Dimensionality Reduction & Modeling Pitfalls

APM Ch 7

APM Ch 13

IMLP 3.4

IMLP 4.7

Table of Contents

Table of Contents

Motivation

- Visualization
- Compressing Data
- Better processing of data

Dimensionality Reduction

- Data reduction techniques are another class of feature transformations.
- These methods reduce the data by generating a smaller set of features that seek to capture a majority of the information in the original variables.
- We have looked at three techniques of dimensionality reduction in earlier chapter: Univariate Feature selection, Model Based Feature Selection and Iterative Feature Selection. These technique “select” most influential features.
- In this way, fewer variables can be used that provide reasonable fidelity to the original data. For most data reduction techniques, the new features are functions of the original features; therefore, all the original features are still needed to create the surrogate variables. This class of methods is often called *feature extraction* techniques.

Feature Extraction

- In feature extraction techniques, a new feature is created that represent original features
- the new features are functions of the original features; therefore, all the original features are still needed to create the surrogate variables.
- This class of methods is often called *feature extraction* techniques.

Principal Component Analysis

- PCA is a commonly used data reduction technique
- Principal component analysis is a method that rotates the dataset in a way such that the rotated features are statistically uncorrelated. This rotation is often followed by selecting only a subset of the new features, according to how important they are for explaining the data.
- To put all this simply, principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

Principal Component Analysis (PCA)

- PCA seeks to find linear combinations of the features, known as principal components (PCs), which capture the most possible variance

- The first PC is defined as the linear combination of the features that captures the most variability of all possible linear combinations. This is done by computed correlation between features and taking the group that have high correlation. Since the features are highly correlated, a linear combination is sufficient to capture the meaning of the feature instead of all features.

- The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.
- This continues until a total of p principal components have been calculated, equal to the original number of variables.

Demonstration of PCA

Or mathematically speaking, it's the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F745275438307707952%2F&psig=AOvVaw0gVyfuo1_3RsTm8icNYhTN&ust=1637330154631000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNiB_KOIovQCFQAAAAAdAAAAABAD

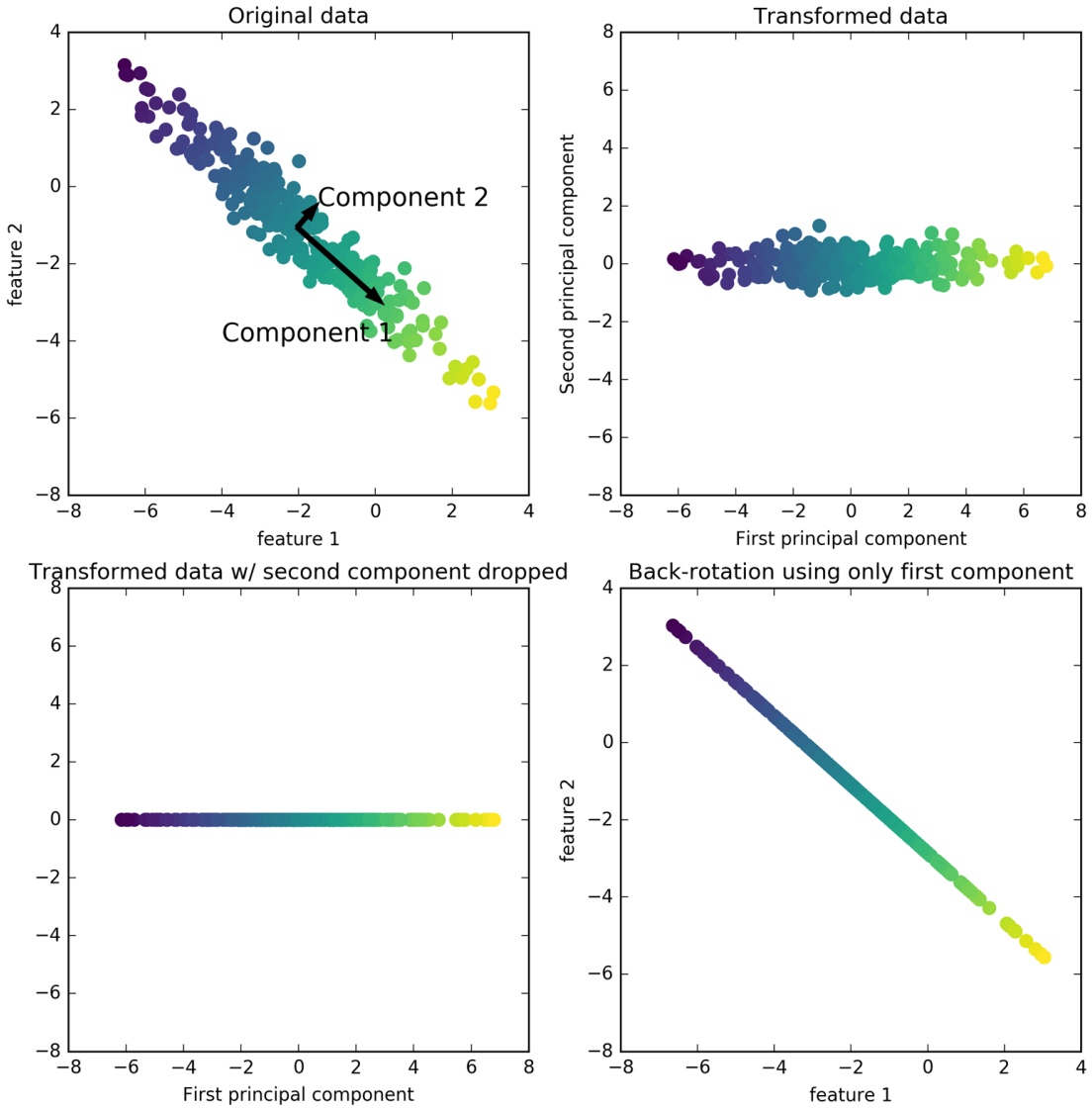
Principal Component Analysis (PCA)

j^{th} PC can be written as

$$\mathrm{PC}_j = (a_{j1} \times \text{Predictor 1}) + (a_{j2} \times \text{Predictor 2}) + \cdots + (a_{jP} \times \text{Predictor } P).$$

