Validating Validation Measures for Clustering and Classification

Martijn Gösgens

This seminar

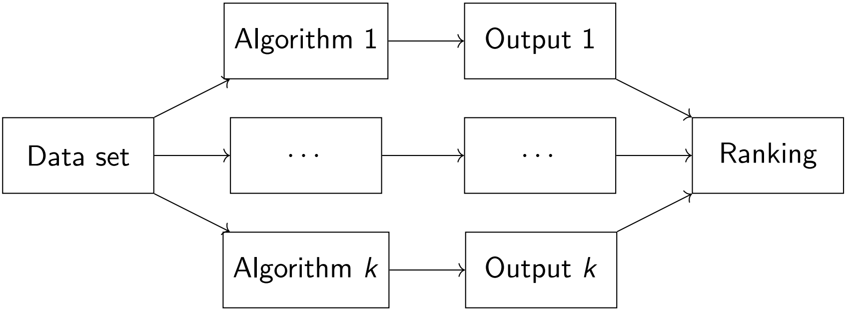
Two papers:

- MG, A. Zhiyanov, A. Tikhonov, & L. Prokhorenkova. (2021). *Good classification measures and how to find them*. Advances in neural information processing systems (NeurIPS).
- MG, A. Tikhonov, & L. Prokhorenkova. (2021). *Systematic analysis of cluster similarity indices: How to validate validation measures*. In International Conference on Machine Learning (ICML).

The general problem

- Need to perform *some task*
- There exist many different algorithms for this task
- How to decide which one is best?

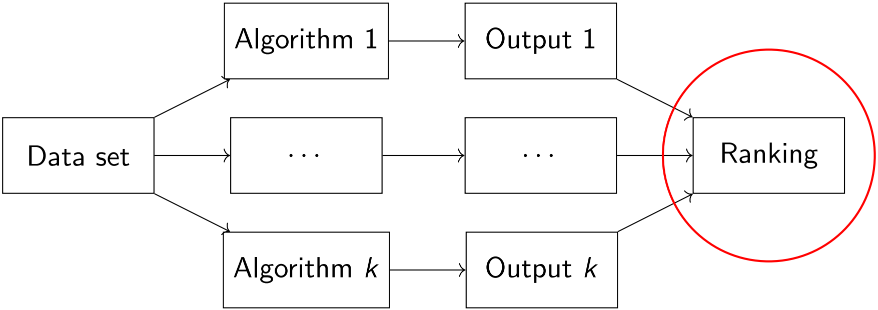
Approach: perform *benchmarking experiment*



The general problem

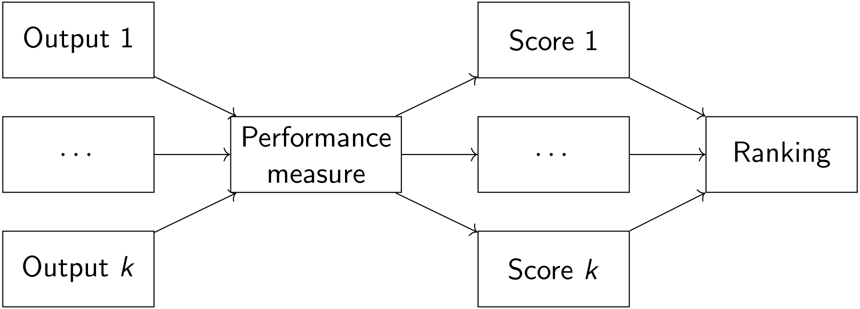
- Need to perform *some task*
- There exist many different algorithms for this task
- How to decide which one is best?

Approach: perform *benchmarking experiment*



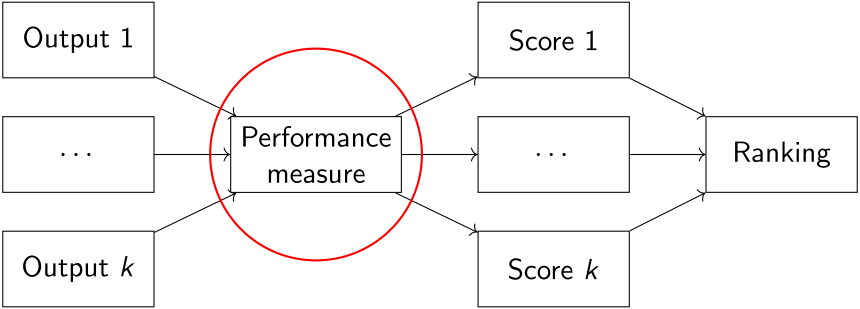
Performance measures

Compare outputs to desired output or *ground truth*.



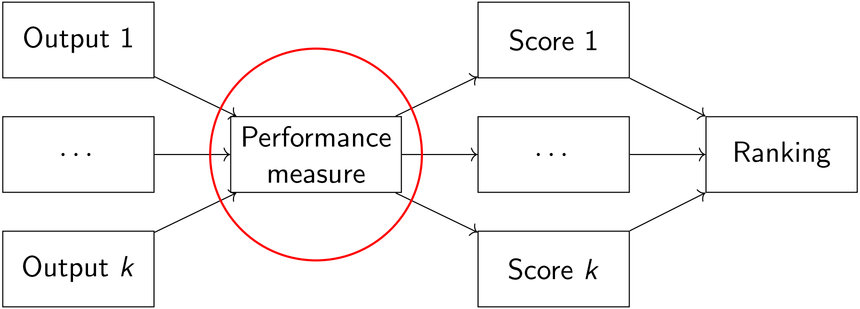
Performance measures

Compare outputs to desired output or *ground truth*.



Performance measures

Compare outputs to desired output or *ground truth*.



Obtaining data with ground truth is a problem of its own, which we will ignore today.

Classification and clustering

- Validation measures for two common ML tasks:
  - Classification: assign each object to one of a set of predefined classes.
  - Clustering: group similar objects together without predefined classes.

Classification and clustering

- Validation measures for two common ML tasks:
  - Classification: assign each object to one of a set of predefined classes.
  - Clustering: group similar objects together without predefined classes.
- Classification yields a *labeling*, clustering a *clustering* (partition of the data points).

Classification

- Input: set of objects with data (features, relations)
- Output: a *labeling* (assignment of each object to one of a set of predefined classes)

Classification

- Input: set of objects with data (features, relations)
- Output: a *labeling* (assignment of each object to one of a set of predefined classes)
- Example applications:
  - Spam detection
  - Handwritten digit recognition
  - Medical diagnosis

Classification

- Input: set of objects with data (features, relations)
- Output: a *labeling* (assignment of each object to one of a set of predefined classes)
- Example applications:
  - Spam detection
  - Handwritten digit recognition
  - Medical diagnosis
- Classification algorithms are often *trained* on some example data with known labels (ground truth). This is known as *supervised learning*.

Classification

- Input: set of objects with data (features, relations)
- Output: a *labeling* (assignment of each object to one of a set of predefined classes)
- Example applications:
  - Spam detection
  - Handwritten digit recognition
  - Medical diagnosis
- Classification algorithms are often *trained* on some example data with known labels (ground truth). This is known as *supervised learning*.
- Many methods: neural networks, decision trees, logistic regression...

Clustering

Definition: partitioning a set of objects into *meaningful* groups.

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)
- Example applications:

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)
- Example applications:
  - Social network analysis

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)
- Example applications:
  - Social network analysis
  - Exploratory data analysis

Clustering

Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

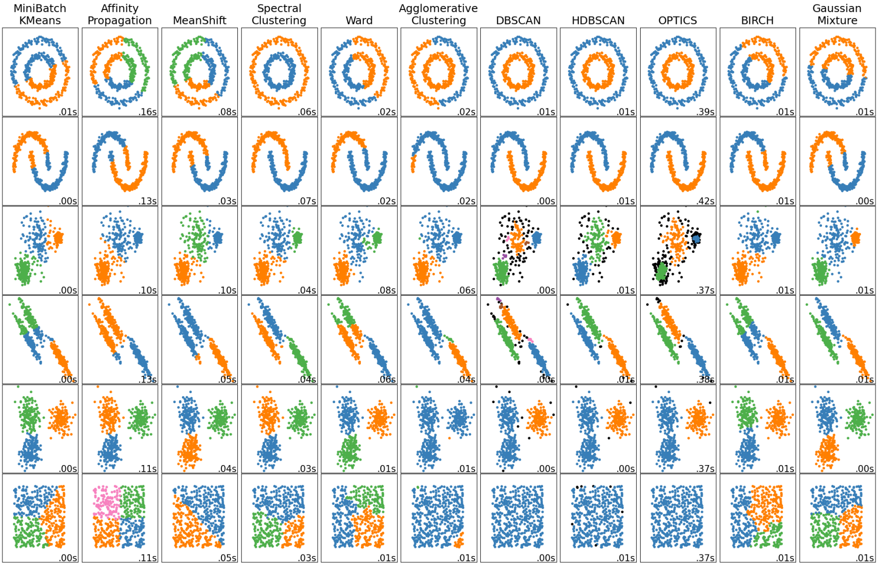
- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)
- Example applications:
  - Social network analysis
  - Exploratory data analysis
  - News aggregation

Clustering

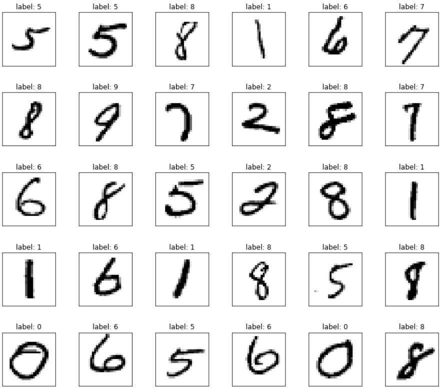
Definition: partitioning a set of objects into *meaningful* groups.
(Intentionally vague)

- Input: set of objects with data (features, relations)
- Output: a *clustering* (partition of the data set into groups of similar objects)
- Example applications:
  - Social network analysis
  - Exploratory data analysis
  - News aggregation
- Clustering is an *unsupervised learning* problem: no training phase.

