Features: representation, normalization, selection

Chapter e-9

features

- Distinguish between instances (e.g. an image that you need to classify), and the features you create for an instance.
- Features are the workhorses of machine learning: the quality of a classifier is crucially dependent on the features used to represent the domain.

- Although many datasets come with pre-defined features, they can be manipulated in many ways

features

- Distinguish between instances (e.g. an image that you need to classify), and the features you create for an instance.
- Features are the workhorses of machine learning: the quality of a classifier is crucially dependent on the features used to represent the domain.

- Although many datasets come with pre-defined features, they can be manipulated in many ways
  - Normalization or other transformations
  - Discretization
  - Select the best features
  - Combine features to compute new ones (feature construction)

types of features

Quantitative (continuous) features: have a meaningful numerical scale

- Examples: age, height etc.

Ordinal features: have an ordering

- Example: house number

Categorical (discrete/nominal) features: not meaningful to describe them using mean, median

- Example: Boolean features

features and classifiers

Some classifiers require quantitative features (all the classifiers we've seen so far).

Need to transform an ordinal/categorical feature into a quantitative feature.

Assume we have a feature with possible values "red", "blue", "green".

How would we transform this into one or more quantitative features?

features and classifiers

Some classifiers require quantitative features (all the classifiers we've seen so far).

Need to transform an ordinal/categorical feature into a quantitative feature.

Assume we have a feature with possible values "red", "blue", "green".

How would we transform this into one or more quantitative features?

Use one-of-c coding

features and classifiers

Some classifiers treat categorical and quantitative features differently.

Examples: Naïve Bayes, decision trees

Sometimes useful to use categorical features even if the underlying data is quantitative.

Solution: discretization

feature transformations

Aim: improve the utility of our features by changing them in some way.

scaling

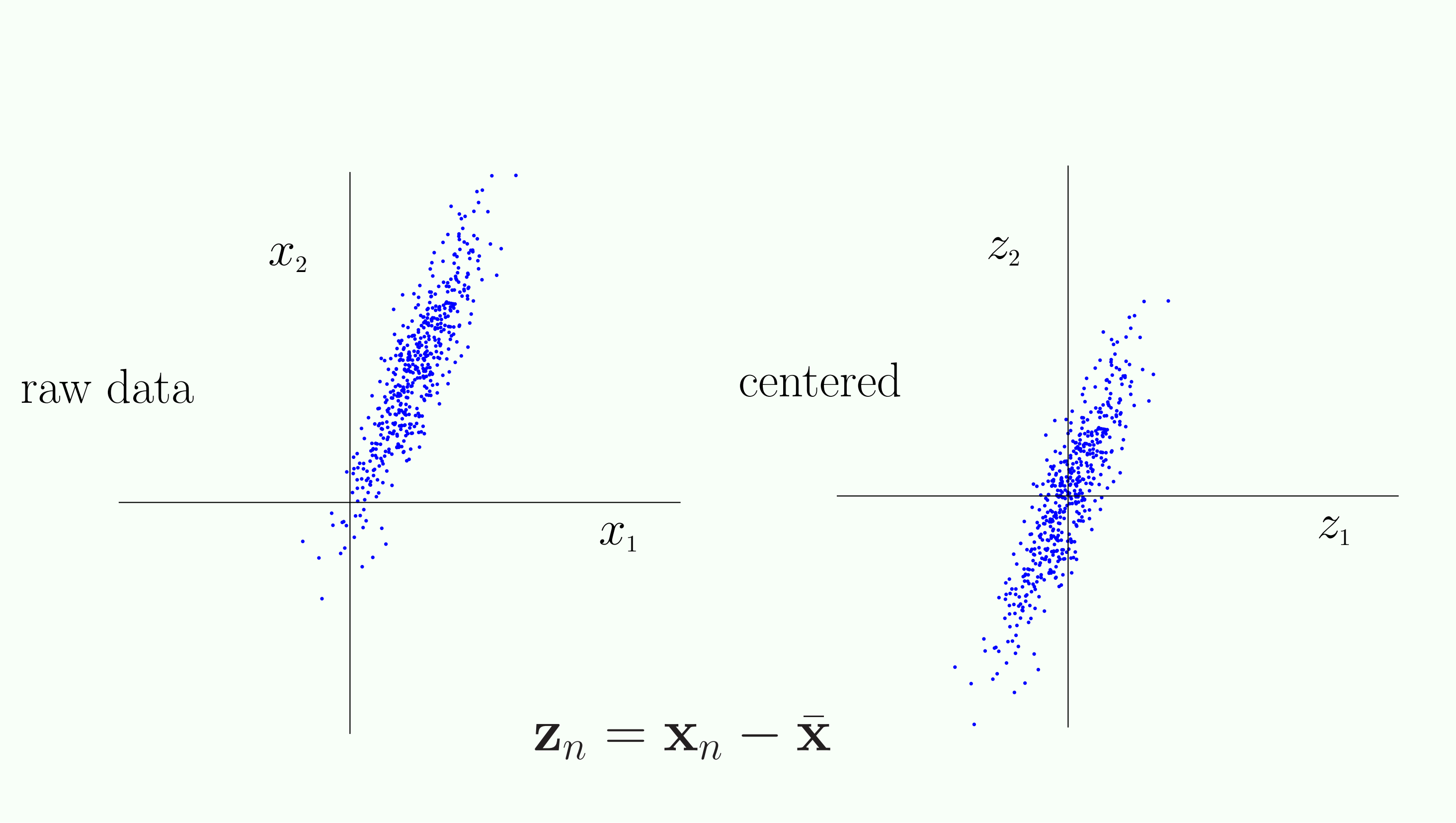
Feature scaling: neutralize the effect of different scales across features (geometric classifiers are sensitive to that).

