Data Mining and Knowledge Discovery:

Data Transformation

Bernard Ženko

Jožef Stefan International Postgraduate School, 2021/22

Outline

- Semi-supervised learning

- Treating missing values
- Discretization of numeric features
- Discrete to numeric transformation
- Multi-class to binary transformation
- Normalization and standardization
- Feature scoring, ranking and selection
- Projections (dimensionality reduction)
- Feature construction (engineering)

Semi-supervised learning

	Ambient pressure	Ambient temperature	Ambient humidity	Air filter pressure	Turbine inlet temperature	Turbine after temperature	CO
1	1018.7	4.5878	83.675	3.5758	1086.2	549.83	0.32663
2	1018.3	4.2932	84.235	3.5709	1086.1	550.05	0.44784
3	1018.4	3.9045	84.858	3.5828	1086.5	550.19	0.45144
4	1018.3	3.7436	85.434	3.5808	1086.5	550.17	0.23107
5	1017.8	3.7516	85.182	3.5781	1085.9	550	0.26747
6	1017.7	3.8858	83.946	3.5824	1086	549.98	?
7	1018	3.6697	84.114	3.5804	1085.9	550.04	?
8	1018.2	3.5892	83.867	3.5777	1086	549.88	?
9	1018.5	3.7108	84.948	3.6027	1086.3	549.98	?
10	1018.5	4.8281	85.346	3.5158	1083.1	549.8	?
11	1018.4	6.0334	86.054	3.5748	1085.8	550.04	?
12	1017.9	7.9896	87.541	4.3905	1094.5	534.81	?
13	1017.2	9.8712	84.249	4.6983	1100.2	530.58	?

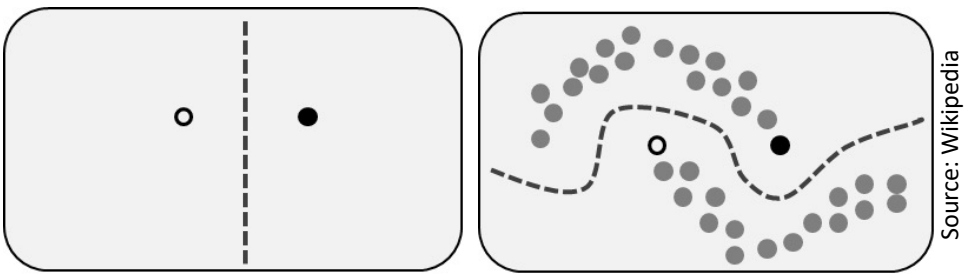
labeled

unlabeled

- Are the unlabeled examples useful in any way?

Semi-supervised learning

- Problem:
  - We have a large amount of unlabeled data, but only small amount of labeled data
  - We want to learn a predictive model (predict the labels)
- Question: Can we use the unlabeled data to improve the accuracy of the model learned exclusively on labeled data?
- Answer: In principle yes, the unlabeled data can be used to better describe attribute's distributions



Source: Wikipedia

Semi-supervised methods

- Generative models: They assume some (parametrized) form of model (e.g., probability distributions) and use all data to fit the parameters; If assumption is incorrect we can get worse results
- Low-density-separation: Boundary between classes should lie in regions with few data points
- Heuristic approaches: Supervised methods applied on labeled and unlabeled data in some way, e.g., self-training

Self-training

1. Train a supervised model on labeled data only
2. Use this model to predict labels of unlabeled data
3. Select examples with *reliably* predicted labels and add them to the training set in step 1
4. Stop when no new example is added to the training set

- Any supervised method that enables estimation of the reliability of predictions can be used

Predictive clustering for semi-supervised learning

- PCT defines clusters with similar examples according to more than one attribute (i.e., all clustering attributes), not just the target one (for which we are missing many labels)
- PCTs can be learned directly on a training set that contains labeled and labeled examples in one step
- Requirement: the similarity measures and prototype functions must be able to deal with missing values

Missing values

- Some algorithms have inherent ability to deal with some missing values:
  - decision trees: if an example is missing an attribute value that appears in the tree split condition, the example is split in half, and half of the example is assigned to the left and half to the right tree branch
- Most of the algorithms cannot deal with missing values directly
- Possible solutions:
  - remove all examples with missing values: we can loose a lot (sometimes all) of data
  - imputation: we replace the missing values with some substitute values

Imputation approaches

- Mean (most frequent) substitution: missing value is replaced by the mean of the same variable for all non-missing examples
- Random substitution: missing value is replaced with a random number, possibly taking into account the known distribution of the variable
- Manual substitution: missing value is replaced manually
- Model-based: a model for predicting the attribute with the missing value based on all other attributes is constructed and used for determining the substitution value:
  - 1-NN: (hot-deck imputation), the value from the most similar example is used
  - simple tree model (Orange: Impute)

On terminology

- The following terms are typically used interchangeably:
  - features, (descriptive) attributes, (descriptive, independent) variables
  - class, target (variable), dependent variable

Discretization of numeric features

- Unsupervised
  - feature is discretized without any knowledge about the class feature
  - equal-width discretization (equal-interval binning, ...): some bins might be empty, because examples can be distributed unevenly
  - equal-frequency discretization (histogram equalization binning, ...): might produce bad boundaries, because it does not respect class divisions
- Supervised
  - discretization of a feature depends also on the class feature
  - entropy-based discretization:
    - we use the technique for splitting numeric attribute when learning decision trees and recursively split the interval into subintervals that are as pure as possible
    - we use the MDL principle (Minimum Description Length, Occam's Razor principle) to stop splitting
    - see, e.g., (Witten et al., 2016), Ch. 8 for more details