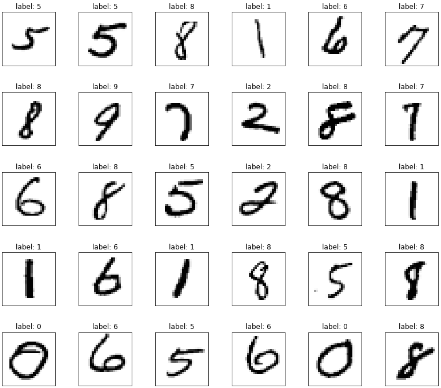
Clustering algorithms



Clustering vs. Classification: MNIST dataset

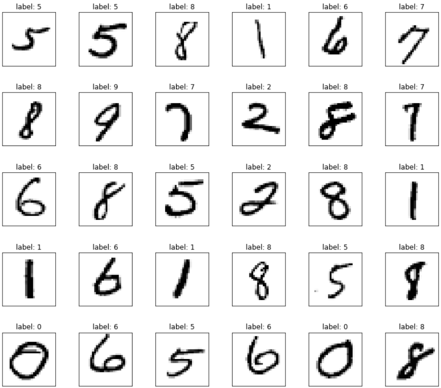


Clustering vs. Classification: MNIST dataset



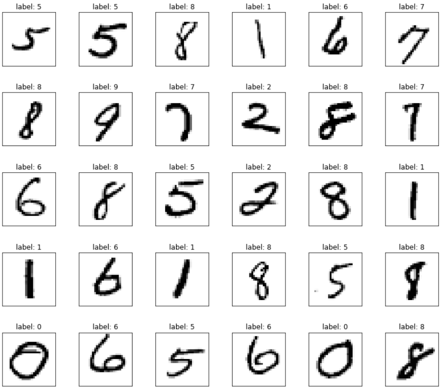
- A classification algorithm would predict which number is written,

Clustering vs. Classification: MNIST dataset



- A classification algorithm would predict which number is written,
- A clustering algorithm would group similar symbols together.

Clustering vs. Classification: MNIST dataset



- A classification algorithm would predict which number is written,
- A clustering algorithm would group similar symbols together.
- For classification, we need to first define the classes and provide labeled training data.

Overview

- Lecture 1: Validating Validation Measures for Classification.
- Lecture 2: Validating Validation Measures for Clustering.

Part I: Validating Validation Measures for Classification

Rain prediction example

Let A denote the true labeling and let B_1, B_2 denote two labelings obtained by different algorithms. We say that two measures M_1, M_2 are *inconsistent* if $M_1(A, B_1) > M_1(A, B_2)$ but $M_2(A, B_1) < M_2(A, B_2)$ (or the other way around).

Table 4: Inconsistency of binary measures for rain prediction, %

	Acc	BA	F_1	κ	CE	GM_1	CC	SBA
Acc	—	96.5	41.0	37.5	3.1	38.7	44.3	55.9
BA	96.5	—	55.6	58.9	99.7	57.7	52.0	40.4
F_1	41.0	55.6	—	3.3	44.2	2.2	3.4	15.0
κ	37.5	58.9	3.3	—	40.7	1.1	6.7	18.3
CE	3.1	99.7	44.2	40.7	—	41.9	47.5	59.1
GM_1	38.7	57.7	2.2	1.1	41.9	—	5.5	17.1
CC	44.3	52.0	3.4	6.7	47.5	5.5	—	11.4
SBA	55.9	40.4	15.0	18.3	59.1	17.1	11.4	—

Confusion matrices

Given labelings A, B of n data points into m classes, we define the *confusion matrix* $\mathcal{C} = (c_{ij})_{i,j=0}^{m-1}$ by

$$c_{ij} = |\{x \mid A(x) = i, B(x) = j\}|.$$

Confusion matrices

Given labelings A, B of n data points into m classes, we define the *confusion matrix* $\mathcal{C} = (c_{ij})_{i,j=0}^{m-1}$ by

$$c_{ij} = |\{x \mid A(x) = i, B(x) = j\}|.$$

For binary classification:

- c_{11}: true positives
- c_{10}: false negatives
- c_{01}: false positives
- c_{00}: true negatives

Confusion matrices

Given labelings A, B of n data points into m classes, we define the *confusion matrix* $\mathcal{C} = (c_{ij})_{i,j=0}^{m-1}$ by

$$c_{ij} = |\{x \mid A(x) = i, B(x) = j\}|.$$

For binary classification:

- c_{11}: true positives
- c_{10}: false negatives
- c_{01}: false positives
- c_{00}: true negatives

(but I will try to avoid these terms since I find them confusing)

Confusion matrices

Given labelings A, B of n data points into m classes, we define the *confusion matrix* $\mathcal{C} = (c_{ij})_{i,j=0}^{m-1}$ by

$$c_{ij} = |\{x \mid A(x) = i, B(x) = j\}|.$$

For binary classification:

- c_{11}: true positives
- c_{10}: false negatives
- c_{01}: false positives
- c_{00}: true negatives

(but I will try to avoid these terms since I find them confusing)

Also, $a_i = \sum_j c_{ij}$, $b_j = \sum_i c_{ij}$, $n = \sum_{i,j} c_{ij}$.

Examples of binary classification measures

- *Accuracy* is the fraction of correctly labeled data points: $\mathrm{Acc}(A, B) = (c_{11} + c_{00})/n$.

Examples of binary classification measures

- *Accuracy* is the fraction of correctly labeled data points: $\mathrm{Acc}(A, B) = (c_{11} + c_{00})/n$.
- *Recall* is the fraction of correctly identified true positives: $\mathrm{Recall}(A, B) = c_{11}/a_1$.

Examples of binary classification measures

- *Accuracy* is the fraction of correctly labeled data points: $\mathrm{Acc}(A, B) = (c_{11} + c_{00})/n$.
- *Recall* is the fraction of correctly identified true positives: $\mathrm{Recall}(A, B) = c_{11}/a_1$.
- *Precision* is the fraction of predicted positives that are true: $\mathrm{Precision}(A, B) = c_{11}/b_1$.

Examples of binary classification measures

- *Accuracy* is the fraction of correctly labeled data points: $\mathrm{Acc}(A, B) = (c_{11} + c_{00})/n$.
- *Recall* is the fraction of correctly identified true positives: $\mathrm{Recall}(A, B) = c_{11}/a_1$.
- *Precision* is the fraction of predicted positives that are true: $\mathrm{Precision}(A, B) = c_{11}/b_1$.
- The F_1 measure is the harmonic mean between recall and precision:

$$F_1(A, B) = \frac{2}{\frac{a_1}{c_{11}} + \frac{b_1}{c_{11}}} = \frac{2c_{11}}{a_1 + b_1}.$$

Examples of binary classification measures

- *Accuracy* is the fraction of correctly labeled data points: $\mathrm{Acc}(A, B) = (c_{11} + c_{00})/n$.
- *Recall* is the fraction of correctly identified true positives: $\mathrm{Recall}(A, B) = c_{11}/a_1$.
- *Precision* is the fraction of predicted positives that are true: $\mathrm{Precision}(A, B) = c_{11}/b_1$.
- The F_1 measure is the harmonic mean between recall and precision:

$$F_1(A, B) = \frac{2}{\frac{a_1}{c_{11}} + \frac{b_1}{c_{11}}} = \frac{2c_{11}}{a_1 + b_1}.$$

- *Matthew's Correlation Coefficient* is the (Pearson) correlation between the indicators:

$$\mathrm{CC}(A, B) = \frac{nc_{11} - a_1 b_1}{\sqrt{a_1(n - a_1)b_1(n - b_1)}}.$$

Exercises

Recall

$$F_1(A,B) = \frac{2c_{11}}{a_1 + b_1}, \quad \text{and} \quad \mathrm{CC}(A,B) = \frac{nc_{11} - a_1 b_1}{\sqrt{a_1(n-a_1)b_1(n-b_1)}}.$$

1. Consider labelings A, B, where B labels all data points to 1 (i.e., $b_1 = n$). Express $F_1(A,B)$ in terms of n and a_1.
2. Define F_1' as the *arithmetic* mean of recall and precision. Compute $F_1'(A,B)$ for the same labelings A, B.
3. Can you explain why the *harmonic* mean rather than the *arithmetic* mean is chosen?
4. Consider a fixed labeling A with a_1 positives and consider a random labeling B with b_1 positives. Express $\mathbb{E}[F_1(A,B)]$ in terms of a_1, b_1, n.
5. Consider the same A, B as above. Compute $\mathbb{E}[\mathrm{CC}(A,B)]$.