- Centering
- Standardization
- Scaling to [0,1]

Instance scaling: scale a feature vector to have unit norm. Appropriate for sparse data.

centering

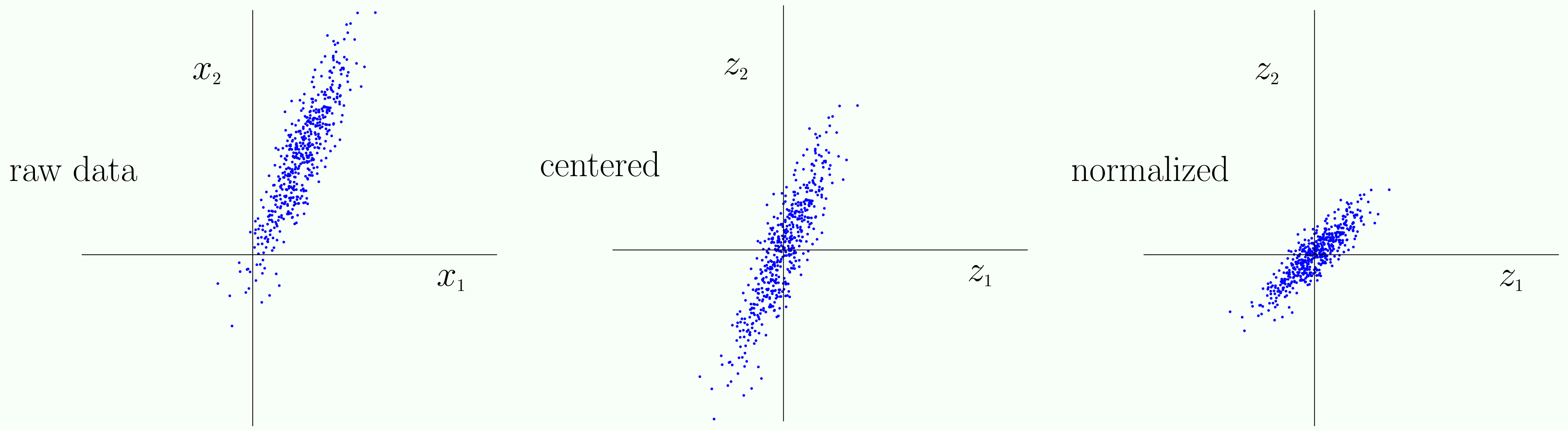
The geometry of centering:



$$\mathbf{z}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

standardizing

The geometry of standardizing:



$$\mathbf{z}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

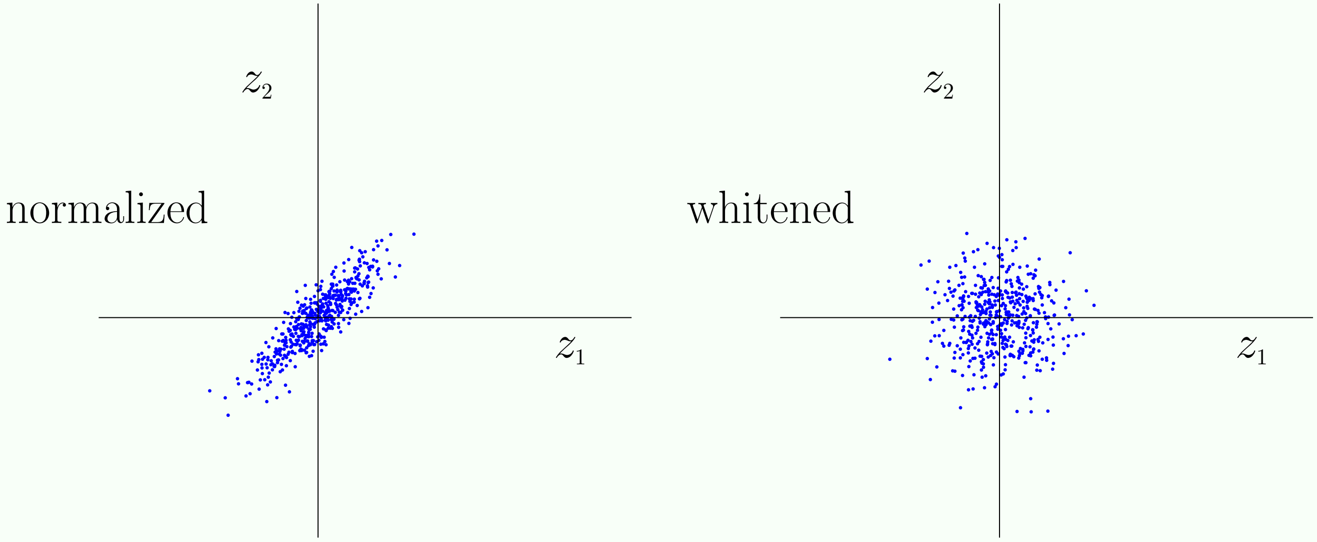
$$\bar{\mathbf{z}} = \mathbf{0}$$

$$\mathbf{z}_n = \mathrm{D}\mathbf{x}_n$$

$$D_{ii} = \frac{1}{\sigma_i}$$

whitening

Whitening: transforming the data so that features are not correlated.



$$\mathbf{z}_n = \mathrm{D}\mathbf{x}_n$$

$$D_{ii} = \frac{1}{\sigma_i}$$

$$\mathbf{z}_n = \Sigma^{-\frac{1}{2}}\mathbf{x}_n$$

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} = \frac{1}{N}\mathrm{X}^{\mathrm{T}}\mathrm{X}$$

missing data

In many cases an instance may be incomplete, i.e. some feature values will be missing.

Decision tree models and some probabilistic models can handle this well.

Geometric models (SVM, KNN) need all features to be specified.

Solution: fill in missing values (imputation).

Simple solution: use the mean

More sophisticated: build a predictive model

Complications: the fact that the feature is missing might be correlated with the label

feature selection and construction

Distinguish between

Feature selection: select a subset of features that allow a classifier to maintain or increase its performance.

Feature construction: construct new features that are a linear/ non-linear combinations of the original features.

feature selection

Why do feature selection?

