2.3. Clustering
2.3.1. Overview of clustering methods
2.3.2. K-means
2.3.2.1. Mini Batch K-Means

2.3.3. Affinity Propagation
2.3.4. Mean Shift
2.3.5. Spectral clustering
2.3.5.1. Different label assignment strategies
2.3.5.2. Spectral Clustering Graphs

2.3.6. Hierarchical clustering
2.3.6.1. Different linkage type: Ward, complete, average, and single linkage
2.3.6.2. Adding connectivity constraints
2.3.6.3. Varying the metric

2.3.7. DBSCAN
2.3.8. Birch
2.3.9. Clustering performance evaluation
2.3.9.1. Adjusted Rand index
2.3.9.1.1. Advantages
2.3.9.1.2. Drawbacks
2.3.9.1.3. Mathematical formulation

2.3.9.2. Mutual Information based scores
2.3.9.2.1. Advantages
2.3.9.2.2. Drawbacks
2.3.9.2.3. Mathematical formulation

2.3.9.3. Homogeneity, completeness and V-measure
2.3.9.3.1. Advantages
2.3.9.3.2. Drawbacks
2.3.9.3.3. Mathematical formulation

2.3.9.4. Fowlkes-Mallows scores
2.3.9.4.1. Advantages
2.3.9.4.2. Drawbacks

2.3.9.5. Silhouette Coefficient
2.3.9.5.1. Advantages
2.3.9.5.2. Drawbacks

2.3.9.6. Calinski-Harabaz Index
2.3.9.6.1. Advantages
2.3.9.6.2. Drawbacks

2.3.9.7. Davies-Bouldin Index
2.3.9.7.1. Advantages
2.3.9.7.2. Drawbacks

2.3.9.8. Contingency Matrix
2.3.9.8.1. Advantages
2.3.9.8.2. Drawbacks

2.4. Biclustering
2.4.1. Spectral Co-Clustering
2.4.1.1. Mathematical formulation

2.4.2. Spectral Biclustering
2.4.2.1. Mathematical formulation

2.4.3. Biclustering evaluation

2.5. Decomposing signals in components (matrix factorization problems)
2.5.1. Principal component analysis (PCA)
2.5.1.1. Exact PCA and probabilistic interpretation
2.5.1.2. Incremental PCA
2.5.1.3. PCA using randomized SVD
2.5.1.4. Kernel PCA
2.5.1.5. Sparse principal components analysis (SparsePCA and MiniBatchSparsePCA)

2.5.2. Truncated singular value decomposition and latent semantic analysis
2.5.3. Dictionary Learning
2.5.3.1. Sparse coding with a precomputed dictionary
2.5.3.2. Generic dictionary learning
2.5.3.3. Mini-batch dictionary learning

2.5.4. Factor Analysis
2.5.5. Independent component analysis (ICA)
2.5.6. Non-negative matrix factorization (NMF or NNMF)
2.5.6.1. NMF with the Frobenius norm
2.5.6.2. NMF with a beta-divergence

2.5.7. Latent Dirichlet Allocation (LDA)

2.6. Covariance estimation
2.6.1. Empirical covariance
2.6.2. Shrunk Covariance
2.6.2.1. Basic shrinkage
2.6.2.2. Ledoit-Wolf shrinkage
2.6.2.3. Oracle Approximating Shrinkage

2.6.3. Sparse inverse covariance
2.6.4. Robust Covariance Estimation
2.6.4.1. Minimum Covariance Determinant

2.7. Novelty and Outlier Detection
2.7.1. Overview of outlier detection methods
2.7.2. Novelty Detection
2.7.3. Outlier Detection
2.7.3.1. Fitting an elliptic envelope
2.7.3.2. Isolation Forest
2.7.3.3. Local Outlier Factor

2.7.4. Novelty detection with Local Outlier Factor

2.8. Density Estimation
2.8.1. Density Estimation: Histograms
2.8.2. Kernel Density Estimation

2.9. Neural network models (unsupervised)
2.9.1. Restricted Boltzmann machines
2.9.1.1. Graphical model and parametrization
2.9.1.2. Bernoulli Restricted Boltzmann machines
2.9.1.3. Stochastic Maximum Likelihood learning

3. Model selection and evaluation
3.1. Cross-validation: evaluating estimator performance
3.1.1. Computing cross-validated metrics
3.1.1.1. The cross_validate function and multiple metric evaluation
3.1.1.2. Obtaining predictions by cross-validation

3.1.2. Cross validation iterators
3.1.2.1. Cross-validation iterators for i.i.d. data
3.1.2.1.1. K-fold
3.1.2.1.2. Repeated K-Fold
3.1.2.1.3. Leave One Out (LOO)
3.1.2.1.4. Leave P Out (LPO)
3.1.2.1.5. Random permutations cross-validation a.k.a. Shuffle & Split

3.1.2.2. Cross-validation iterators with stratification based on class labels.
3.1.2.2.1. Stratified k-fold
3.1.2.2.2. Stratified Shuffle Split

3.1.2.3. Cross-validation iterators for grouped data.
3.1.2.3.1. Group k-fold
3.1.2.3.2. Leave One Group Out
3.1.2.3.3. Leave P Groups Out
3.1.2.3.4. Group Shuffle Split

3.1.2.4. Predefined Fold-Splits / Validation-Sets
3.1.2.5. Cross validation of time series data
3.1.2.5.1. Time Series Split