Answers

1. $F_1(A, B) = \frac{2a_1}{a_1 + n}$.
2. $F_1'(A, B) = \frac{1}{2} + \frac{a_1}{2n}$.
3. Harmonic mean penalizes extreme values more than arithmetic mean. If either recall or precision is low, F_1 will be low as well.
4. $\mathbb{E}[F_1(A, B)] = \frac{2a_1 b_1}{n(a_1 + b_1)}$.
5. $\mathbb{E}[\mathrm{CC}(A, B)] = 0$.

More measures

	Binary	Multiclass
F-measure (F_β)	$\frac{(1+\beta^2)\cdot c_{11}}{(1+\beta^2)\cdot c_{11}+\beta^2\cdot c_{10}+c_{01}}$	micro / macro / weighted
Jaccard (J)	$\frac{c_{11}}{c_{11}+c_{10}+c_{01}}$	micro / macro / weighted
Matthews Coefficient (CC)	$\frac{c_{11}c_{00}-c_{01}c_{10}}{\sqrt{b_1\cdot a_1\cdot b_0\cdot a_0}}$	$\frac{n\sum_{i=0}^{m-1}c_{ii}-\sum_{i=0}^{m-1}b_i a_i}{\sqrt{\left(n^2-\sum_{i=0}^{m-1}b_i^2\right)\left(n^2-\sum_{i=0}^{m-1}a_i^2\right)}}$
Accuracy (Acc)	$\frac{\sum_{i=0}^{m-1}c_{ii}}{n}$	
Balanced Accuracy (BA)	$\frac{1}{m}\sum_{i=0}^{m-1}\frac{c_{ii}}{a_i}$	
Cohen's Kappa (κ)	$\frac{n\sum_{i=0}^{m-1}c_{ii}-\sum_{i=0}^{m-1}a_i b_i}{n^2-\sum_{i=0}^{m-1}a_i b_i}$	
Confusion Entropy (CE)	$-\frac{1}{2n}\sum_{i,j:i\neq j}\left(c_{ji}\log_{2m-2}\frac{c_{ji}}{a_j+b_j}+c_{ij}\log_{2m-2}\frac{c_{ij}}{a_j+b_j}\right)$	
Symmetric Balanced Accuracy (SBA)	$\frac{1}{2m}\sum_{i=0}^{m-1}\left(\frac{c_{ii}}{a_i}+\frac{c_{ii}}{b_i}\right)$	
Generalized Means (GM)	$\frac{n\,c_{11}-a_1 b_1}{\sqrt[r]{\frac{1}{2}\left(a_1^r a_0^r+b_1^r b_0^r\right)}}$	micro / macro / weighted
Correlation Distance (CD)	$\frac{1}{\pi}\arccos(\mathrm{CC})$	

F1 vs CC

The advantages of the **Matthews correlation** coefficient (MCC) over F1 score and accuracy in binary classification evaluation

D Chicco, G Jurman - BMC genomics, 2020 - Springer

… We believe that the **Matthews correlation** coefficient should be preferred to accuracy and F 1 score in evaluating binary classification tasks by all scientific communities. …

☆ Save 99 Cite Cited by 6584 Related articles All 19 versions Import into BibTeX

- Makes the argument that F1 only uses 3 out of four confusion matrix entries, ignoring c_{00} (true negatives).

F1 vs CC

The advantages of the **Matthews correlation** coefficient (MCC) over F1 score and accuracy in binary classification evaluation

D Chicco, G Jurman - BMC genomics, 2020 - Springer

… We believe that the **Matthews correlation** coefficient should be preferred to accuracy and F 1 score in evaluating binary classification tasks by all scientific communities. …

☆ Save 99 Cite Cited by 6584 Related articles All 19 versions Import into BibTeX

- Makes the argument that F1 only uses 3 out of four confusion matrix entries, ignoring c_{00} (true negatives).
- Also performs experiments that show that F1 and accuracy can be misleading, especially with imbalanced data.

Experiments vs theory

Practical examples and experiments are valuable, but context-dependent.

Experiments vs theory

Practical examples and experiments are valuable, but context-dependent.

Instead, we formalize theoretical properties and provide mathematical proofs or counter-examples for each measure.

Experiments vs theory

Practical examples and experiments are valuable, but context-dependent.

Instead, we formalize theoretical properties and provide mathematical proofs or counter-examples for each measure.

Measure	Max	Min	CSym	Sym	Dist	Mon	SMon	CB	ACB
F_1 (binary)	✓	✗	✗	✓	✗	✓	✗	✗	✗
J (binary)	✓	✗	✗	✓	✓	✓	✗	✗	✗
CC	✓	✓/✗	✓	✓	✗	✓/✗	✓/✗	✓	✓
Acc	✓	✓	✓	✓	✓	✓	✓	✗	✗
BA	✓	✓	✓	✗	✗	✓	✓	✓	✓
κ	✓	✗	✓	✓	✗	✓/✗	✗	✓	✓
CE	✓	✗	✓	✓	✗	✗	✗	✗	✗
SBA	✓	✓	✓	✓	✗	✓	✓	✓	✓
GM (binary)	✓	✓	✓	✓	✗	✓	✓	✓	✓
CD	✓	✓/✗	✓	✓	✓	✓/✗	✓/✗	✗	✓

Properties of validation measures and averagings, ✓/✗ indicates that property is satisfied only in the binary case.

Before defining these properties...

Can you come up with desirable properties for validation measures?

Before defining these properties...

Can you come up with desirable properties for validation measures?

We start with defining some properties that are easy to check.

Maximum agreement

It's helpful if we can see from $M(A, B)$, whether $A = B$.

Definition

A measure M satisfies *maximal agreement* if there exists a constant c_{max} such that for all $\mathcal{C}$, $M(\mathcal{C}) \leq c_{\mathsf{max}}$ with equality iff $\mathcal{C}$ is diagonal.

Maximum agreement

It's helpful if we can see from $M(A, B)$, whether $A = B$.

Definition

A measure M satisfies *maximal agreement* if there exists a constant c_{max} such that for all $\mathcal{C}$, $M(\mathcal{C}) \leq c_{\text{max}}$ with equality iff $\mathcal{C}$ is diagonal.

For example,

- $c_{\text{max}} = 1$ for accuracy, F_1, CC and κ,

Maximum agreement

It's helpful if we can see from $M(A, B)$, whether $A = B$.

Definition

A measure M satisfies *maximal agreement* if there exists a constant $c_{\max}$ such that for all $\mathcal{C}$, $M(\mathcal{C}) \leq c_{\max}$ with equality iff $\mathcal{C}$ is diagonal.

For example,

- $c_{\max} = 1$ for accuracy, F_1, CC and κ,
- but not Recall $= \frac{c_{11}}{a_1}$ (e.g., $b_1 = n$).

Minimum agreement

For interpretability, it helps if a measure assigns values to a fixed interval, i.e., $M(A, B) \in [c_{\min}, c_{\max}]$ for all A, B, where both $c_{\min}, c_{\max}$ are attainable for fixed A.

Minimum agreement

For interpretability, it helps if a measure assigns values to a fixed interval, i.e., $M(A, B) \in [c_{\min}, c_{\max}]$ for all A, B, where both $c_{\min}, c_{\max}$ are attainable for fixed A.

Definition

A measure M satisfies *minimal agreement* if there exists a constant $c_{\min}$ such that for all $\mathcal{C}$, $M(\mathcal{C}) \geq c_{\min}$ with equality iff the diagonal of $\mathcal{C}$ is zero, i.e., $c_{ii} = 0$ for all i.

Minimum agreement

For interpretability, it helps if a measure assigns values to a fixed interval, i.e., $M(A, B) \in [c_{\min}, c_{\max}]$ for all A, B, where both $c_{\min}, c_{\max}$ are attainable for fixed A.

Definition

A measure M satisfies *minimal agreement* if there exists a constant $c_{\min}$ such that for all $\mathcal{C}$, $M(\mathcal{C}) \geq c_{\min}$ with equality iff the diagonal of $\mathcal{C}$ is zero, i.e., $c_{ii} = 0$ for all i.

- Acc satisfies this with $c_{\min} = 0$.
- CC satisfies this with $c_{\min} = -1$.
- F_1 does not satisfy this.

Symmetry

Definition

A measure M is *symmetric* if $M(A, B) = M(B, A)$ (i.e. $M(\mathcal{C}) = M(\mathcal{C}^\top)$) holds for all $\mathcal{C}$.

Symmetry

Definition

A measure M is *symmetric* if $M(A, B) = M(B, A)$ (i.e. $M(\mathcal{C}) = M(\mathcal{C}^\top)$) holds for all $\mathcal{C}$.

- Balanced accuracy $\mathrm{BA}(A, B) = \frac{1}{m} \sum_{i=0}^{m-1} \frac{c_{ii}}{a_i}$ is not symmetric. Most others are.

Symmetry

Definition

A measure M is *symmetric* if $M(A, B) = M(B, A)$ (i.e. $M(\mathcal{C}) = M(\mathcal{C}^\top)$) holds for all $\mathcal{C}$.