P is the number of predictors. The coefficients $a_{j1}, a_{j2}, \ldots, a_{jP}$ are called component weights and help us understand which predictors are most important to each PC.

Principal Component Analysis (PCA)



Math behind PCA transformation

1. Step 1 : Find the covariance matrix.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Covariance Matrix for 3-Dimensional Data

Step 2: Find eigen values: 2D, $|A-\lambda I| = 0$, solve for λ. For 2D matrix you will get two values

Step 3: Find eigenvectors: 2D, $Av = \lambda v$, for each value of λ above.

Math behind PCA transformation

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

Step 4: since the first eigenvalue is 96% of the sum of the two eigen values we say that first eigen Vector represents 96% of the variance of the data.
The second one represents the 4% of the data.

Math behind PCA transformation

Step 5: Feature Vector Determination: keep both as principle component 1 and 2.

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

Or keep v1 alone as feature vector

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

Step 6: Find the transformed data set using the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Principal Component Analysis (PCA)

- The primary advantage of PCA, and the reason that it has retained its popularity as a data reduction method, is that it creates components that are uncorrelated.

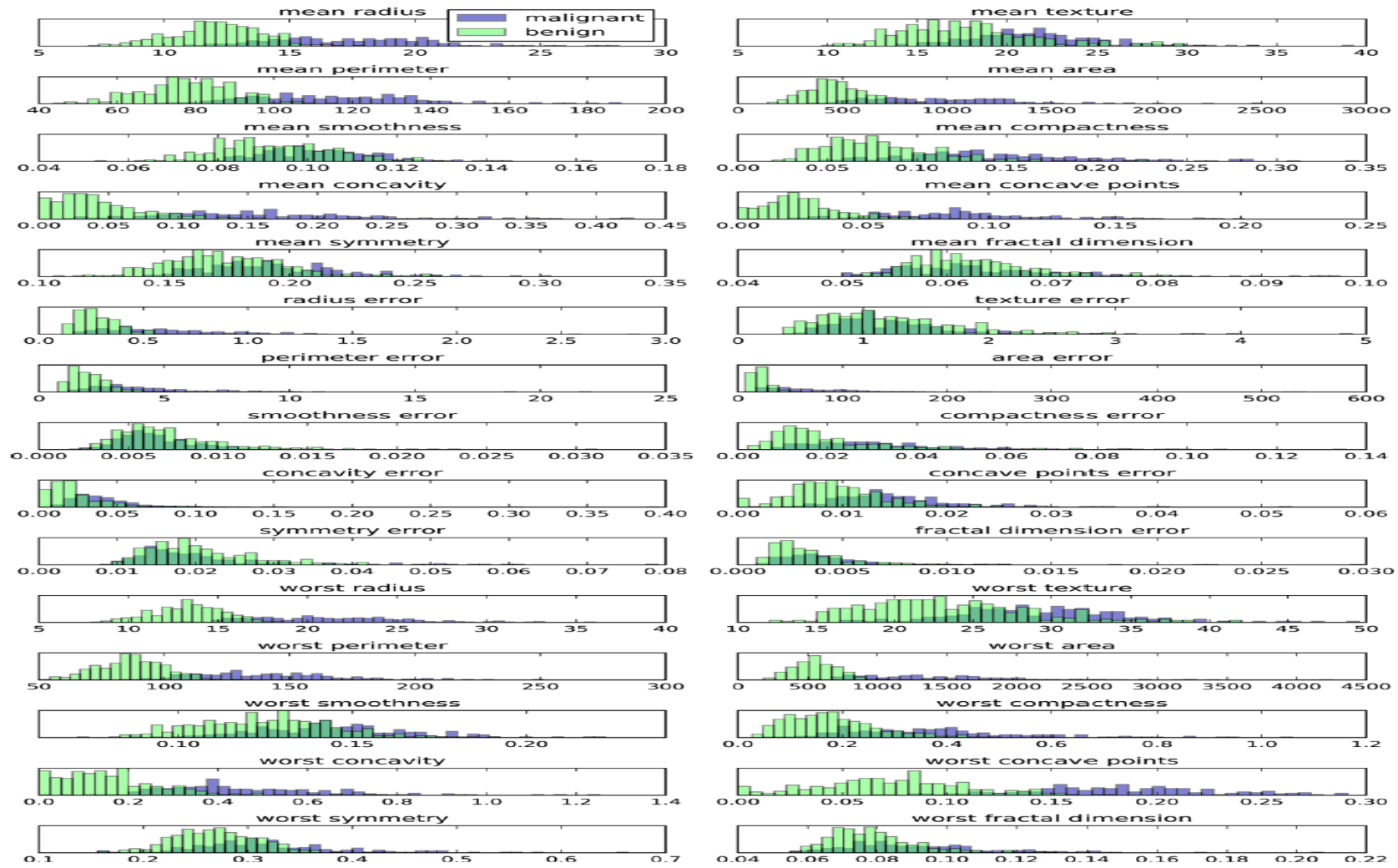
- some predictive models prefer predictors to be uncorrelated (or at least low correlation) in order to find solutions and to improve the model's numerical stability.

