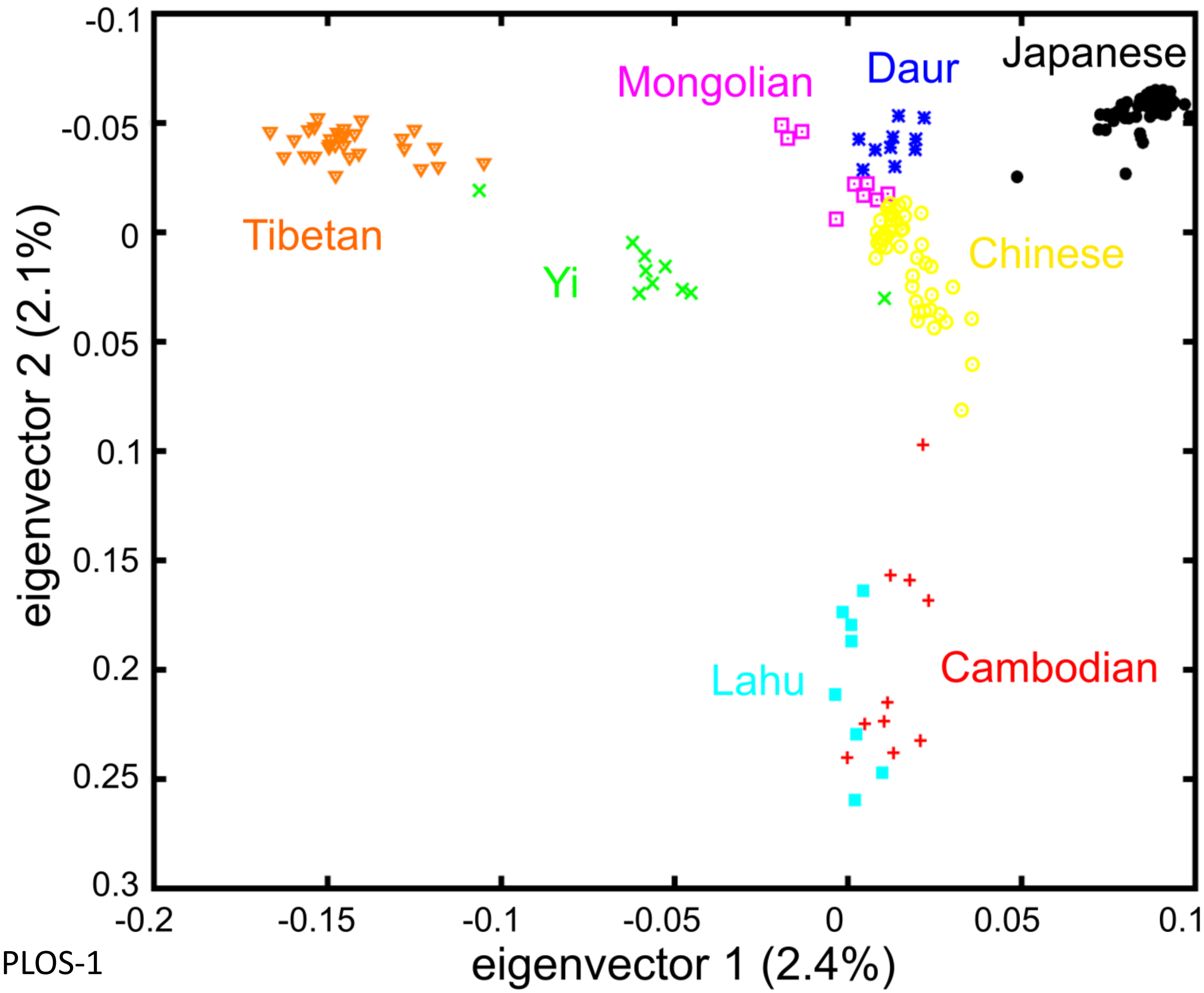
Visualization



Visualization

500,000+ genetic markers for eight of the largest East Asian populations (N = 192)
- Human Genome Diversity Project

Visualization



Wang, et al. 2011 PLOS-1

Why visualization?