- Balanced accuracy $\mathrm{BA}(A, B) = \frac{1}{m} \sum_{i=0}^{m-1} \frac{c_{ii}}{a_i}$ is not symmetric. Most others are.
- The labelings A, B often take *different roles*, which may justify asymmetry. However, it is not straightforward what such asymmetry should look like.

Class symmetry

Definition

A measure M is *class-symmetric* if, for any permutation π of the classes $\{1, \ldots, m\}$ and any confusion matrix $\mathcal{C}$, $M(\mathcal{C}) = M(\tilde{\mathcal{C}})$ holds, where $\tilde{\mathcal{C}}$ is given by $\tilde{c}_{ij} = c_{\pi(i),\pi(j)}$.

Class symmetry

Definition

A measure M is *class-symmetric* if, for any permutation π of the classes $\{1, \ldots, m\}$ and any confusion matrix $\mathcal{C}$, $M(\mathcal{C}) = M(\tilde{\mathcal{C}})$ holds, where $\tilde{\mathcal{C}}$ is given by $\tilde{c}_{ij} = c_{\pi(i),\pi(j)}$.

- Class asymmetry can be justified if the classes have different roles. But again, what should this asymmetry look like?

Class symmetry

Definition

A measure M is *class-symmetric* if, for any permutation π of the classes $\{1, \ldots, m\}$ and any confusion matrix $\mathcal{C}$, $M(\mathcal{C}) = M(\tilde{\mathcal{C}})$ holds, where $\tilde{\mathcal{C}}$ is given by $\tilde{c}_{ij} = c_{\pi(i),\pi(j)}$.

- Class asymmetry can be justified if the classes have different roles. But again, what should this asymmetry look like?
- Most measures are class-symmetric.

Exercises

(for binary classification)

1. Give a counter-example that shows that F_1 does not satisfy minimal agreement.
2. Let A be a binary labeling and let A^C denote its opposite. Calculate $\kappa(A, A^C)$.
3. Is F_1 class-symmetric?

Answers

1. $A = [1, 0, 0], B = [0, 1, 0]$. Then $F_1(A, B) = 0$ but $c_{00} = 1$.
2. We get $c_{11} = c_{00} = 0$ and $b_1 = n - a_1$.

$$\kappa(A, A^C) = \frac{-2a_1(n - a_1)}{n^2 - 2a_1(n - a_1)}.$$

3. No, the inverse is $\frac{2c_{00}}{a_0 + b_0} = 2\frac{n - a_1 - b_1 + c_{11}}{2n - a_1 - b_1}$.

Distance

Recall that a function d is a *distance* whenever it is symmetric, non-negative (with $d(A, B) = 0$ iff $A = B$), and satisfies the triangle inequality

$$d(A, C) \leq d(A, B) + d(B, C).$$

Definition

A measure has the *distance* property if it can be linearly transformed to a metric distance.

Distance

Recall that a function d is a *distance* whenever it is symmetric, non-negative (with $d(A, B) = 0$ iff $A = B$), and satisfies the triangle inequality

$$d(A, C) \leq d(A, B) + d(B, C).$$

Definition

A measure has the *distance* property if it can be linearly transformed to a metric distance.

- $1 - \mathrm{Acc}$ is the *Hamming distance*.

Distance

Recall that a function d is a *distance* whenever it is symmetric, non-negative (with $d(A, B) = 0$ iff $A = B$), and satisfies the triangle inequality

$$d(A, C) \leq d(A, B) + d(B, C).$$

Definition

A measure has the *distance* property if it can be linearly transformed to a metric distance.

- $1 - \mathrm{Acc}$ is the *Hamming distance*.
- A distance interpretation can be useful for theoretical analysis (e.g., deriving performance guarantees).

Distance

Recall that a function d is a *distance* whenever it is symmetric, non-negative (with $d(A, B) = 0$ iff $A = B$), and satisfies the triangle inequality

$$d(A, C) \leq d(A, B) + d(B, C).$$

Definition

A measure has the *distance* property if it can be linearly transformed to a metric distance.

- $1 - \mathrm{Acc}$ is the *Hamming distance*.
- A distance interpretation can be useful for theoretical analysis (e.g., deriving performance guarantees).
- Difficult to check.

Exercises

1. Which of the other properties are necessary for the distance property?

Let us represent a binary labeling A as an n-dimensional binary vector, so that $\langle A, B\rangle = c_{11}$. Let $\mathbf{1}$ denote the all-ones vector.

2. Let $\hat{A}$ be the projection of A onto the surface $\langle x, \mathbf{1}\rangle = 0$. Calculate $\hat{A}$.
3. Calculate the angle between $\hat{A}$ and $\hat{B}$. Express your answer in terms of c_{11}, a_1, b_1, n.
4. Which validation measure do you recognize?

Answers

1. Symmetry and maximal agreement.
2. $\hat{A} = A - \frac{a_1}{n}\mathbf{1}$.
3. $$\angle(\hat{A}, \hat{B}) = \frac{\langle \hat{A}, \hat{B} \rangle}{\sqrt{\langle \hat{A}, \hat{A} \rangle \cdot \langle \hat{B}, \hat{B} \rangle}} = \frac{nc_{11} - a_1 b_1}{\sqrt{a_1(n - a_1)b_1(n - b_1)}}.$$
4. CD (arccos of Matthew's Correlation Coefficient).

Monotonicity

Informally, if we change B to agree more with A, then $M(A, B)$ should increase.

Monotonicity

Informally, if we change B to agree more with A, then $M(A, B)$ should increase.

Definition

A measure M is *monotone* if $M(\mathcal{C}) < M(\tilde{\mathcal{C}})$ for any confusion matrices $\mathcal{C}$ and $\tilde{\mathcal{C}}$ such that $\tilde{\mathcal{C}}$ is obtained from $\mathcal{C}$ by decrementing an off-diagonal entry c_{ab} and incrementing c_{aa} or c_{bb} (and none of the row- or column-sums of $\mathcal{C}$ equal n).

Monotonicity

Informally, if we change B to agree more with A, then $M(A, B)$ should increase.

Definition

A measure M is *monotone* if $M(\mathcal{C}) < M(\tilde{\mathcal{C}})$ for any confusion matrices $\mathcal{C}$ and $\tilde{\mathcal{C}}$ such that $\tilde{\mathcal{C}}$ is obtained from $\mathcal{C}$ by decrementing an off-diagonal entry c_{ab} and incrementing c_{aa} or c_{bb} (and none of the row- or column-sums of $\mathcal{C}$ equal n).

Definition

A measure M is *strongly monotone* if $M(\mathcal{C}) < M(\tilde{\mathcal{C}})$ for any confusion matrices $\mathcal{C}$ and $\tilde{\mathcal{C}}$ such that $\tilde{\mathcal{C}}$ is obtained from $\mathcal{C}$ by either increasing a diagonal entry or decreasing an off-diagonal entry (and none of the row- or column-sums of $\mathcal{C}$ equal n and that $\mathcal{C}$ and $\tilde{\mathcal{C}}$ are not both diagonal or zero-diagonal matrices).

Monotonicity

- Tricky to prove, annoying to find counter-examples.
- F_1 and Jaccard are monotone, but not strongly monotone. (remains constant when changing c_{00})
- κ sometimes even *increases* when increasing off-diagonal entries: $\kappa\left(\begin{smallmatrix}1 & 2\\ 1 & 0\end{smallmatrix}\right) < \kappa\left(\begin{smallmatrix}1 & 3\\ 1 & 0\end{smallmatrix}\right)$.

Constant baseline

Many measures are biased towards certain types of labelings:

- Accuracy is biased towards the majority class.
- F_1 is biased towards labelings with many positives.

Constant baseline

Many measures are biased towards certain types of labelings:

- Accuracy is biased towards the majority class.
- F_1 is biased towards labelings with many positives.

Recall that for fixed A and random B, we have

$$\mathbb{E}[F_1(A, B)] = \frac{2a_1 b_1}{n(a_1 + b_1)},$$

which is increasing in b_1.

Constant baseline

Many measures are biased towards certain types of labelings:

- Accuracy is biased towards the majority class.
- F_1 is biased towards labelings with many positives.

Recall that for fixed A and random B, we have

$$\mathbb{E}[F_1(A, B)] = \frac{2a_1 b_1}{n(a_1 + b_1)},$$

which is increasing in b_1.
Let $B \sim U(b_1, \dots, b_m)$ denote a random labeling with sizes $b_1, \dots, b_m$.

Definition

A measure M has a *constant baseline* if there exists $c_{\text{base}}(m)$ that does not depend on n but may depend on m, such that for any A and non-unary class sizes $b_1, \dots, b_m$, it holds that $\mathbb{E}_{B \sim U(b_1, \dots, b_m)}[M(A, B)] = c_{\text{base}}(m)$.

Constant baseline

- Note that $\mathbb{E}[c_{ij}] = \frac{a_i b_j}{n}$.

Constant baseline

- Note that $\mathbb{E}[c_{ij}] = \frac{a_i b_j}{n}$.
- Easy to prove, but difficult to find counter-examples.