feature selection

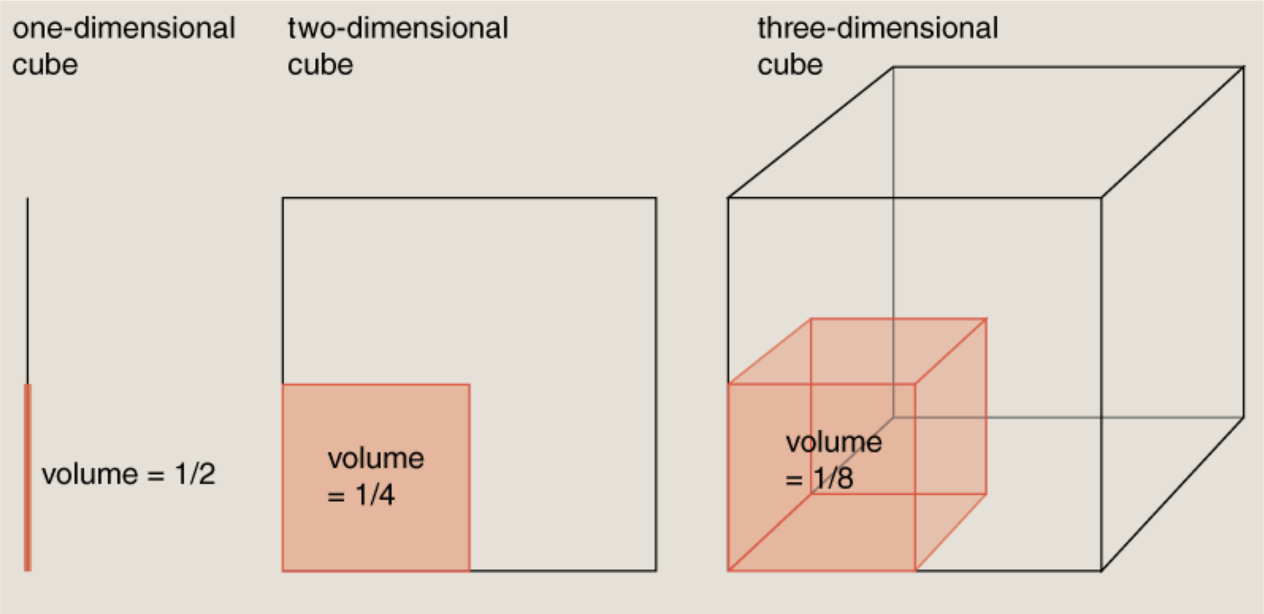
Why do feature selection?

- Better understanding of the data and the classification rule.
- Expensive to compute/measure all features
- Improve classifier performance: some machine learning algorithms, are known to degrade in performance when faced with many irrelevant/noisy features

the curse of dimensionality

A fixed number of data points sparsely populates a space as its dimensionality increases.

In other words: the number of data points required for populating the space with equal density grows exponentially with dimension.



the curse of dimensionality

The VC dimension of a linear classifier in d dimensions is equal to d+1

As a consequence, our ability to fit the data increases with increasing dimensionality.

feature selection

Objective: find a subset of the original features, such that a classifier that is run on data containing only these features generates a classifier with the highest possible accuracy.

Feature selection can lead to improved performance.

This depends on the classifier used: this is definitely the case for KNN or naive Bayes classifiers, but not necessarily the case for SVMs/neural networks.

Comment: Relevance does not imply that a feature should be in an optimal feature subset (redundancy)

approaches

Filter. Features or groups of features are scored by some measure of relation with the labels. Examples of scoring functions include Pearson correlation between a feature and the labels, Fisher scores, or mutual information.

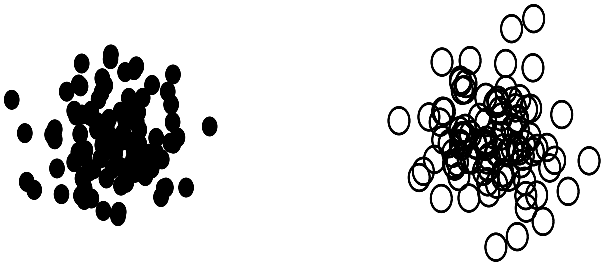
Wrapper. Uses the classifier to guide the process of selection. A feature subset is typically evaluated using CV.

Embedded. The selection method uses properties of the classifier.

filter methods

- Pearson correlation of a feature with the labels
- ROC score of a feature
- Difference in a feature's mean across classes:

the Golub score: $$\frac{|\mu_i(\oplus) - \mu_i(\ominus)|}{\sigma_i(\oplus) + \sigma_i(\ominus)}$$



potential drawbacks of filter methods

Consider the following data:

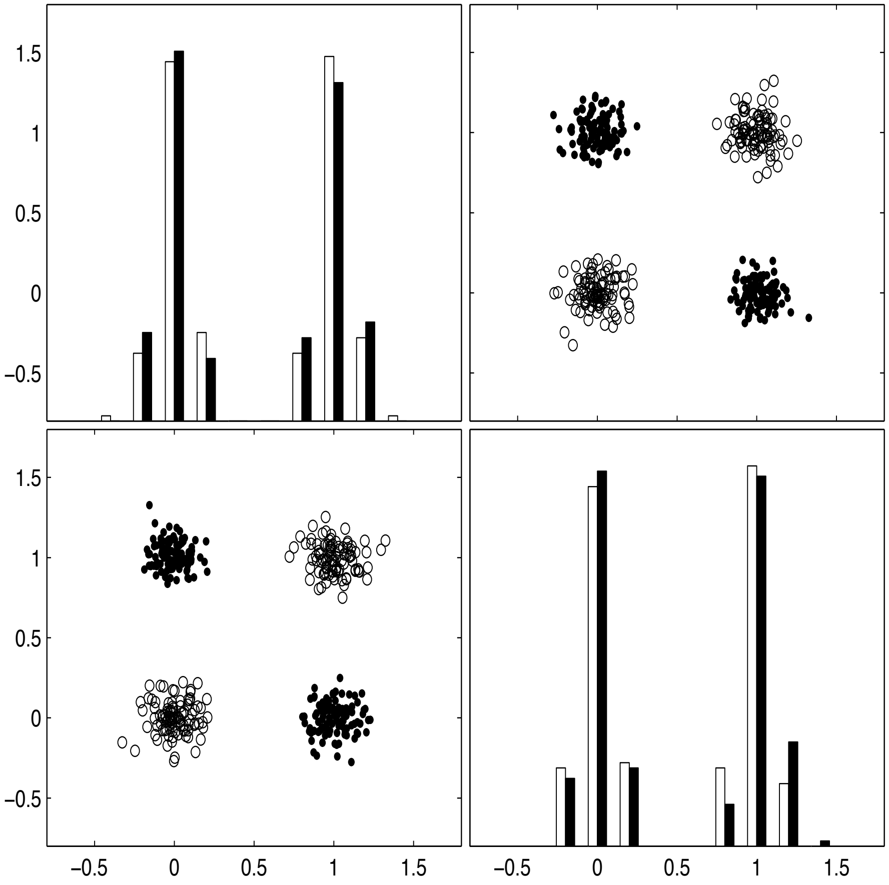


Figure from: An Introduction to Variable and Feature Selection
Isabelle Guyon, André Elisseeff; 3(Mar):1157-1182, 2003.

how many features to select?

Drawbacks of filter methods?

How to decide on how many features?

- Decide ahead of time.
- Use CV.

wrapper methods

Iteratively add features (forward selection) or eliminate them (backward elimination)

Use cross validation to guide feature inclusion/removal:

Rank features by how much they add to accuracy, or by how much accuracy decreases by their removal.

This is a greedy approach.

How many runs of cross-validation do we need to run on a dataset with d features?

The idea of wrappers for feature selection comes from:
Wrappers for feature subset selection. Ron Kohavi and George H. John, 1997

embedded methods

We have seen that the magnitude of the weight vector of a linear classifier can be used as a measure if feature importance.

Recursive Feature Elimination (RFE):

Alternate between training an SVM and removing the feature with the lowest magnitude of the weight vector.

For high dimensional datasets you can remove a fraction of the features at each iteration

Gene selection for cancer classification using support vector machines.
I. Guyon, J. Weston, S. Barnhill, and V. Vapnik.
Machine Learning, Vol. 46, pp. 389—422, 2002

embedded methods

We have seen that the magnitude of the weight vector of a linear classifier can be used as a measure if feature importance.

Recursive Feature Elimination (RFE):

Alternate between training an SVM and removing the feature with the lowest magnitude of the weight vector.

Inspired by LeCun's "Optimal brain damage" weight pruning technique used in neural networks.

Gene selection for cancer classification using support vector machines.
I. Guyon, J. Weston, S. Barnhill, and V. Vapnik.
Machine Learning, Vol. 46, pp. 389—422, 2002

which type of methods should I use?

The wrapper and embedded approaches take into account the heuristics and biases of the classifier that will ultimately be used, and thus can potentially generate better feature subsets.

Filter methods are less likely to suffer from overfitting.

So ultimately, it depends on the data.

bias in feature selection

The wrong way to evaluate feature selection:

Perform feature selection on the whole dataset and then run cross-validation.

Why is this wrong?

bias in feature selection

The wrong way to evaluate feature selection:

Perform feature selection on the whole dataset and then run cross-validation.

The right way:

Perform feature selection using training data only.

Feature selection in scikit-learn

Feature selection methods:

- RFE
- Filter methods

Support proper experiment design

Show a demo!