- Our visual systems are incredibly good at detecting patterns
  - In a few dimensions (max 3)
  - Visualizing datasets is a fundamental aspect of good data science
- Often visualizing a dataset in a few dimensions can lead to insights
  - Or at least make a concept easier to convey to other people

Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds

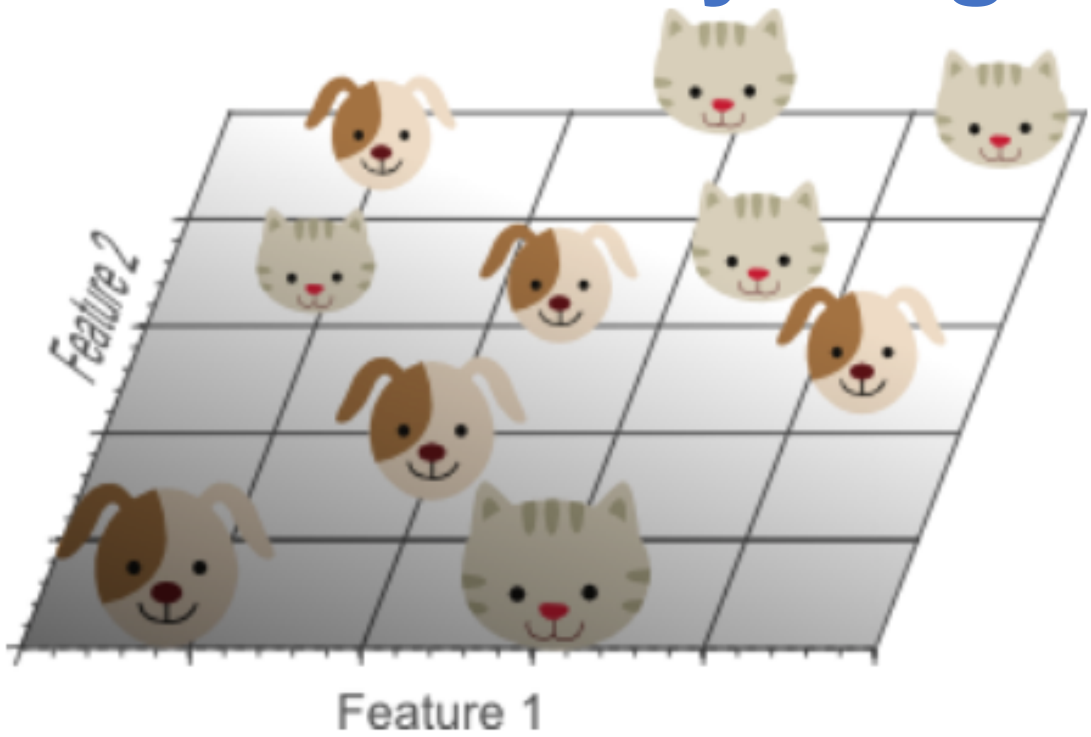
The curse of dimensionality

- First, let’s consider a simple linear decision boundary to solve the CAT-DOG classification task...

More features may be good

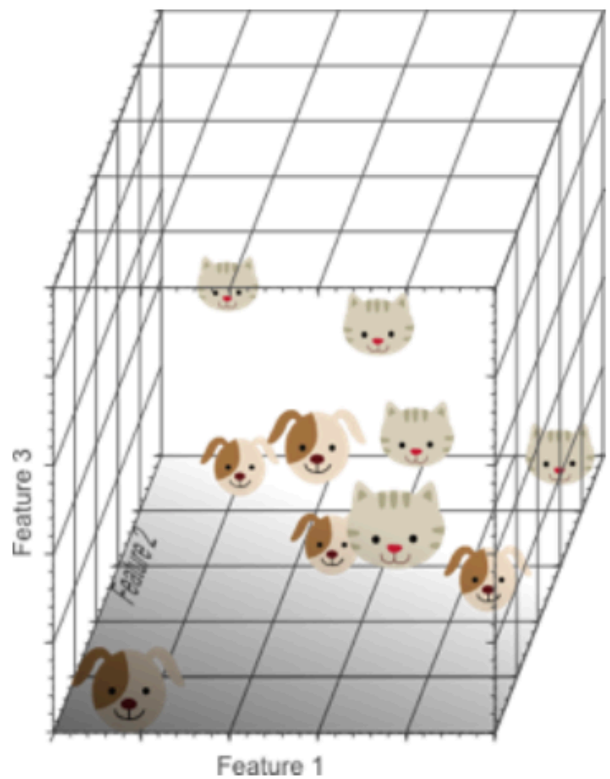
- 1 feature is insufficient to separate cats and dogs

More features may be good



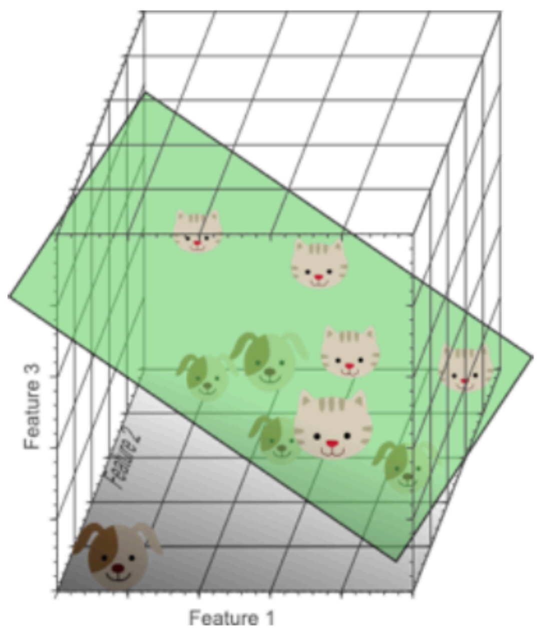
- 2 feature is insufficient to separate cats and dogs