Constant baseline

- Note that $\mathbb{E}[c_{ij}] = \frac{a_i b_j}{n}$.
- Easy to prove, but difficult to find counter-examples.
- CB is particularly important when trying to find the best *threshold* for a classifier.

Constant baseline

- Note that $\mathbb{E}[c_{ij}] = \frac{a_i b_j}{n}$.
- Easy to prove, but difficult to find counter-examples.
- CB is particularly important when trying to find the best *threshold* for a classifier.
- Concretely, rain forecasters optimized for F_1 will predict more rain than rain forecasters optimized for Acc.

Approximate constant baseline

Definition

M has an *approximate constant baseline* if there exists a function $c_{\text{base}}(m)$ that does not depend on n but may depend on m such that for any class sizes $a_1, \ldots, a_m$ and any non-unary $b_1, \ldots, b_m$, $M(\bar{\mathcal{C}}) = c_{\text{base}}(m)$, where $\bar{c}_{ij} = \frac{a_i b_j}{n}$.

Approximate constant baseline

Definition

M has an *approximate constant baseline* if there exists a function $c_{\text{base}}(m)$ that does not depend on n but may depend on m such that for any class sizes $a_1, \ldots, a_m$ and any non-unary $b_1, \ldots, b_m$, $M(\bar{\mathcal{C}}) = c_{\text{base}}(m)$, where $\bar{c}_{ij} = \frac{a_i b_j}{n}$.

- Constant baseline implies approximate constant baseline.

Approximate constant baseline

Definition

M has an *approximate constant baseline* if there exists a function $c_{\text{base}}(m)$ that does not depend on n but may depend on m such that for any class sizes $a_1, \ldots, a_m$ and any non-unary $b_1, \ldots, b_m$, $M(\bar{\mathcal{C}}) = c_{\text{base}}(m)$, where $\bar{c}_{ij} = \frac{a_i b_j}{n}$.

- Constant baseline implies approximate constant baseline.
- CC has a constant baseline, but CD does not (due to the arccos).

Approximate constant baseline

Definition

M has an *approximate constant baseline* if there exists a function $c_{\text{base}}(m)$ that does not depend on n but may depend on m such that for any class sizes $a_1, \ldots, a_m$ and any non-unary $b_1, \ldots, b_m$, $M(\bar{\mathcal{C}}) = c_{\text{base}}(m)$, where $\bar{c}_{ij} = \frac{a_i b_j}{n}$.

- Constant baseline implies approximate constant baseline.
- CC has a constant baseline, but CD does not (due to the arccos).
- CD does have ACB.

The properties

Measure	Max	Min	CSym	Sym	Dist	Mon	SMon	CB	ACB
F_1 (binary)	✓	✗	✗	✓	✗	✓	✗	✗	✗
J (binary)	✓	✗	✗	✓	✓	✓	✗	✗	✗
CC	✓	✓/✗	✓	✓	✗	✓/✗	✓/✗	✓	✓
Acc	✓	✓	✓	✓	✓	✓	✓	✗	✗
BA	✓	✓	✓	✗	✗	✓	✓	✓	✓
κ	✓	✗	✓	✓	✗	✓/✗	✗	✓	✓
CE	✓	✗	✓	✓	✗	✗	✗	✗	✗
SBA	✓	✓	✓	✓	✗	✓	✓	✓	✓
GM (binary)	✓	✓	✓	✓	✗	✓	✓	✓	✓
CD	✓	✓/✗	✓	✓	✓	✓/✗	✓/✗	✗	✓

Properties of validation measures and averagings, ✓/✗ indicates that property is satisfied only in the binary case.

Note that there is no measure that satisfies Dist, Mon and CB.

Impossibility theorem

Theorem

A binary validation measure cannot simultaneously satisfy the distance property, monotonicity, and constant baseline.

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1, 1)$ and $B_2 \sim U(n-2, 2)$. Let $d = c_{\mathsf{max}} - M$.

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1,1)$ and $B_2 \sim U(n-2,2)$. Let $d = c_{\max} - M$.
CB requires $\mathbb{E}[M(A,B_1)] = \mathbb{E}[M(A,B_2)]$, which gives

$$\begin{aligned}
&\frac{1}{n}c_{\max} + \frac{n-1}{n}M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} = \frac{2}{n}M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} + \frac{n-2}{n}M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix} \\
&\quad\Leftrightarrow 2M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} - c_{\max} = (n-1)M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} - (n-2)M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix}. \qquad (1)
\end{aligned}$$

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1,1)$ and $B_2 \sim U(n-2,2)$. Let $d = c_{\max} - M$.
CB requires $\mathbb{E}[M(A,B_1)] = \mathbb{E}[M(A,B_2)]$, which gives

$$\begin{aligned} \frac{1}{n}c_{\max} + \frac{n-1}{n} M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} &= \frac{2}{n} M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} + \frac{n-2}{n} M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix} \\ \Leftrightarrow 2M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} - c_{\max} &= (n-1)M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} - (n-2)M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix}. \qquad (1) \end{aligned}$$

Consider a labeling C with a single positive that does not coincide with the positive of A and a labeling $B = A + C$ (i.e., two positives). Using $d(A,C) \leq d(A,B) + d(B,C)$:

$$c_{\max} - M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} \overset{\text{Dist}}{\leq} 2c_{\max} - M\begin{pmatrix} 1 & 1 \\ 0 & n-2 \end{pmatrix} - M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} \overset{\text{Sym}}{=} 2\left(c_{\max} - M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix}\right),$$

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1,1)$ and $B_2 \sim U(n-2,2)$. Let $d = c_{\max} - M$.
CB requires $\mathbb{E}[M(A,B_1)] = \mathbb{E}[M(A,B_2)]$, which gives

$$\begin{aligned}\frac{1}{n}c_{\max} + \frac{n-1}{n}M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} &= \frac{2}{n}M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} + \frac{n-2}{n}M\begin{pmatrix}0 & 1\\ 2 & n-3\end{pmatrix}\\ \Leftrightarrow 2M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} - c_{\max} &= (n-1)M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} - (n-2)M\begin{pmatrix}0 & 1\\ 2 & n-3\end{pmatrix}. \qquad (1)\end{aligned}$$

Consider a labeling C with a single positive that does not coincide with the positive of A and a labeling $B = A + C$ (i.e., two positives). Using $d(A,C) \leq d(A,B) + d(B,C)$:

$$c_{\max} - M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} \overset{\text{Dist}}{\leq} 2c_{\max} - M\begin{pmatrix}1 & 1\\ 0 & n-2\end{pmatrix} - M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} \overset{\text{Sym}}{=} 2\left(c_{\max} - M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix}\right),$$

This is rewritten to

$$2M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} - c_{\max} \leq M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix}. \qquad (2)$$

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1,1)$ and $B_2 \sim U(n-2,2)$. Let $d = c_{\max} - M$.
CB requires $\mathbb{E}[M(A,B_1)] = \mathbb{E}[M(A,B_2)]$, which gives

$$\frac{1}{n}c_{\max} + \frac{n-1}{n}M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} = \frac{2}{n}M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} + \frac{n-2}{n}M\begin{pmatrix}0 & 1\\ 2 & n-3\end{pmatrix}$$

$$\Leftrightarrow 2M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} - c_{\max} = (n-1)M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} - (n-2)M\begin{pmatrix}0 & 1\\ 2 & n-3\end{pmatrix}. \qquad (1)$$

Consider a labeling C with a single positive that does not coincide with the positive of A and a labeling $B = A + C$ (i.e., two positives). Using $d(A,C) \le d(A,B) + d(B,C)$:

$$c_{\max} - M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} \overset{\text{Dist}}{\le} 2c_{\max} - M\begin{pmatrix}1 & 1\\ 0 & n-2\end{pmatrix} - M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} \overset{\text{Sym}}{=} 2\left(c_{\max} - M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix}\right),$$

This is rewritten to

$$2M\begin{pmatrix}1 & 0\\ 1 & n-2\end{pmatrix} - c_{\max} \le M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix}. \qquad (2)$$

Combining (1) and (2), we obtain

$$(n-1)M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix} - (n-2)M\begin{pmatrix}0 & 1\\ 2 & n-3\end{pmatrix} \le M\begin{pmatrix}0 & 1\\ 1 & n-2\end{pmatrix}.$$

Impossibility proof

Let A have a single positive and $n-1$ negatives. Let $B_1 \sim U(n-1,1)$ and $B_2 \sim U(n-2,2)$. Let $d = c_{\text{max}} - M$.
CB requires $\mathbb{E}[M(A,B_1)] = \mathbb{E}[M(A,B_2)]$, which gives

$$\begin{aligned} \frac{1}{n}c_{\text{max}} + \frac{n-1}{n}M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} &= \frac{2}{n}M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} + \frac{n-2}{n}M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix} \\ \Leftrightarrow 2M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} - c_{\text{max}} &= (n-1)M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} - (n-2)M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix}. \end{aligned} \tag{1}$$

Consider a labeling C with a single positive that does not coincide with the positive of A and a labeling $B = A + C$ (i.e., two positives). Using $d(A,C) \leq d(A,B) + d(B,C)$:

$$c_{\text{max}} - M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} \overset{\text{Dist}}{\leq} 2c_{\text{max}} - M\begin{pmatrix} 1 & 1 \\ 0 & n-2 \end{pmatrix} - M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} \overset{\text{Sym}}{=} 2\left(c_{\text{max}} - M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix}\right),$$