- PCA preprocessing creates new predictors with desirable characteristics for these kinds of models.

Histogram to understand feature interation

- Here we create a histogram for each of the features, counting how often a data point appears with a feature in a certain range (called a *bin*).
- Each plot overlays two histograms, one for all of the points in the benign class and one for all the points in the malignant class.
- This gives us some idea of how each feature is distributed across the two classes, and allows us to venture a guess as to which features are better at distinguishing malignant and benign samples.
- For example, the feature “smoothness error” seems quite uninformative, because the two histograms mostly overlap, while the feature “worst concave points” seems quite informative, because the histograms are quite disjoint.

Principal Component Analysis (PCA)



gure 3-4. Per-class feature histograms on the Breast Cancer dataset

Plotting the PCA against target

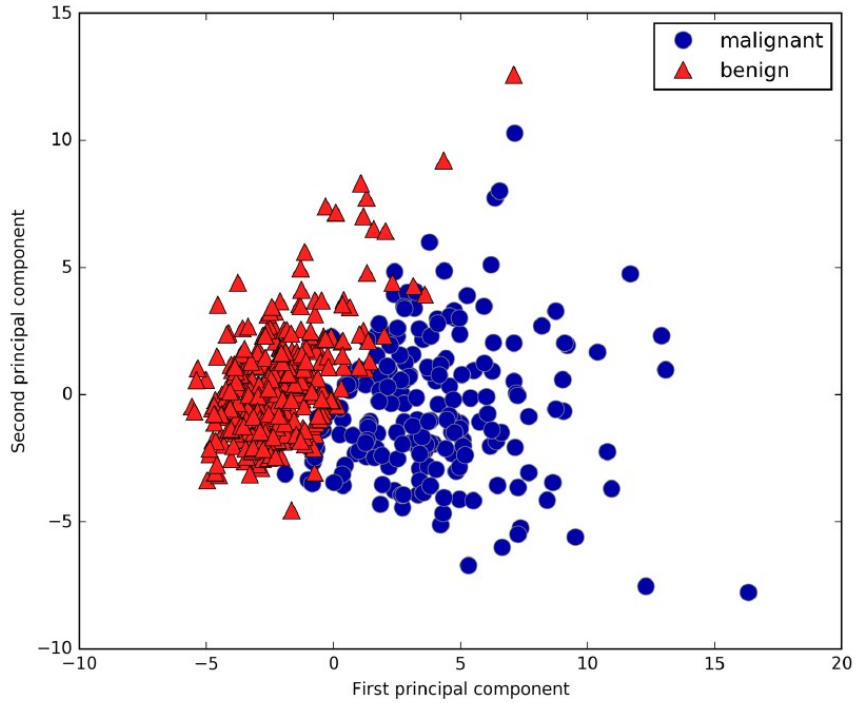


Figure 3-5. Two-dimensional scatter plot of the Breast Cancer dataset using the first two principal components

Computation

```
from sklearn.decomposition import PCA
# keep the first two principal components of the data
pca = PCA(n_components=2)
# fit PCA model to breast cancer data
pca.fit(X_scaled)
# transform data onto the first two principal components
X_pca = pca.transform(X_scaled)
print("Original shape: {}".format(str(X_scaled.shape)))
print("Reduced shape: {}".format(str(X_pca.shape)))
```

- Original shape: (569, 30)
- Reduced shape: (569, 2)

Computation

```
plot first vs. second principal component, colored by class
plt.figure(figsize=(8, 8))
mglearn.discrete_scatter(X_pca[:, 0], X_pca[:, 1],
cancer.target)
plt.legend(cancer.target_names, loc="best")
plt.gca().set_aspect("equal")
plt.xlabel("First principal component")
plt.ylabel("Second principal component")
```

Computation

```
print("PC A component shape: {}".format(pca.components_.shape))
```

PCA component shape: (2, 30)

```
print("PCA components:\n{}".format(pca.components_))
```

PCA components:

[[0.219 0.104 0.228 0.221 0.143 0.239 0.258 0.261 0.138 0.064 0.206 0.017 0.211 0.203 0.015 0.17 0.154 0.183 0.042 0.103 0.228 0.104 0.237 0.225 0.128 0.21 0.229 0.251 0.123 0.132]

[-0.234 -0.06 -0.215 -0.231 0.186 0.152 0.06 -0.035 0.19 0.367 -0.106 0.09 -0.089 -0.152 0.204 0.233 0.197 0.13 0.184 0.28 -0.22 -0.045 -0.2 -0.219 0.172 0.144 0.098 -0.008 0.142 0.275]]

Table of Contents

Clustering

clustering is the task of partitioning the dataset into groups,
called clusters. The goal is to split up the data in such a way that points within a single cluster are very similar and points in different clusters are different. Similarly to classification algorithms, clustering algorithms assign (or predict) a number to each data point, indicating which cluster a particular point belongs to.