More features may be good



- 3 feature is sufficient to separate cats and dogs

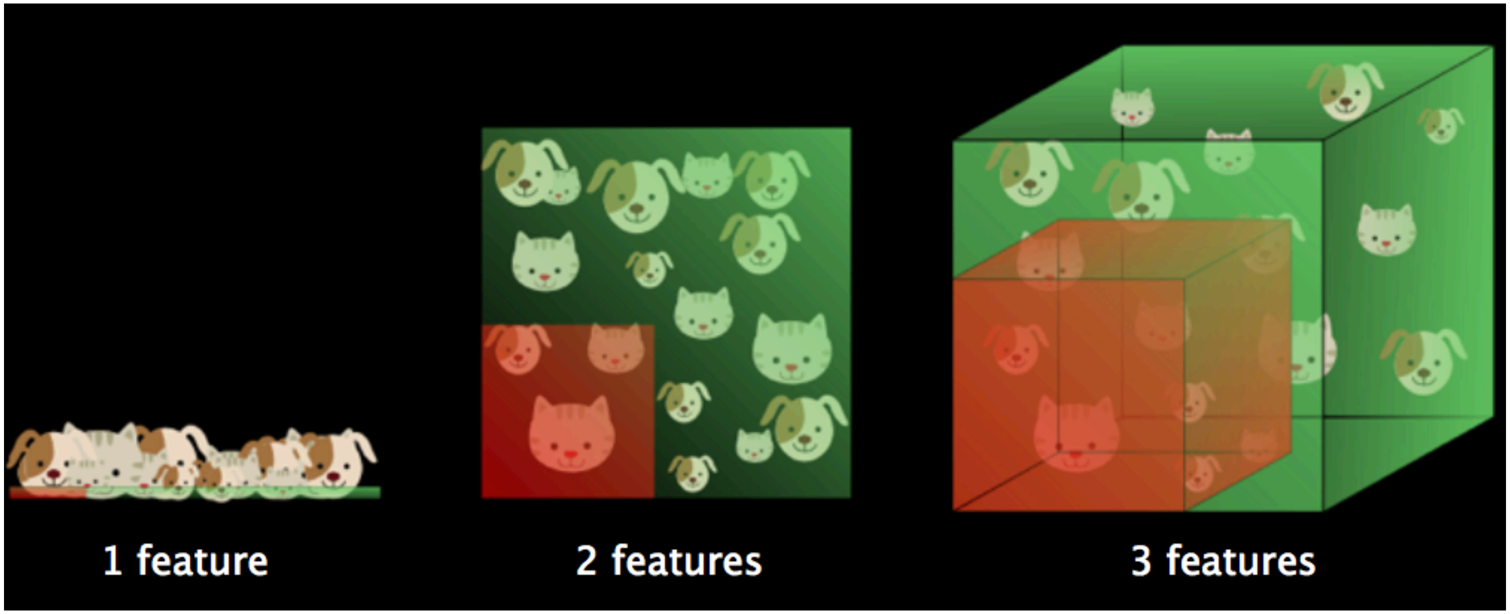
More features may be good



- 3 features are sufficient to separate cats and dogs

The cost of more dimensions

However, the amount of training data needed to obtain the same amount of cover grows exponentially with the number of dimensions



The cost of more dimensions

However, the amount of training data needed to obtain the same amount of cover grows exponentially with the number of dimensions

The math for this is terrible news for high dimensions:

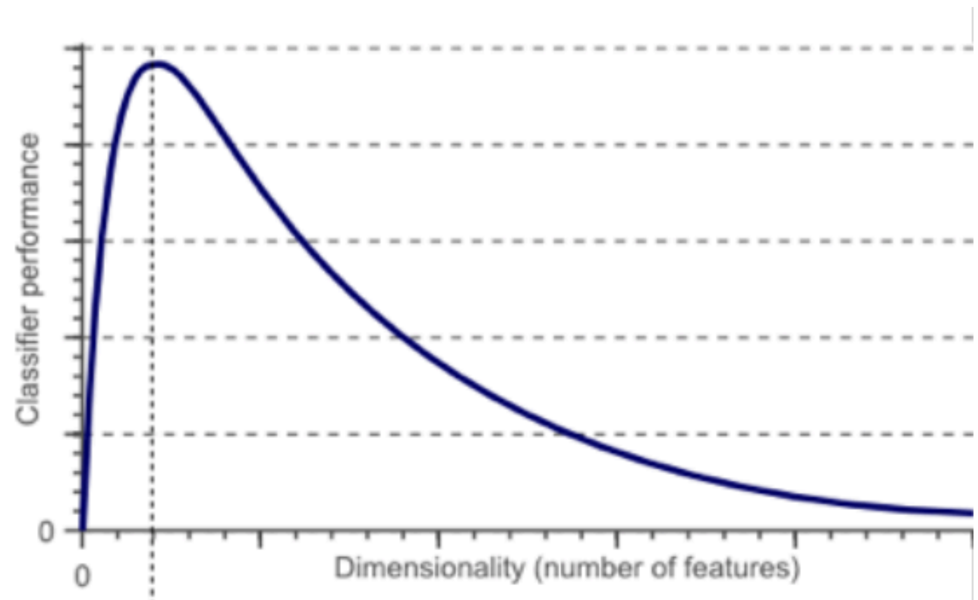
If we want N training items per unit of "feature space"
Then for each additional dimension we would need to
MULTIPLY the number of training items by N!

More features may be bad

- With more feature dimensions, each instance becomes more unique
  - Less similar to all other items

More features may be bad

- With more feature dimensions, each instance becomes more unique
  - Less similar to all other items
- For example, if we assume that the CAT-DOG classification task requires 20% of the data for training
  - As number of features becomes too large, generalization performance drops

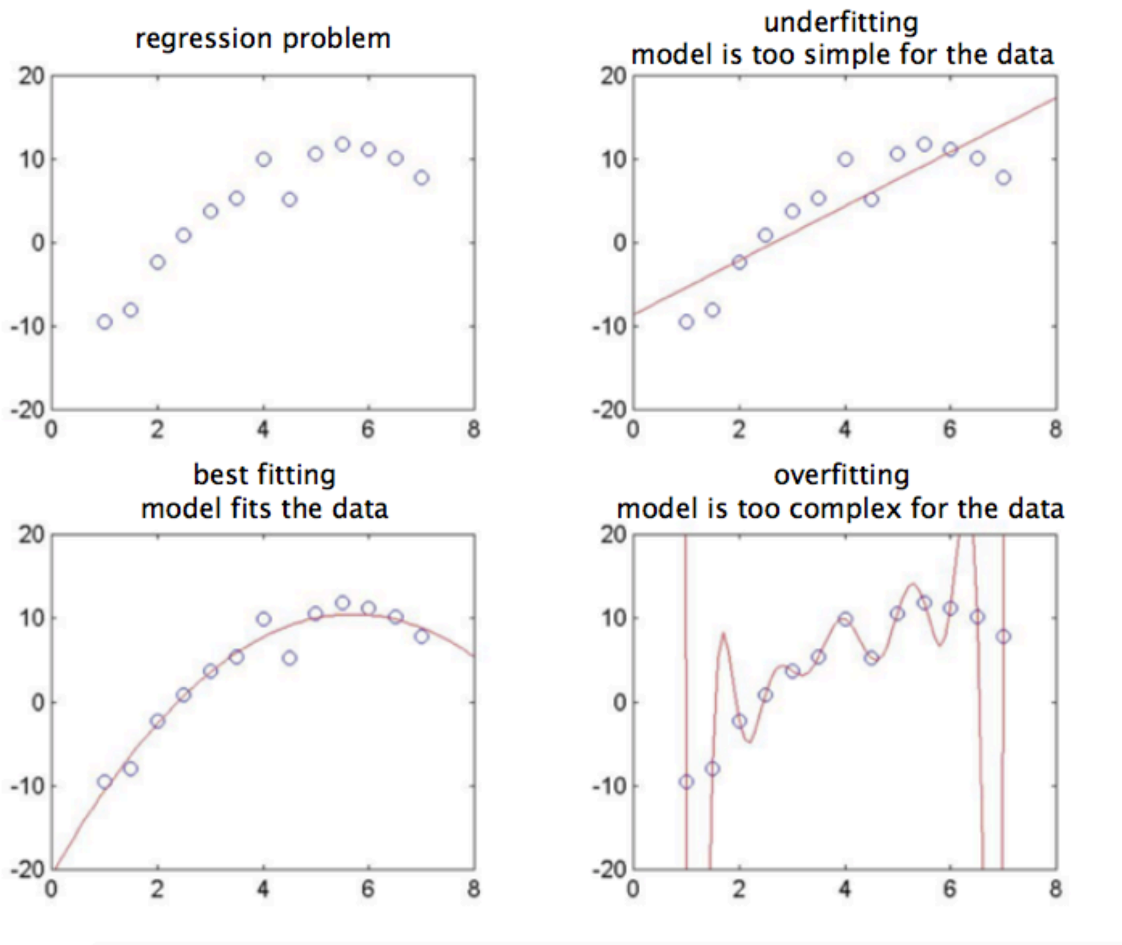


More features may be bad

- With more feature dimensions, each instance becomes more unique
  - Less similar to all other items
- Additionally, it becomes easier for models to OVERFIT the training data when the number of dimensions is higher