This is rewritten to

$$2M\begin{pmatrix} 1 & 0 \\ 1 & n-2 \end{pmatrix} - c_{\text{max}} \leq M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix}. \tag{2}$$

Combining (1) and (2), we obtain

$$(n-1)M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} - (n-2)M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix} \leq M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix}.$$

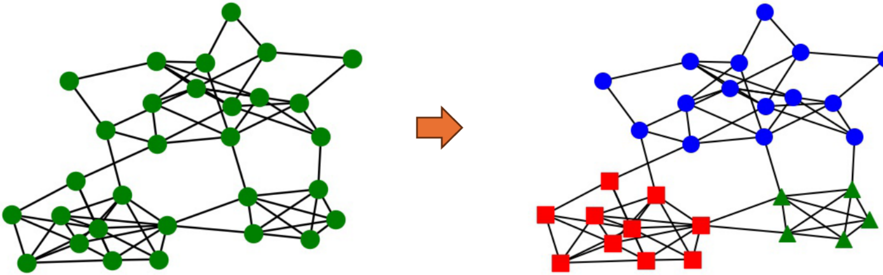
We rewrite this to $M\begin{pmatrix} 0 & 1 \\ 1 & n-2 \end{pmatrix} \leq M\begin{pmatrix} 0 & 1 \\ 2 & n-3 \end{pmatrix}$, which contradicts monotonicity.

End of Part I

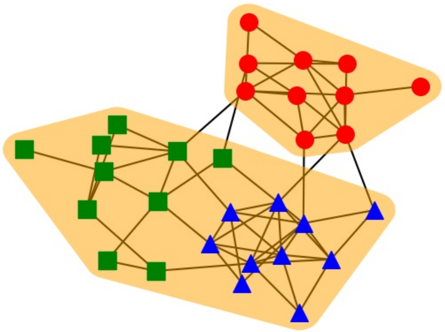
Questions?

Part II: Validating Validation Measures for Clustering

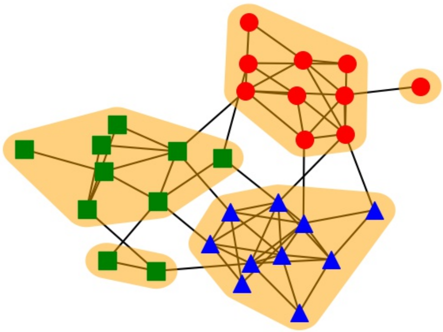
Community detection



How to measure performance?



Algorithm 1.



Algorithm 2.

Which algorithm was least wrong?

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

- A labeling can be used to represent a clustering, but we don't care about the ordering of the labels.

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

- A labeling can be used to represent a clustering, but we don't care about the ordering of the labels.
- We can define a confusion matrices as before, they don't need to be square.

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

- A labeling can be used to represent a clustering, but we don't care about the ordering of the labels.
- We can define a confusion matrices as before, they don't need to be square.
- But $M(\mathcal{C}) = M(\mathcal{C}')$ for any $\mathcal{C}'$ that is obtained by permuting the rows or columns of $\mathcal{C}$.

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

- A labeling can be used to represent a clustering, but we don't care about the ordering of the labels.
- We can define a confusion matrices as before, they don't need to be square.
- But $M(\mathcal{C}) = M(\mathcal{C}')$ for any $\mathcal{C}'$ that is obtained by permuting the rows or columns of $\mathcal{C}$.
- *Matching-based* measures use a (multiclass) classification measure M_M to define $M(\mathcal{C}) = M_M(\mathcal{C}^*)$, where $\mathcal{C}^*$ corresponds to the mapping between clusters of A and B that maximizes $M_M(\mathcal{C}^*)$.

Clustering vs classification

Clusterings (partitions) have a more complex structure than labelings, since we don't care about the identities of the clusters.

- A labeling can be used to represent a clustering, but we don't care about the ordering of the labels.
- We can define a confusion matrices as before, they don't need to be square.
- But $M(\mathcal{C}) = M(\mathcal{C}')$ for any $\mathcal{C}'$ that is obtained by permuting the rows or columns of $\mathcal{C}$.
- *Matching-based* measures use a (multiclass) classification measure M_M to define $M(\mathcal{C}) = M_M(\mathcal{C}^*)$, where $\mathcal{C}^*$ corresponds to the mapping between clusters of A and B that maximizes $M_M(\mathcal{C}^*)$.
- We won't discuss these because they are complicated and don't have good properties.

Pair-counting measures and binary classification

For two clusterings A, B, we can count the number of pairs of data points that are

- in the same cluster in both A and B (c_{11}),
- in the same cluster in A but in different clusters in B (c_{10}),
- in different clusters in A but in the same cluster in B (c_{01}),
- in different clusters in both A and B (c_{00}).

Pair-counting measures and binary classification

For two clusterings A, B, we can count the number of pairs of data points that are

- in the same cluster in both A and B (c_{11}),
- in the same cluster in A but in different clusters in B (c_{10}),
- in different clusters in A but in the same cluster in B (c_{01}),
- in different clusters in both A and B (c_{00}).

This gives a 2×2 confusion matrix that can be used with any binary classification measure.

- $c_{11} + c_{10} + c_{01} + c_{00} = \binom{n}{2} =: N$
- $c_{11} + c_{10} = \sum_i \binom{a_i}{2} =: m_A$

Pair-counting measures and binary classification

For two clusterings A, B, we can count the number of pairs of data points that are

- in the same cluster in both A and B (c_{11}),
- in the same cluster in A but in different clusters in B (c_{10}),
- in different clusters in A but in the same cluster in B (c_{01}),
- in different clusters in both A and B (c_{00}).

This gives a 2×2 confusion matrix that can be used with any binary classification measure.

- $c_{11} + c_{10} + c_{01} + c_{00} = \binom{n}{2} =: N$
- $c_{11} + c_{10} = \sum_i \binom{a_i}{2} =: m_A$

The most popular pair-counting measures are

- Rand Index (Acc)
- Adjusted Rand Index (Cohen's kappa)

Information-theoretic measures

We can interpret a clustering as a discrete distribution over the labels $0, \ldots, m-1$ with probability a_i/n. The corresponding entropy is

$$H(A) = -\sum_{i=0}^{m-1} \frac{a_i}{n} \log \frac{a_i}{n},$$

and

$$H(A, B) = -\sum_{i=0}^{m-1} \sum_{j=i}^{m-1} \frac{c_{ij}}{n} \log \frac{c_{ij}}{n}.$$

Information-theoretic measures

We can interpret a clustering as a discrete distribution over the labels $0, \ldots, m-1$ with probability a_i/n. The corresponding entropy is

$$H(A) = -\sum_{i=0}^{m-1} \frac{a_i}{n} \log \frac{a_i}{n},$$

and

$$H(A, B) = -\sum_{i=0}^{m-1} \sum_{j=i}^{m-1} \frac{c_{ij}}{n} \log \frac{c_{ij}}{n}.$$

The *mutual information* is

$$M(A, B) = H(A) + H(B) - H(A, B).$$

Information-theoretic measures

We can interpret a clustering as a discrete distribution over the labels $0, \ldots, m-1$ with probability a_i/n. The corresponding entropy is

$$H(A) = -\sum_{i=0}^{m-1} \frac{a_i}{n} \log \frac{a_i}{n},$$

and

$$H(A, B) = -\sum_{i=0}^{m-1} \sum_{j=i}^{m-1} \frac{c_{ij}}{n} \log \frac{c_{ij}}{n}.$$

The *mutual information* is

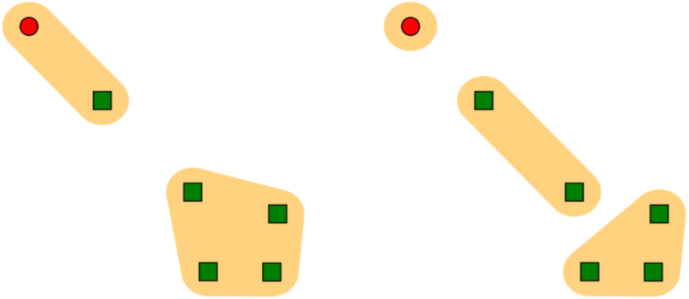
$$M(A, B) = H(A) + H(B) - H(A, B).$$

This can be normalized (in several ways) to obtain *Normalized Mutual Information* (NMI)

$$NMI(A, B) = \frac{M(A, B)}{\sqrt{H(A) \cdot H(B)}}.$$

Does it matter which measure we use?

There are many different measures. Do they differ significantly?



(a) FNMI, R, AR, J, D, W, FMeasure, BCubed

(b) NMI, $\mathrm{NMI}_{\mathrm{max}}$, VI, AMI, S&S, CC, CD

We found four such clustering triplets that fully distinguish the 16 measures we considered.

Does it *really* matter which measure we use?

We took 16 benchmark clustering datasets and applied 8 standard clustering algorithms. We count the fraction of times that two measures *disagree* on which of two algorithms is better.