Clustering

k-means clustering is one of the simplest and most commonly used clustering algorithms. It tries to find *cluster centers* that are representative of certain regions of the data. The algorithm alternates between two steps:

1. initialized by declaring n data points randomly as cluster centers
2. assigning each data point to the closest cluster center, and
3. then setting each cluster center as the mean of the data points that are assigned to it.

The algorithm is finished when the assignment of instances to clusters no longer changes.

Clustering

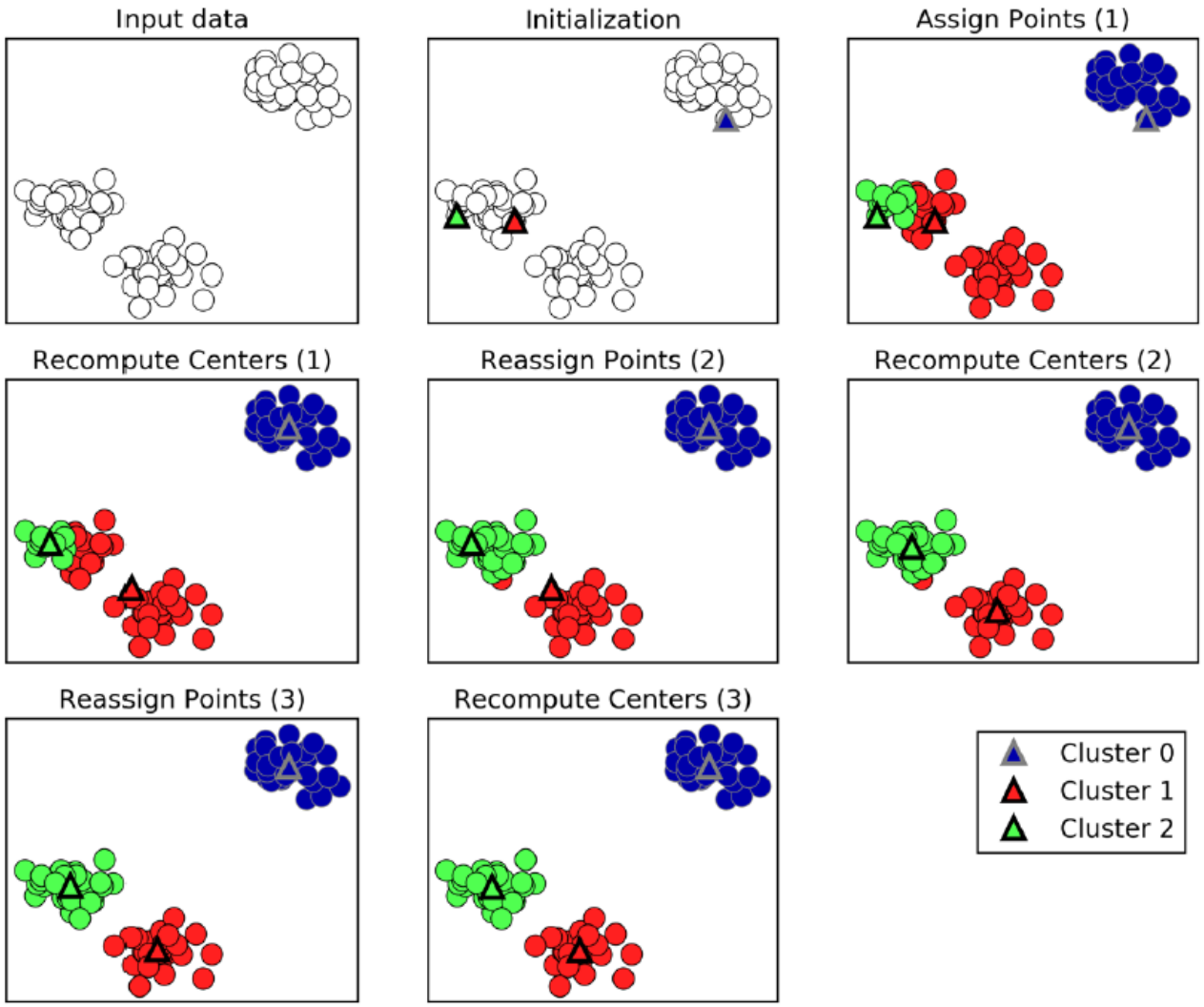


Figure 3-23. Input data and three steps of the k-means algorithm

Clustering

```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
# generate synthetic two-dimensional data
X, y = make_blobs(random_state=1)
# build the clustering model
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
```