More features may be bad

- With mo ore unique
  - Less
- Addition training
  data whe



Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds

Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds

Feature Selection

The concept behind feature selection is simple:

- When you try to reduce the number of dimensions, don’t add additional dimensions
- Just remove some dimensions

(seriously, there are other options? Wait for it…)

Feature Selection

The concept behind feature selection is simple:

- When you try to reduce the number of dimensions, don't add additional dimensions
- Just remove some dimensions

(seriously, there are other options? Wait for it…)

All the action in Feature Selection is in how to decide which dimensions to remove

Feature Selection: The filtering strategy

- Perhaps the most basic method of eliminating features
  - It considers each feature dimension separately
  - It does not depend on the type of model you are using
- For each feature dimension, consider the relationship between it and the value to predict
- Based on some criteria, determine if that feature should be retained
- Example criteria: correlation, mutual information, or various significance tests
  - Criteria should be relatively fast to compute

Feature Selection: The wrapper strategy

- Treat the set of feature dimensions as a hyperparameter of the model
- Fit the model to training data using a subset of the possible features
- Evaluate the restricted model on a test set of data (this is Cross-Validation)
- Use the set of features that provide the best fit for this model

Feature Selection: The wrapper strategy

- Treat the set of feature dimensions as a hyperparameter of the model
  - Select, Train, Test
- Issues:
  - This process can be really slow if the model isn't fast
  - The set of possible feature dimensions to include could be really, really large
    - With N dimensions, the number of possible sets are 2^N
  - The set of features is completely dependent on the model

Feature Selection: The embedded strategy

- Embed the filtering strategy within the wrapper strategy

Feature Selection: The embedded strategy

- Embed the filtering strategy within the wrapper strategy
- These are special cases of wrapper strategies where part of fitting the model involves filtering out some feature dimensions
  - Canonical example is LASSO regression where as part of the iterative estimation of the parameter weights (a wrapper strategy), the regression weight for many features is set to 0 (a filtering strategy)

Feature Selection: The embedded strategy

- Benefits:
  - Possibly faster than a pure wrapper strategy because many fewer sets of parameters need to be considered
  - Evaluates combinations of features (better than a pure filtering strategy)
- Issues:
  - Can be slow relative to filtering strategies
  - The selected features are model dependent
  - Only reduces dimensions, can’t find any new ones!

Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds

Feature Extraction

- If Feature Selection is selecting a subset of features
- Then Feature Extraction is building a set of new and better (and fewer) features

Feature Extraction

Maybe none of the existing features is a particularly good feature

- Let’s build some new ones!
- You’ve already partially done this when normalizing and standardizing the existing features for regression and classification

Feature Extraction example: PCA

- The most common Feature Extraction technique is Principle Components Analysis (PCA)
- The steps of PCA are:
    1. Transform the existing features
        1. New features are uncorrelated with each other
        2. New features are ranked based on ‘importance’
    2. Extract the N most ‘important’ features

PCA:
Recoding into new features

- Assuming we have N original features
- These N original features can be recombined into N new features without losing any information where:
  - The new features are linear combinations of the original features
  - The new features are uncorrelated with each other
  - The new features are weighted in ‘importance’

PCA:
Iteratively finding new features

1. Start with the data in N dimensional space
2. Rotate the space such that the new X_1 axis captures the maximal amount of variation in the full dataset
3. Lock that dimension into place
4. Repeat steps 2 and 3 for all X_N dimensions