Table 3. Inconsistency of indices on real-world clustering datasets, %

	NMI	NMI_{max}	VI	FNMI	AMI	R	AR	J	W	S&S	CC	FMeas	BCub
NMI	–	5.4	40.3	17.3	9.2	13.4	15.7	35.2	68.4	20.1	18.5	31.7	32.0
NMI_{max}		–	41.1	16.5	13.2	12.5	14.1	34.3	68.8	21.1	18.9	30.3	32.4
VI			–	34.7	41.8	45.2	37.6	17.1	28.8	36.0	37.2	18.1	13.6
FNMI				–	23.3	24.0	19.0	29.9	57.0	26.7	23.8	27.5	26.7
AMI					–	21.1	17.3	33.3	61.3	15.1	13.6	35.0	34.4
R						–	15.5	35.6	71.5	21.1	20.7	32.5	35.8
AR							–	23.5	59.4	11.7	8.3	25.3	28.1
J								–	35.9	23.1	23.8	10.7	9.7
W									–	53.5	54.8	40.7	37.4
S&S										–	3.6	26.2	27.8
CC											–	27.0	28.8
FMeas												–	7.7
BCub													–

Properties

	Max. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Const. baseline
NMI	✓	✓	✗	✓	✓	✗
$\mathbf{NMI}_{\max}$	✓	✓	✓	✓	✗	✗
FNMI	✓	✗	✗	✓	✗	✗
VI	✓	✓	✓	✓	✓	✗
SMI	✗	✓	✗	✗	✗	✓
FMeasure	✓	✓	✗	✓	✗	✗
BCubed	✓	✓	✗	✓	✓	✗
AMI	✓	✓	✗	✗	✓	✓

Properties for general measures.

	Max. agreement	Min. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Strong monotonicity	Const. baseline	Apx. CB
R	✓	✓	✓	✓	✓	✓	✓	✗	✗
AR	✓	✗	✓	✗	✓	✓	✗	✓	✓
J	✓	✗	✓	✓	✓	✓	✗	✗	✗
W	✗	✗	✗	✗	✓	✗	✗	✗	✗
D	✓	✗	✓	✗	✓	✓	✗	✗	✗
CC	✓	✓	✓	✗	✓	✓	✓	✓	✓
S&S	✓	✓	✓	✗	✓	✓	✓	✓	✓
CD	✓	✓	✓	✓	✓	✓	✓	✗	✓

Properties for pair-counting measures.

Properties

	Max. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Const. baseline
NMI	✓	✓	✗	✓	✓	✗
NMI$_{\max}$	✓	✓	✓	✓	✗	✗
FNMI	✓	✗	✗	✓	✗	✗
VI	✓	✓	✓	✓	✓	✗
SMI	✗	✓	✗	✗	✗	✓
FMeasure	✓	✓	✗	✓	✗	✗
BCubed	✓	✓	✗	✓	✓	✗
AMI	✓	✓	✗	✗	✓	✓

Properties for general measures.

	Max. agreement	Min. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Strong monotonicity	Const. baseline	Apx. CB
R	✓	✓	✓	✓	✓	✓	✓	✗	✗
AR	✓	✗	✓	✗	✓	✓	✗	✓	✓
J	✓	✗	✓	✓	✓	✓	✗	✗	✗
W	✗	✗	✗	✗	✓	✗	✗	✗	✗
D	✓	✗	✓	✗	✓	✓	✗	✗	✗
CC	✓	✓	✓	✗	✓	✓	✓	✓	✓
S&S	✓	✓	✓	✗	✓	✓	✓	✓	✓
CD	✓	✓	✓	✓	✓	✓	✓	✗	✓

Properties for pair-counting measures.

Some properties can only be defined for pair-counting measures (e.g., minimal agreement, approximate constant baseline).

Monotonicity

Monotonicity is more complicated for clusterings.
Let A_i denote the ith cluster of clustering A.

- *Perfect merge*: If $A_1, A_2 \subset B_i$, then merging A_1, A_2 should increase the measure.
- *Perfect split*: If $A_1 \cap B_i \neq \emptyset$ but $A_1 \not\subset B_i$, then splitting A_1 into $A_1 \cap B_i$ and $A_1 \setminus B_i$ should increase the measure.

Monotonicity

Monotonicity is more complicated for clusterings.
Let A_i denote the ith cluster of clustering A.

- *Perfect merge*: If $A_1, A_2 \subset B_i$, then merging A_1, A_2 should increase the measure.
- *Perfect split*: If $A_1 \cap B_i \neq \emptyset$ but $A_1 \not\subset B_i$, then splitting A_1 into $A_1 \cap B_i$ and $A_1 \setminus B_i$ should increase the measure.

For any A, B, we can obtain B from A by a sequence of perfect splits and merges.

Monotonicity

Monotonicity is more complicated for clusterings.
Let A_i denote the ith cluster of clustering A.

- *Perfect merge*: If $A_1, A_2 \subset B_i$, then merging A_1, A_2 should increase the measure.
- *Perfect split*: If $A_1 \cap B_i \neq \emptyset$ but $A_1 \not\subset B_i$, then splitting A_1 into $A_1 \cap B_i$ and $A_1 \setminus B_i$ should increase the measure.

For any A, B, we can obtain B from A by a sequence of perfect splits and merges.
If a binary classification measure is monotone, then the corresponding pair-counting clustering measure is also monotone w.r.t. perfect splits and merges.

Linear complexity

Clustering and community detection is often applied to large datasets.

- Only near-linear complexity algorithms are feasible.
- If a validation measure has super-linear complexity, it would form a bottleneck.

Linear complexity

Clustering and community detection is often applied to large datasets.

- Only near-linear complexity algorithms are feasible.
- If a validation measure has super-linear complexity, it would form a bottleneck.

Definition

A measure satisfies the linear complexity property if it can be computed in linear time.

Adjustment for chance

Constant baseline is just as important in clustering as in constant baseline.

Adjustment for chance

Constant baseline is just as important in clustering as in constant baseline.

- Many popular measures do not have a constant baseline.

Adjustment for chance

Constant baseline is just as important in clustering as in constant baseline.

- Many popular measures do not have a constant baseline.
- A common approach is to define an *adjusted-for-chance* version:

$$M'(A, B) = \frac{M(A, B) - \mathbb{E}_{B'}[M(A, B')]}{N(A, B) - \mathbb{E}_{B'}[M(A, B')]},$$

  where B' is a random partition with the same sizes as B and $N(A, B)$ is some normalization of $M(A, B)$.

Adjustment for chance

Constant baseline is just as important in clustering as in constant baseline.

- Many popular measures do not have a constant baseline.
- A common approach is to define an *adjusted-for-chance* version:

$$M'(A, B) = \frac{M(A, B) - \mathbb{E}_{B'}[M(A, B')]}{N(A, B) - \mathbb{E}_{B'}[M(A, B')]},$$

  where B' is a random partition with the same sizes as B and $N(A, B)$ is some normalization of $M(A, B)$.

Examples:

- *Adjusted Rand* (equivalent to Cohen's Kappa)
- *Adjusted Mutual Information*:

$$AMI(A, B) = \frac{M(A, B) - \mathbb{E}_{B'}[M(A, B')]}{\sqrt{H(A) \cdot H(B)} - \mathbb{E}_{B'}[M(A, B')]}.$$

These ‘adjustments’ often lead to new problems:

- AMI has worst-case complexity $\mathcal{O}(n^2)$.
- Adjusted Rand also loses several properties compared to Rand.
- Also, these measures are hard to interpret and analyze.

These 'adjustments' often lead to new problems:

- AMI has worst-case complexity $\mathcal{O}(n^2)$.
- Adjusted Rand also loses several properties compared to Rand.
- Also, these measures are hard to interpret and analyze.

Instead of 'patching' existing measures, we searched for existing measures that perform well in terms of our properties.

These ‘adjustments’ often lead to new problems:

- AMI has worst-case complexity $\mathcal{O}(n^2)$.
- Adjusted Rand also loses several properties compared to Rand.
- Also, these measures are hard to interpret and analyze.

Instead of ‘patching’ existing measures, we searched for existing measures that perform well in terms of our properties.

- CC (equivalent to Matthew’s Correlation Coefficient) satisfies all properties except for being a distance
- $\mathrm{CD} = \arccos \mathrm{CC}$ satisfies all properties except for constant baseline (but including approximate constant baseline).

Exercises

Consider a clustering $A_{k,s}$ consisting of k clusters of size 2 and one of size s (i.e., $n = 2k + s$).

- Pick one element from the cluster of size s and assign it to a cluster of size 1. Let $B_{k,s}$ be this clustering.
- Let $C_{k,s}$ denote a clustering that splits each of the 2-clusters of $A_{k,s}$ into two clusters of size 1.