Failure cases of K means

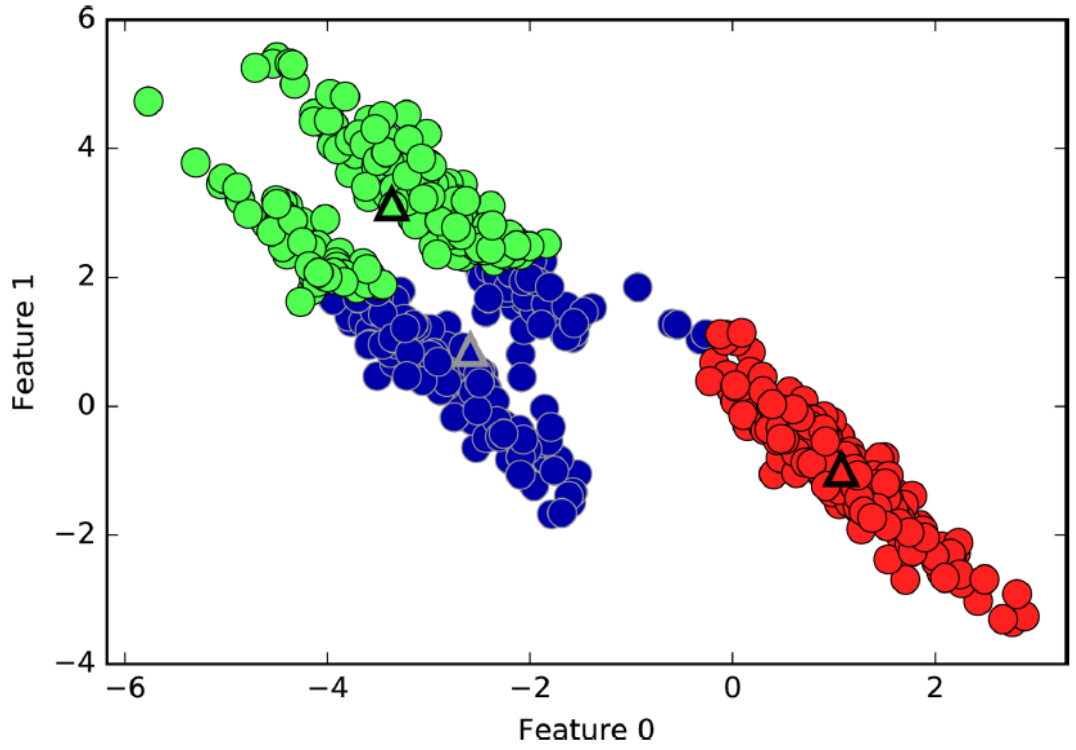


Figure 3-28. k-means fails to identify nonspherical clusters

Failure cases of K means

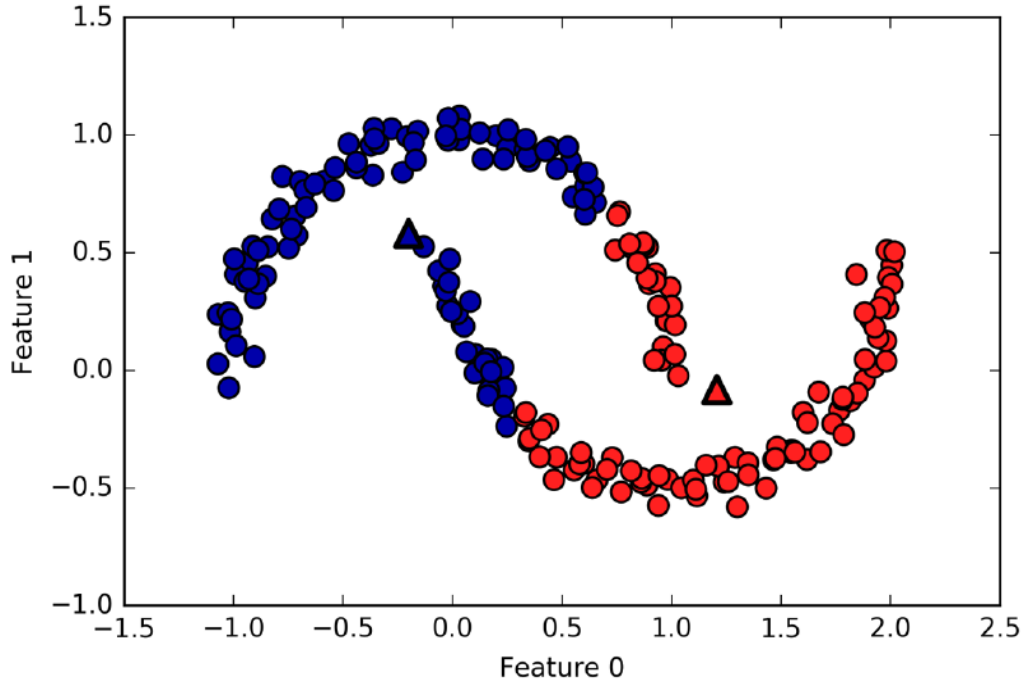
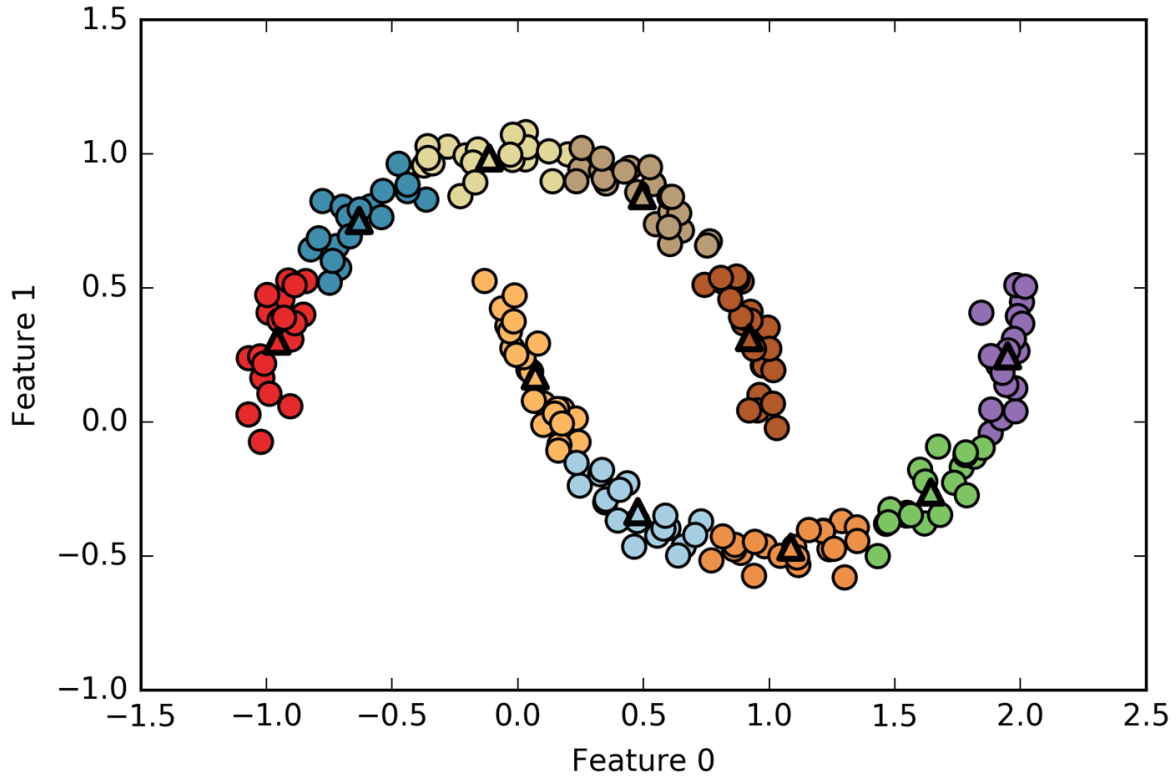