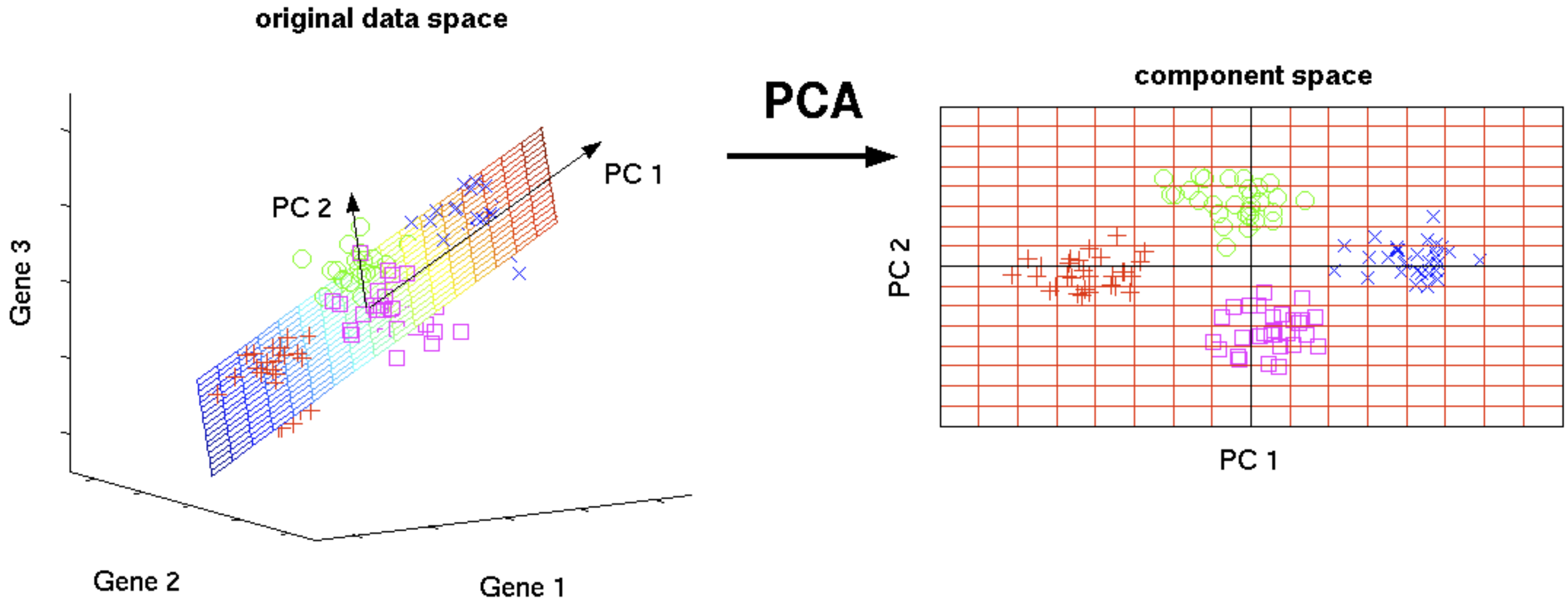
Importance of the N^{th} dimension is how much variance in the original dataset is accounted for by the X_N PCA dimension

Feature Extraction example: PCA

- Only keep a few of the new dimensions
  - Decide based on ‘importance’ scores
  - Or use model fitting and cross-validation
  - Often for visualization, this is the first 2 dimensions
- New dimensions are linear combinations of the original dimensions

Feature Extraction example: PCA

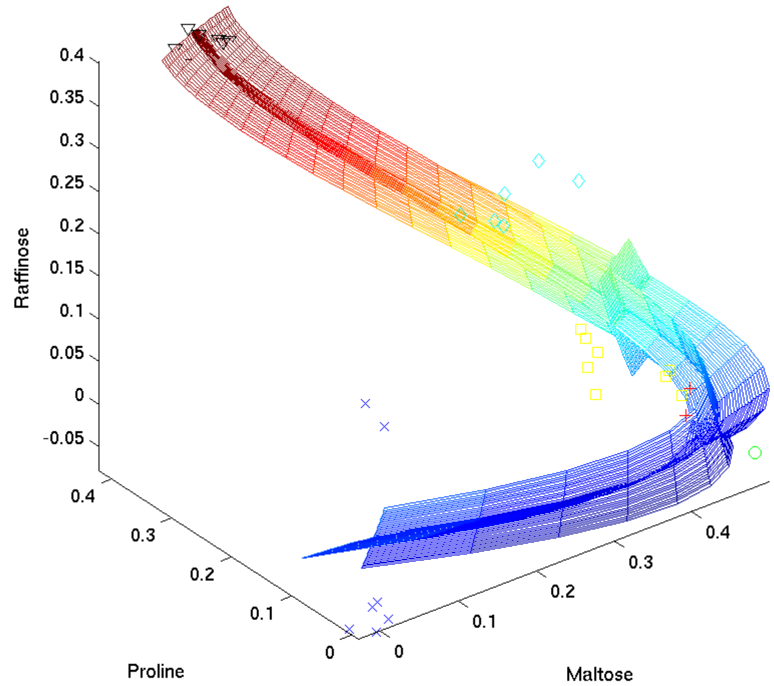
PCA is a way of projecting a high dimensional space into a lower space that has nice properties