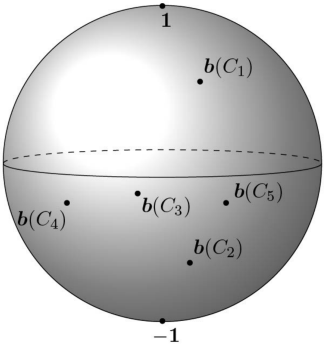
1. For given k, s, calculate the (pair-counting) confusion matrices for $M(A_{k,s}, B_{k,s})$ and $M(A_{k,s}, C_{k,s})$ in terms of n, k, s.
2. For what $k(s)$ does $M(A_{k,s}, B_{k,s}) = M(A_{k,s}, C_{k,s})$ hold? (for any pair-counting M)
3. Calculate the entropies of $A_{k,s}, B_{k,s}$ and $C_{k,s}$.
4. For what $k(s)$ does $\mathrm{NMI}(A_{k,s}, B_{k,s}) = \mathrm{NMI}(A_{k,s}, C_{k,s})$ hold?

Answers

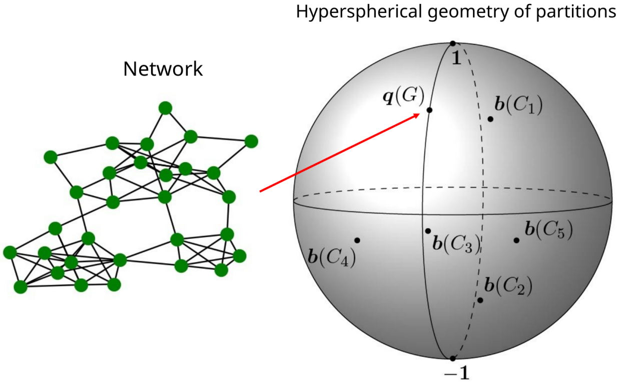
1. For $M(A,B)$, we have $m_A = c_{11} + c_{10} = k + \binom{s}{2}$, $c_{10} = s-1$, $c_{01} = 0$, $c_{11} = m_B = k + \binom{s-1}{2}$, and $c_{00} = \binom{n}{2} - m_A$. For $M(A,C)$, we have $c_{11} = m_C = \binom{s}{2}$, $c_{10} = k$, $c_{01} = 0$ and $c_{00} = n - m_A$.
2. We need to match the c_{10}'s: $k = s-1$.
3. We get
$$H(A) = -k \cdot \frac{2}{n} \log \frac{2}{n} - \frac{s}{n} \log \frac{s}{n} = \log n - \frac{2k}{n} \log 2 - \frac{s}{n} \log s.$$
For B, we only change the s-cluster:
$$H(B) = \log n - \frac{2k}{n} \log 2 - \frac{s-1}{n} \log(s-1).$$
For C, each of the $\log 2$ terms gets replaced by two two $\log 1 = 0$ terms, i.e., $H(C) = \log n - \frac{s}{n} \log s$.
4. Note that $M(A,B) = M(A,C) = H(A)$, so that $\mathrm{NMI}(A,B) = \sqrt{H(A)/H(B)}$. Solving $H(B) = H(C)$ yields
$$k = \frac{\log s + (s-1) \log \left(1 + \frac{1}{s-1}\right)}{2 \log 2} \approx \frac{1 + \log s}{2 \log 2}.$$

The Projection Method

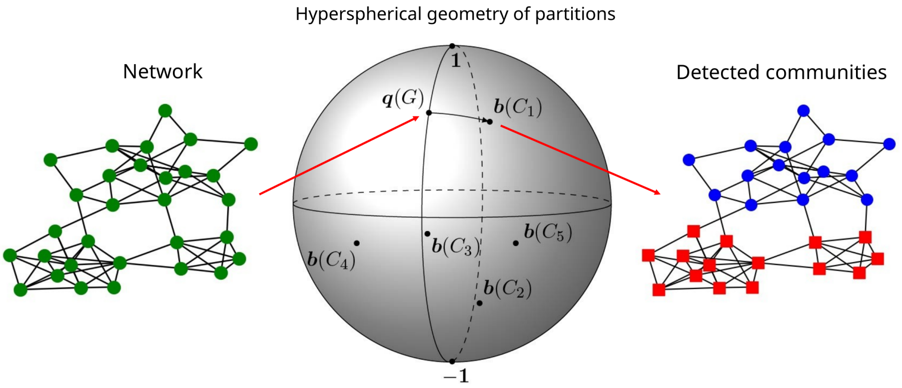
Hyperspherical geometry of partitions



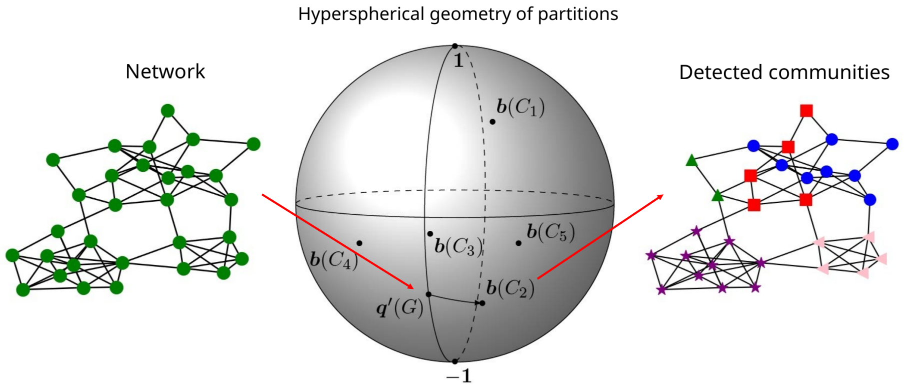
The Projection Method



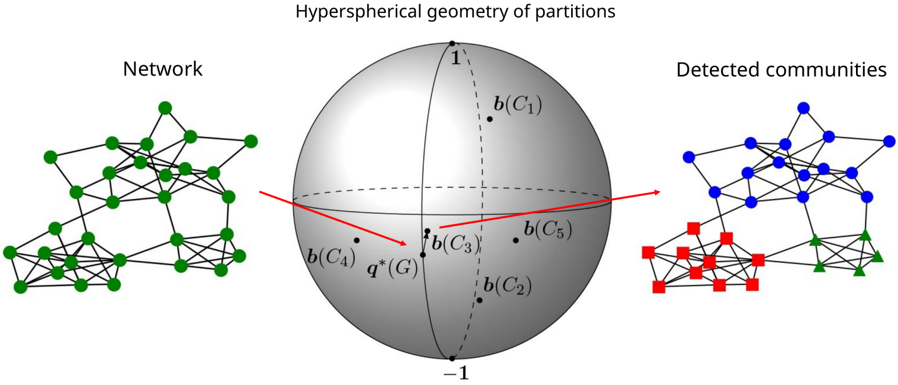
The Projection Method



The Projection Method



The Projection Method



The Diamond percolation algorithm

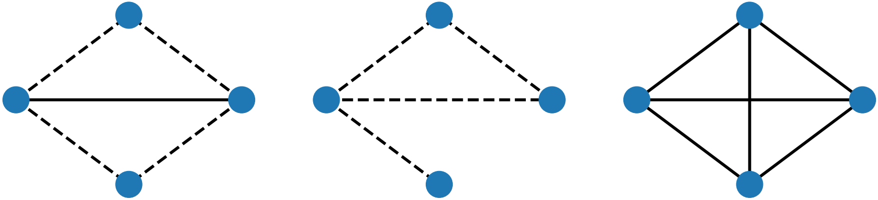
Partitions G into communities C in two steps:

1. We construct G^* by keeping all edges that are part of *at least two triangles*
2. Return C as the *connected components* of G^*

The Diamond percolation algorithm

Partitions G into communities C in two steps:

1. We construct G^* by keeping all edges that are part of *at least two triangles*
2. Return C as the *connected components* of G^*

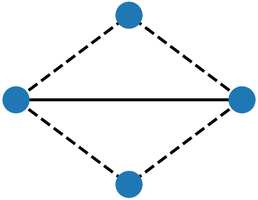


Middle edge kept in G^* | No edge kept in G^* | All edges kept in G^*

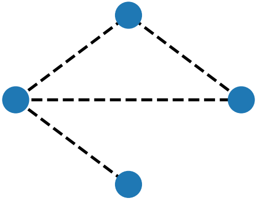
The Diamond percolation algorithm

Partitions G into communities C in two steps:

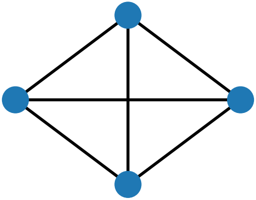
1. We construct G^* by keeping all edges that are part of *at least two triangles*
2. Return C as the *connected components* of G^*



Middle edge kept in G^*



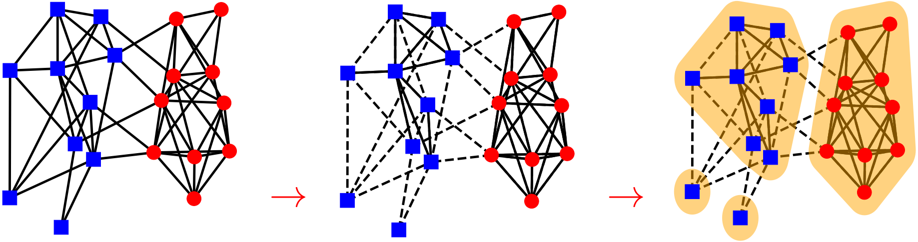
No edge kept in G^*



All edges kept in G^*

Can prove performance guarantees in terms of the correlation coefficient!

Example



Original graph G Percolated G^* Communities C

Thank you for your attention!