DAT630
Classification and Clustering Evaluation

28/09/2016

Krisztian Balog | University of Stavanger

Classification Evaluation

Binary Classification

- Confusion matrix

		Predicted class	
		Positive	Negative
Actual class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Measures

- **Accuracy**
  - Fraction of correct predictions

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision**
  - Fraction of positive records among those that are classified as positive

$$P = \frac{TP}{TP + FP}$$

- **Recall**
  - Fraction of positive examples correctly predicted

$$R = \frac{TP}{TP + FN}$$

Measures

- **F1-measure** (or F1-score)
  - Harmonic mean between precision and recall
    - The relative contribution of precision and recall to the F1-score are equal

$$F1 = \frac{2RP}{R + P}$$

Multiclass Classification

- Measures: Precison, Recall, F1
- Two averaging methods
  - Micro-averaging
    - Equal weight to each instance
  - Macro-averaging
    - Equal weight to each category

Multiclass Classification

- **Micro-average method**
  - Sum up the individual TPs, FPs, TNs, FNs and compute precision and recall
  - F1-score will be the harmonic mean of precision and recall
  - "Each instance is equally important"

$$P = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FP_i)} \qquad R = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FN_i)}$$

- M is the number of categories

Multiclass Classification

- **Macro-average method**
  - Consider the confusion matrix for each class to compute the measures (precision, recall, F1-score) for the given class
  - Take the average of these values to get overall (macro-averaged) precision, recall, F1-score
  - "Each class is equally important"
  - Class imbalance is not taken into account
    - Influenced more by the classifier’s performance on rare categories

Example

- Compute micro- and macro-averaged precision, recall, and F1-score from the following classification results

True class	Predicted class
0	0
1	2
2	1
0	0
2	1
1	2
1	0
2	2
1	2

Confusion matrices

class 0		Predicted	
		0	not 0
Actual	0	2	0
	not 0	1	6

class 1		Predicted	
		1	not 1
Actual	1	0	4
	not 1	2	3

class 2		Predicted	
		2	not 2
Actual	2	1	2
	not 2	3	3

Micro-averaging

combined		Predicted	
		C	not C
Actual	C	3	6
	not C	6	12

$$P = \frac{3}{3+6} = \frac{1}{3}$$

$$R = \frac{3}{3+6} = \frac{1}{3}$$

$$F1 = \frac{2 \cdot \frac{1}{3} \cdot \frac{1}{3}}{\frac{1}{3} + \frac{1}{3}} = \frac{1}{3}$$

Macro-averaging

class	P	R	F1
0	2/3	1	4/5
1	0	0	0
2	1/4	1/3	2/7
avg	11/36 =0.305	4/9 =0.444	38/105 =0.361

Classification Evaluation Using scikit-learn

- See code on GitHub

Clustering Evaluation

Types of Evaluation

- Unsupervised
  - Measuring the goodness of a clustering structure without respect to external information ("ground truth")
- Supervised
  - Measuring how well clustering matches externally supplied class labels ("ground truth")
- Relative
  - Compares two different clusterings

Unsupervised Evaluation

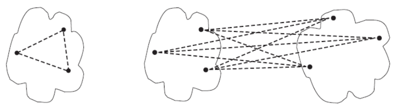
- Cohesion and separation
- Graph-based vs. prototype-based views

$$overall\ validity = \sum_{i=1}^{K} w_i \cdot validity(C_i)$$

cluster weight (can be set to 1)

The *validity* function can be
- *cohesion* (higher values are better) or
- *separation* (lower values are better) or
- some combination of them

Graph-based view



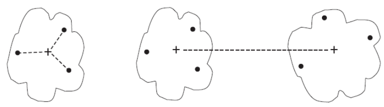
(a) Cohesion. (b) Separation.

$$cohesion(C_i) = \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_i} proximity(\mathbf{x}, \mathbf{y})$$

$$separation(C_i, C_j) = \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} proximity(\mathbf{x}, \mathbf{y})$$

Proximity can be any similarity function

Prototype-base view



(a) Cohesion. (b) Separation.

$$cohesion(C_i) = \sum_{\mathbf{x} \in C_i} proximity(\mathbf{x}, \mathbf{c}_i)$$

$$separation(C_i, C_j) = proximity(\mathbf{c}_i, \mathbf{c}_j)$$

Supervised Evaluation

- We have external label information ("ground truth")
- **Purity**
  - Analogous to precision; the extent to which a cluster contains objects of a single class
- **Inverse purity**
  - Focuses on recall; rewards a clustering that gathers more elements of each class into a corresponding single cluster

Purity

$$\text{Purity} = \sum_i \frac{|C_i|}{N} \max_j \text{Precision}(C_i, L_j)$$

- L is the reference (ground truth) clustering
- C is the generated clustering
- N is the number of documents

$$\text{Precision}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Inverse Purity

$$\text{Inv. Purity} = \sum_i \frac{|L_i|}{N} \max_j \text{Precision}(L_i, C_j)$$

- L is the reference (ground truth) clustering
- C is the generated clustering
- N is the number of documents

$$\text{Precision}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Purity vs. Inverse Purity

- **Purity** penalizes the noise in a cluster, but it does not reward grouping items from the same category together
  - By assigning each document to a separate cluster, we reach trivially a maximum purity value
- **Inverse Purity** rewards grouping items together, but it does not penalize mixing items from different categories
  - We can reach a maximum value for Inverse purity by making a single cluster with all documents

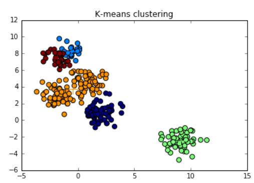
F-Measure

- More robust metric by combining the concepts of Purity and Inverse Purity

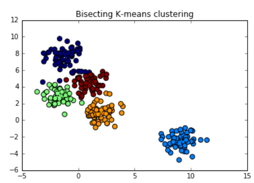
$$F = \frac{1}{0.5\frac{1}{\text{Purity}} + 0.5\frac{1}{\text{Inv. Purity}}}$$

Relative Evaluation

- E.g., comparing two K-means clusterings in terms of SSE



SSE = 376.44



SSE = 304.79