Feature Extraction

Linear PCA is not the only game in town

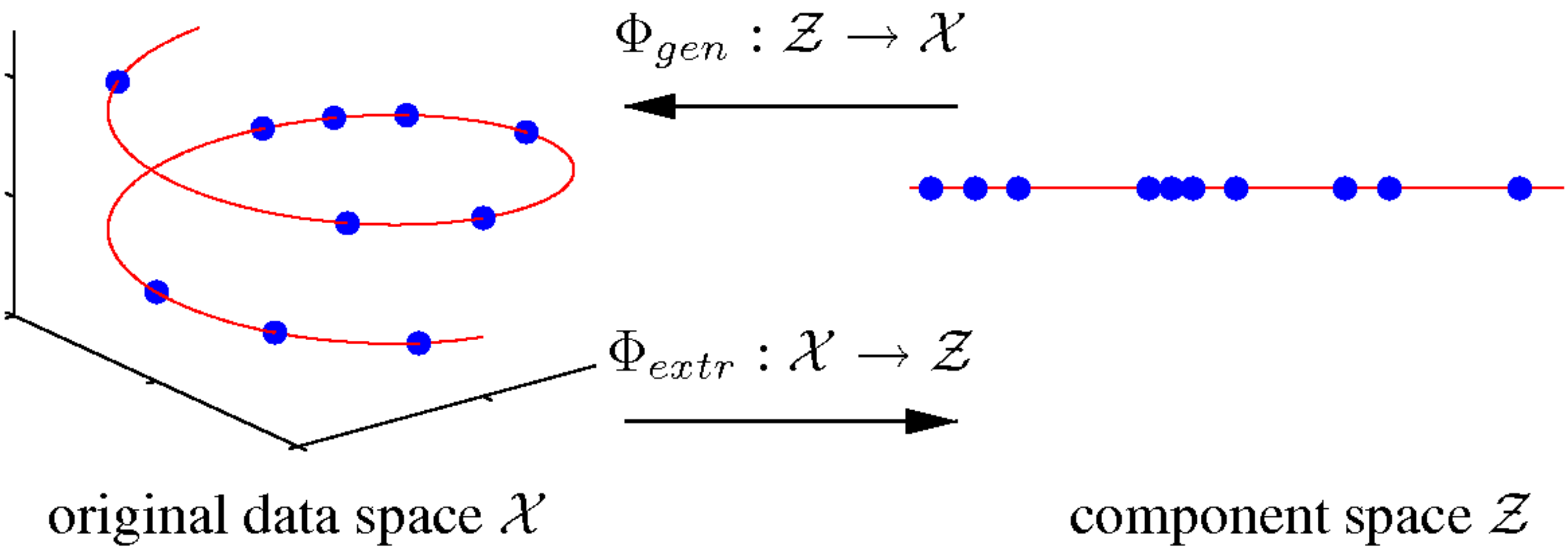
- PCA with non-linear weights



Feature Extraction

Linear PCA is not the only game in town

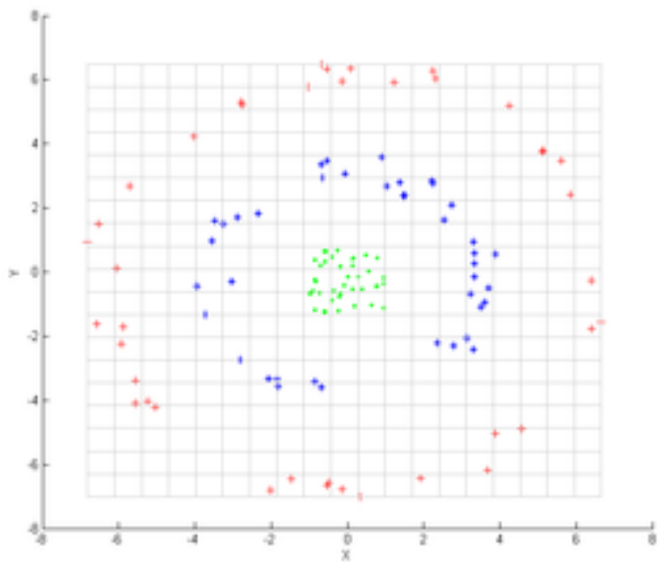
- PCA with non-linear weights
- Manifolds or curves