- These approaches could also be applied to ordered nominal features, when we want to reduce the number of their distinct values
- Orange: Discretize

Discrete to numeric transformation

- Distance based methods (e.g., kNN):
  - we can just update the distance $d(x_1, x_2) = \begin{cases} 0; & x_1 = x_2 \\ 1; & x_1 \neq x_2 \end{cases}$

- If the nominal attribute is ordered, the labels can be replaces with integers, but be aware that this also implies the metric

- Orange: Continuize

Multi-class to binary transformation

- One-vs-rest technique (one-vs-all):
  - for each class value we create one data set (learning problem)
  - the original class values are changed into 'yes' or 'no', depending whether the example belongs to the selected class or not
  - each of the resulting models predicts whether an example belongs to this class or not
  - we need to run all models and aggregate their results – if the models also provide us with confidence, we select the class associated with the model that most confidently says 'yes'
- One-vs-rest can also used for transforming (non-class) nominal attributes into a set of binary attributes (one-hot encoding)
- Orange: within Continuize

- Other approaches:
  - pairwise classification (round robin binarization): we generate one data set for each class value pair including only examples belonging to these two classes
  - error-correcting output codes: we can generate more binary attributes than the number of nominal values and encode the values in a way to maximize the Hamming distance among them

Normalization (and standardization)

- When a learning algorithm is sensitive to absolute values of the (numeric) features (e.g., kNN) we need to somehow 'equalize' them
- In general, this is a nontrivial problem due to different value distributions
- (If we assume normal distribution,) we can compute mean and variance and:
  - center to: mean = 0
  - scale to: variance = 1
  - standardize to: mean = 0, variance = 1
- min-max normalization:
  - to [0,1] interval
  - to [-1,1] interval
  - in general to [a,b] interval: $x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$

- Question: How do we compute min, max, mean, variance?

- For more details, see e.g., https://en.wikipedia.org/wiki/Feature_scaling
- Orange: within Continuize

Feature (scoring, ranking,) selection

- Data can include a lot of features that are not relevant to our problem (target variable)
- For modeling we want to use only relevant features:
  - to simplify the work for the learning algorithm
  - to get better results (some learning algorithms have problems with too many features, and some features might be included in the model by chance, even if they are not relevant)
- Given a feature evaluation measure, we can:
  - score features
  - rank them based on the scores
  - select N best ranked features
- In general, scoring measures can be unsupervised or supervised

- Orange: Rank

Feature scoring measures (in Orange)

- Information Gain: the expected amount of information (reduction of entropy)
- Gain Ratio: a ratio of the information gain and the attribute's intrinsic information, which reduces the bias towards multivalued features that occurs in information gain
- Gini: the inequality among values of a frequency distribution
- ANOVA: the difference between average values of the feature in different classes
- Chi2: dependence between the feature and the class as measured by the chi-square statistic
- ReliefF: the ability of an attribute to distinguish between classes on similar data instances
- FCBF (Fast Correlation Based Filter): entropy-based measure, which also identifies redundancy due to pairwise correlations between features

- Specific learning algorithms can (in addition to a model) also provide feature scores:
  - Random forest
  - Logistic Regression
  - Gradient Boosting
  - …

Wrapper approach to feature selection

- Feature selection can also be performed with an arbitrary learning algorithm:
  - given a subset of features, we can learn a model and estimate its performance, e.g., classification accuracy with cross-validation
  - find a subset that results in the best performance
- In practice we use forward or backward greedy search for finding the 'best looking' subset of features
- Benefit: The features are not evaluated independently, and in principle we can identify a subset of features that are complementary, without the redundant ones
- Weka: Select attributes/WrapperSubsetEval

Feature importance

- How much does a feature contribute to the performance of a model?
- For interpretable models (trees, rules, equations) we can inspect the model and ...
- For all models we can use techniques such as permutation feature importance:
  - Given a trained model we evaluate it on (training) data and measure the decrease of model's performance when some values of a given feature are permuted (randomly changed)
- This approach could also be used as a (supervised) feature scoring measure
- Orange: Feature Importance

Projections

- Sometimes it is useful to transform our data in some way. In addition to problem specific transformations, there are also 'universal' transformations:
  - principal component analysis (PCA)
  - random projections
  - partial least squares regression (PLS)
  - independent component analysis (ICA)
  - linear discriminant analysis
  - ... see, e.g., (Witten et al., 2016), Ch. 8 for more details
- PCA:
  - the original data set can be represented in a k-dimensional space, where k is the number of attributes (attributes represent (possibly non-orthogonal) coordinates of this coordinate system)
  - PCA finds a new orthogonal coordinate system (new attributes) and transforms data into it
  - coordinates are ordered according to importance, i.e., how much variance of the data they describe (more important new attributes come first)
- Feature selection (dimensionality reduction): we can select only the first N new attributes that describe enough variance in the data
- Orange: PCA

Feature construction

- The original data can be used to construct (new) features that will:
  - make relevant information more easily available to the learning algorithm
    - general (e.g., PCA) and problem-tailored transformations
    - Orange: Feature Constructor
  - enable tabular (attribute-value) representation:
    - text: bag of words representation
    - images: scale invariant descriptors (SIFT, SURF)
    - time-series: sliding window approach (tsfresh Python library)
  - deep neural networks: can learn on data directly and automatically construct features, but are computationally expensive are require a lot of data