3.1.3. A note on shuffling
3.1.4. Cross validation and model selection

3.2. Tuning the hyper-parameters of an estimator
3.2.1. Exhaustive Grid Search
3.2.2. Randomized Parameter Optimization
3.2.3. Tips for parameter search
3.2.3.1. Specifying an objective metric
3.2.3.2. Specifying multiple metrics for evaluation
3.2.3.3. Composite estimators and parameter spaces
3.2.3.4. Model selection: development and evaluation
3.2.3.5. Parallelism
3.2.3.6. Robustness to failure

3.2.4. Alternatives to brute force parameter search
3.2.4.1. Model specific cross-validation
3.2.4.1.1. sklearn.linear_model.ElasticNetCV
3.2.4.1.2. sklearn.linear_model.LarsCV
3.2.4.1.3. sklearn.linear_model.LassoCV
3.2.4.1.3.1. Examples using sklearn.linear_model.LassoCV

3.2.4.1.4. sklearn.linear_model.LassoLarsCV
3.2.4.1.4.1. Examples using sklearn.linear_model.LassoLarsCV

3.2.4.1.5. sklearn.linear_model.LogisticRegressionCV
3.2.4.1.6. sklearn.linear_model.MultiTaskElasticNetCV
3.2.4.1.7. sklearn.linear_model.MultiTaskLassoCV
3.2.4.1.8. sklearn.linear_model.OrthogonalMatchingPursuitCV
3.2.4.1.8.1. Examples using sklearn.linear_model.OrthogonalMatchingPursuitCV

3.2.4.1.9. sklearn.linear_model.RidgeCV
3.2.4.1.9.1. Examples using sklearn.linear_model.RidgeCV

3.2.4.1.10. sklearn.linear_model.RidgeClassifierCV

3.2.4.2. Information Criterion
3.2.4.2.1. sklearn.linear_model.LassoLarsIC
3.2.4.2.1.1. Examples using sklearn.linear_model.LassoLarsIC

3.2.4.3. Out of Bag Estimates
3.2.4.3.1. sklearn.ensemble.RandomForestClassifier
3.2.4.3.1.1. Examples using sklearn.ensemble.RandomForestClassifier

3.2.4.3.2. sklearn.ensemble.RandomForestRegressor
3.2.4.3.2.1. Examples using sklearn.ensemble.RandomForestRegressor

3.2.4.3.3. sklearn.ensemble.ExtraTreesClassifier
3.2.4.3.3.1. Examples using sklearn.ensemble.ExtraTreesClassifier

3.2.4.3.4. sklearn.ensemble.ExtraTreesRegressor
3.2.4.3.4.1. Examples using sklearn.ensemble.ExtraTreesRegressor

3.2.4.3.5. sklearn.ensemble.GradientBoostingClassifier
3.2.4.3.5.1. Examples using sklearn.ensemble.GradientBoostingClassifier

3.2.4.3.6. sklearn.ensemble.GradientBoostingRegressor
3.2.4.3.6.1. Examples using sklearn.ensemble.GradientBoostingRegressor

3.3. Model evaluation: quantifying the quality of predictions
3.3.1. The scoring parameter: defining model evaluation rules
3.3.1.1. Common cases: predefined values
3.3.1.2. Defining your scoring strategy from metric functions
3.3.1.3. Implementing your own scoring object
3.3.1.4. Using multiple metric evaluation

3.3.2. Classification metrics
3.3.2.1. From binary to multiclass and multilabel
3.3.2.2. Accuracy score
3.3.2.3. Balanced accuracy score
3.3.2.4. Cohen's kappa
3.3.2.5. Confusion matrix
3.3.2.6. Classification report
3.3.2.7. Hamming loss
3.3.2.8. Jaccard similarity coefficient score
3.3.2.9. Precision, recall and F-measures
3.3.2.9.1. Binary classification
3.3.2.9.2. Multiclass and multilabel classification

3.3.2.10. Hinge loss
3.3.2.11. Log loss
3.3.2.12. Matthews correlation coefficient
3.3.2.13. Receiver operating characteristic (ROC)
3.3.2.14. Zero one loss
3.3.2.15. Brier score loss

3.3.3. Multilabel ranking metrics
3.3.3.1. Coverage error
3.3.3.2. Label ranking average precision
3.3.3.3. Ranking loss

3.3.4. Regression metrics
3.3.4.1. Explained variance score
3.3.4.2. Mean absolute error
3.3.4.3. Mean squared error
3.3.4.4. Mean squared logarithmic error
3.3.4.5. Median absolute error
3.3.4.6. R^2 score, the coefficient of determination

3.3.5. Clustering metrics
3.3.6. Dummy estimators

3.4. Model persistence
3.4.1. Persistence example
3.4.2. Security & maintainability limitations

3.5. Validation curves: plotting scores to evaluate models
3.5.1. Validation curve
3.5.2. Learning curve

4. Dataset transformations
4.1. Pipelines and composite estimators
4.1.1. Pipeline: chaining estimators
4.1.1.1. Usage
4.1.1.2. Notes
4.1.1.3. Caching transformers: avoid repeated computation

4.1.2. Transforming target in regression
4.1.3. FeatureUnion: composite feature spaces
4.1.3.1. Usage

4.1.4. ColumnTransformer for heterogeneous data

4.2. Feature extraction
4.2.1. Loading features from dicts
4.2.2. Feature hashing
4.2.2.1. Implementation details

4.2.3. Text feature extraction
4.2.3.1. The Bag of Words representation
4.2.3.2. Sparsity
4.2.3.3. Common Vectorizer usage
4.2.3.3.1. Using stop words