Figure 3-29. k-means fails to identify clusters with complex shapes

Clustering



Class Work

Given the following dataset on class grades on test1 and test 2, compute PCA with two components by computing

a) the covariance matrix A,

b) finding two eigen values using $|A-\lambda I|=0$,

c) computing eigenvectors by solving $Av = \lambda v$, for each λ

d) Determine the contribution of each eigenvalue to the principal component.

e) If there is one that is above 90% then use one component only, otherwise use two components.

f) Create the feature vector by using one or two Eigen vectors.

g) Translate the original data (after standardization) to the new component by using the following transformation

$$Final\_Data = (feature\ vector)^T * (original\_data\_standardized)^T$$

Test1	Test 2
9	8
10	8.5
10	10
10	9.5
9	7.5
10	8.5
8	8
9.5	8.5
9	10
10	7.5
9	8

END SESSION

Table of Contents

Motivation

- Several of the preceding chapters have focused on technical pitfalls of predictive models, such as over-fitting and class imbalances.
- Often, true success may depend on aspects of the problem that are not directly related to the model itself.
- For example, there may be limitations to what the data can support or perhaps there may be subtle obstacles related to the goal of the modeling effort.

Noise

Three general forms in most data sets:

• Since many predictors are measured, they contain some level of systematic noise associated with the measurement system.

• A second way noise can be introduced into the data is by the inclusion of non-informative predictors Some models have the ability to filter out irrelevant information, and hence their predictive performance is relatively unaffected.

• A third way noise enters the modeling process is through the response variable. As with predictors, some outcomes can be measured with a degree of systematic, unwanted noise. This type of error gives rise to an upper bound on model performance for which no pre-processing, model complexity, or tuning can overcome.

Type III Error

- One of the most common mistakes in modeling is to develop a model that answers the wrong question, otherwise known as a “Type III” error

- The true goal of the clothing store is to increase profits, but this particular campaign did not sample from all of the appropriate populations. It only utilized customers who had been contacted and made the assumption that *all customers* would mimic the behavior demonstrated in this population.1 Any model built from these data is limited to predicting the probability of a purchase *only if the customer was contacted*. This conditional statement is contrary to the goal of increasing overall profit.

Type III Error

		No contact	
		Response	Non response
Contact	Response	A	B
	Non response	C	D

The cells in the table are:

- A: customers who would respond regardless of contact
- B: customers who would respond solely because of the promotion
- C: customers who have a negative response to the promotion and would have responded if they would not have been contacted
- D: customers who have absolutely no interest in responding

- To increase profits, a model that accurately predicts which customers are in cell *B* is the most useful as this is the population associated with *new profit*.
- Costs are increased by sending promotions to customers in cells *C* or *D*.
- However, in medical research, this problem is often faced when evaluating a new treatment against an existing therapy. Here, clinical trials sometimes use *matched samples*.

Measurement Error in the Outcome

What is the minimum error rate that a model could achieve? Recall the linear regression model shown in Eq. 6.1:

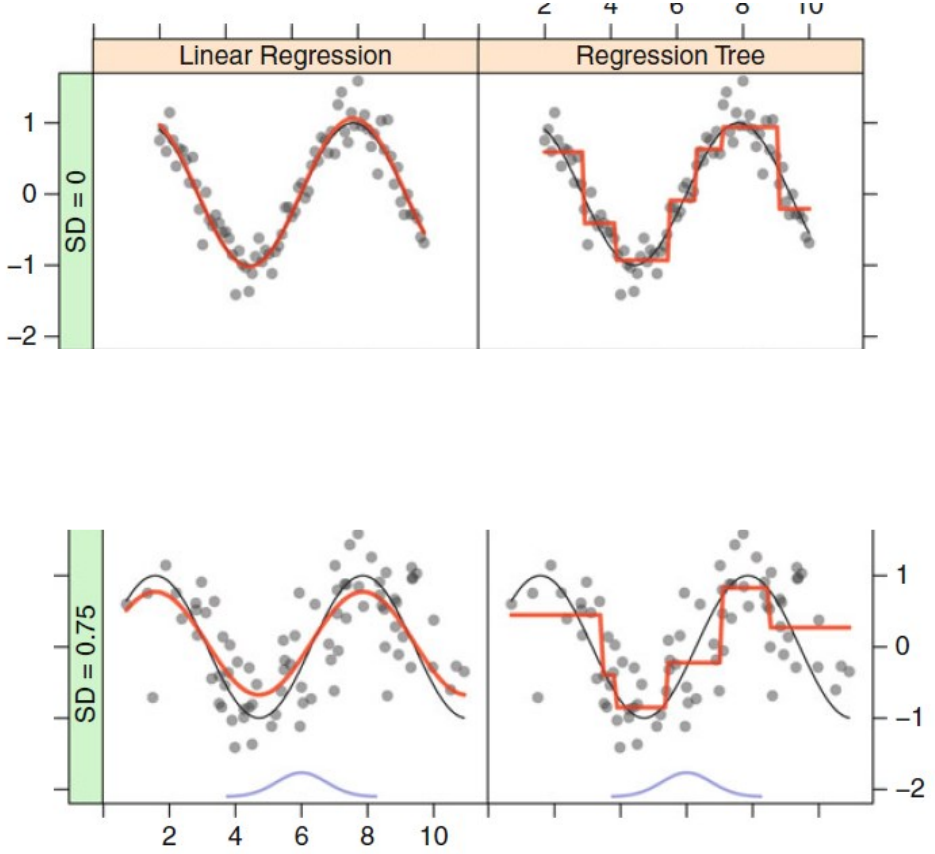
$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_p x_{ip} + e_i.$$

- e_i is the error rate.
- First, this type of noise is, noise that no model can predict—we're simply stuck with it and we cannot break through the RMSE floor or R^2 ceiling. Thus, the more the modeler knows about the measurement system, the better one can understand expectations about model performance.
- Second, as noise increases, the models become virtually indistinguishable in terms of their predictive performance.
- This means that the advantages that some of the more complex models, like ensembles, bring are only advantageous when the measurement system error is relatively low. This makes sense because the complex underlying structure that a model (such as an ensemble) can find will become more fuzzy as the noise increases.
- Therefore, we will likely be able to perform just as well with a simple, computationally efficient model when measurement system noise is high.

Measurement Error in the Predictors

- Traditional statistical models typically assume that the predictors are measured without error, but this is not always the case.
- The effect of randomness in the predictors can be drastic, depending on several factors: the amount of randomness, the importance's of the predictors, the type of model being used, as well as others.
- In some cases, if the initial data that are in the training set are generated in very controlled conditions, then the randomness can be hidden.
- For example, suppose data are generated by a person manually measuring an object (such as a rating). There may be differences in how people perceive the object, resulting in rater-to-rater noise.
- If a single rater is used for the training set data but another rate is used for test data, the bias between raters is likely to cause issues.