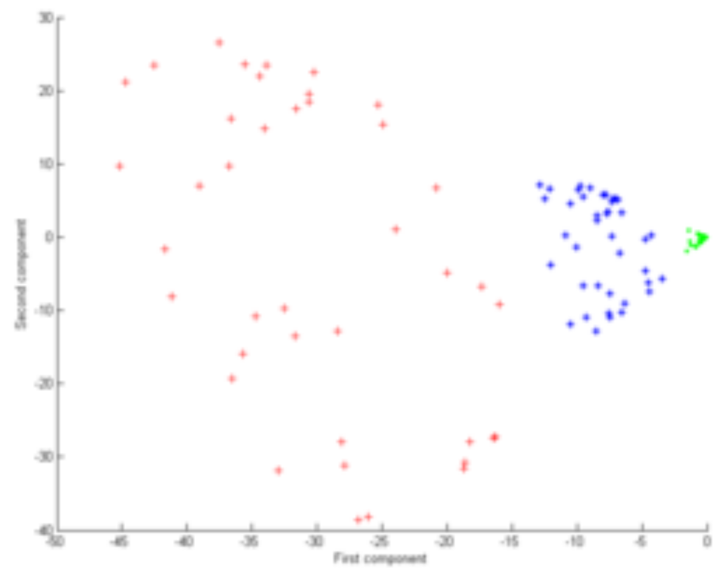


Feature Extraction

Linear PCA is not the only game in town

- PCA with non-linear weights
- Manifolds or curves
- Kernel PCA

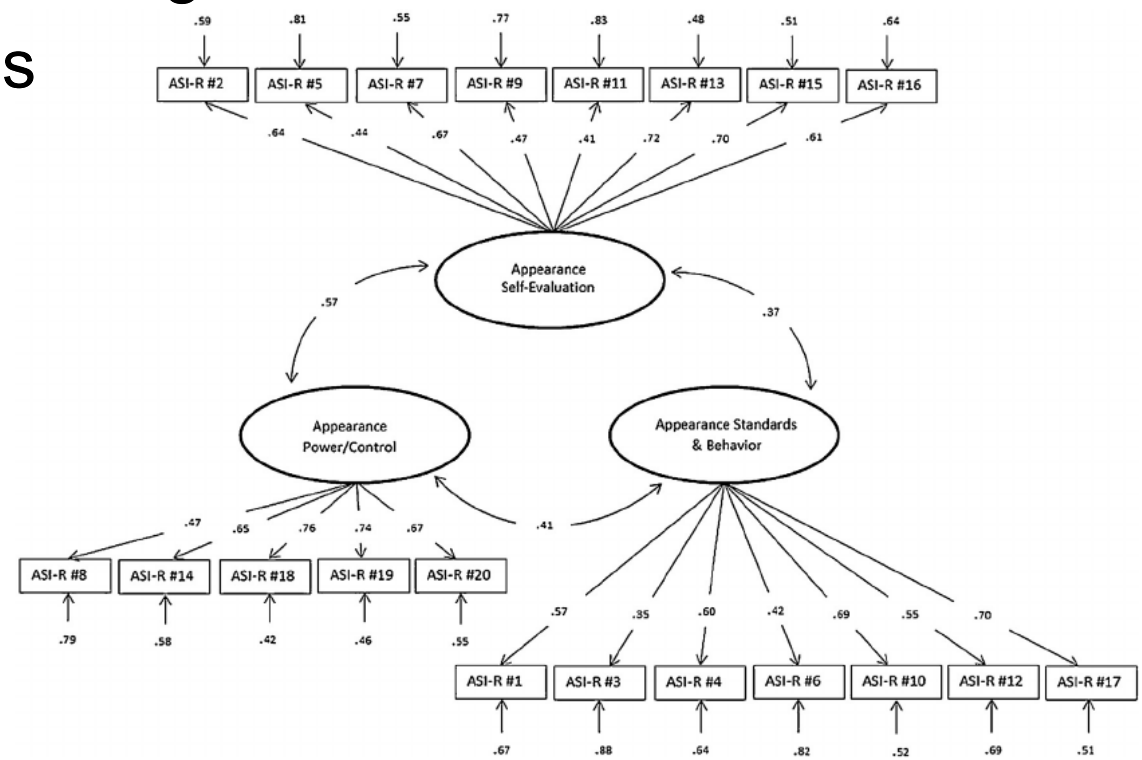




Feature Extraction

Linear PCA is not the only game in town

- PCA with non-linear weights
- Manifolds or curves
- Kernel PCA
- Factor analysis



Feature Extraction

Linear PCA is not the only game in town

- PCA with non-linear weights
- Manifolds or curves
- Kernel PCA
- Factor analysis

- https://en.wikipedia.org/wiki/Feature_extraction
- http://scikit-learn.org/stable/modules/decomposition.html

Overview: Dimension Reduction

- Why dimension reduction?
  - Visualization
  - The curse of dimensionality
- How: Feature Selection
  - Filtering strategy
  - Wrapper strategy
  - Embedding strategy
- How: Feature Extraction
  - Linear: PCA, Factor analysis
  - Non-linear: Kernel PCA, Curves, Manifolds