Example of noisy measurements



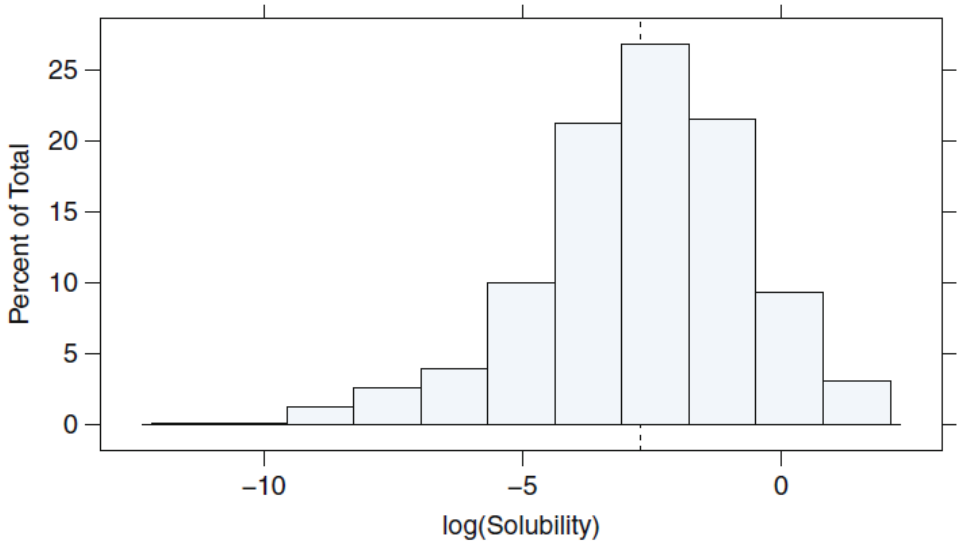
Discretizing Continuous Outcomes

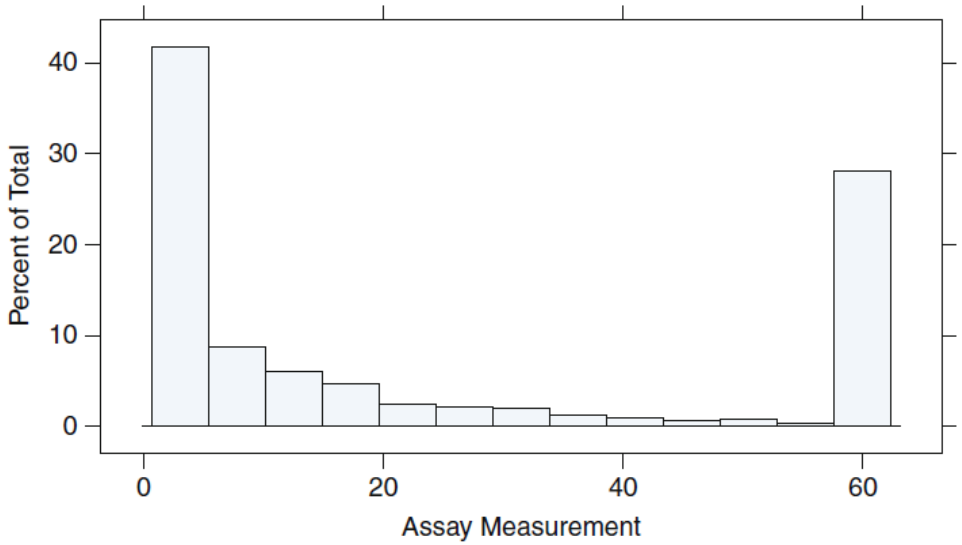
- Binning or finding bins to place continuous outcomes can have effect on the performance
  - How many bins to select
  - How do we select bins so that there is balance between different bins

- A common reason for wanting to categorize a continuous response is that the scientist may believe that the continuous response contains a high degree of error, so much so that only the response values in either extreme of the distribution are likely to be correctly categorized.

- If this is the case, then the data can be partitioned into three categories, where data in either extreme are classified generically as positive and negative, while the data in the midrange are classified as unknown or indeterminate. The middle category can be included as such in a model (or specifically excluded from the model tuning process) to help the model more easily discriminant between the two categories.

Discretizing Continuous Outcomes





When Should You (Not)Trust Your Model's Prediction?

- The predictive modeling process assumes that the underlying mechanism that generated the current, existing data for both the predictors and the response will continue to generate data from the same mechanism.
- Commercial food companies strive to create the same product over time so that customers can consistently have the same food tasting experience. Therefore companies keep the recipe, ingredients, and preparation process the same over time. If we were modeling moisture content of chocolate chip cookies from a commercial bakery, then we would expect that the predictors measured on a new batch of cookies would likely fall in the range as the predictors of the training set

When Should You (Not)Trust Your Model's Prediction?

- However, if the new data are not generated by the same mechanism, or if the training set was too small or sparse to adequately cover the range of space typically covered by the data-generating mechanism, then predictions from the model may not be trustworthy.

- The *applicability domain* of a model is the region of predictor space where "the model makes predictions with a given reliability" (Netzeva et al. 2005). Different variations of this definition exist, but the most simplistic is to define the domain in terms of similarity to the training set data.

When Should You (Not)Trust Your Model's Prediction

1 Compute the variable importance for the original model and identify the top 20 predictors

2 Randomly permute these predictors from the training set

3 Row-wise concatenate the original training set's top predictors and the randomly permuted version of these predictors

4 Create a classification vector that identifies the rows of the original training set and the rows of the permuted training set

5 Train a classification model on the newly created data

6 Use the classification model to predict the probability of new data being in the class of the training set.

Algorithm 20.1: Algorithm for determining similarity to the training set

Applicability Domain

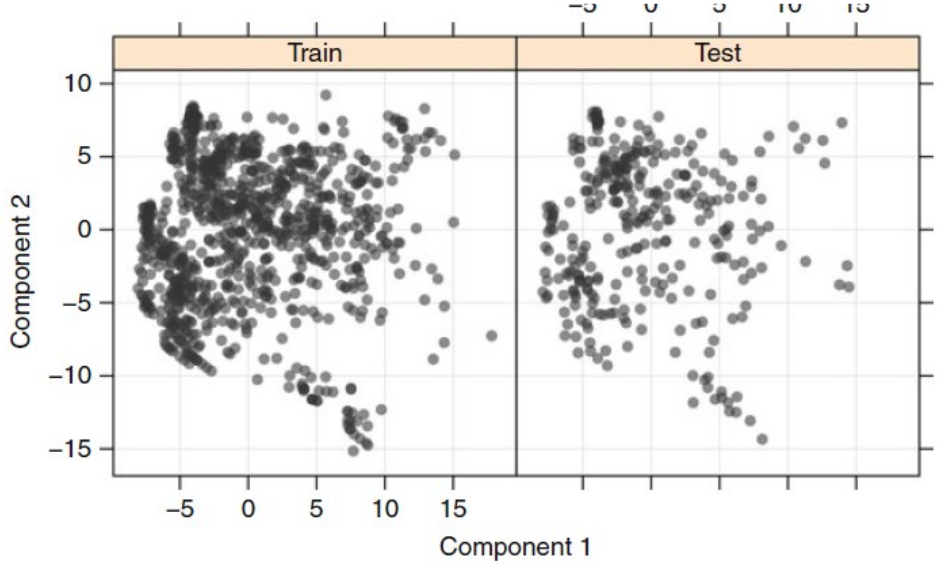


Fig. 20.7: PCA plots for the solubility data

Classification algorithms to check similarity to training set

- One can develop classification algorithms to check if a given sample is in the same class as original data.

Impact of size of the data

- As the dimensions of the data increase, computation time likewise increases. Moreover the computational burden for ensembles of trees is even greater, often requiring more expensive hardware and/or special implementations of models that make the computations feasible.
- Second, there are diminishing returns on adding more *of the same data* from the same population. Since models stabilize with a sufficiently large number of samples, garnering more samples is less likely to change the model fit.