Clustering Metrics: Complete Guide

Introduction

Clustering is unsupervised learning—we don't have true labels. This makes evaluation different from classification. We need metrics to answer: "Are these clusters good?"

Two main scenarios:

1. **With ground truth labels** (semi-supervised evaluation)
2. **Without labels** (unsupervised evaluation)

Part 1: Metrics Using Ground Truth Labels

When you have true labels for validation, use these metrics.

1. Purity

Formula: Purity = $(1/N) \times \Sigma \max(n_{ij})$

Where n_{ij} = number of samples from class j assigned to cluster i

What it means: For each cluster, find the most common class and count those samples. Sum across all clusters and divide by total samples.

Example:

```
Cluster 1: 30 class A, 10 class B, 5 class C → max = 30
Cluster 2: 25 class B, 20 class A → max = 25
Cluster 3: 18 class C, 7 class A → max = 18
Total samples: 115

Purity = (30 + 25 + 18) / 115 = 0.73
```

Range: 0 to 1 (higher is better)

Pros: Simple to understand and compute

Cons: Biased toward many clusters (more clusters = higher purity), doesn't penalize false positives

---

2. Homogeneity

What it means: Each cluster contains only samples from a single class.

Formula: H = 1 - (entropy of classes within clusters) / (entropy of classes overall)

Intuition: If clusters perfectly separate classes, homogeneity = 1.0

Example:

> Perfect homogeneity (1.0): Each cluster has only one class
> Poor homogeneity (0.3): Clusters are mixed with multiple classes

Range: 0 to 1 (higher is better)

Pros: Penalizes mixing different classes in same cluster

Cons: Doesn't measure if all samples of a class are together

---

3. Completeness

What it means: All samples from the same class are assigned to the same cluster.

Formula: C = 1 - (entropy of clusters within classes) / (entropy of clusters overall)

Intuition: If all instances of a class are in one cluster, completeness = 1.0

Example:

> Perfect completeness (1.0): Each class is in exactly one cluster
> Poor completeness (0.4): Class A is scattered across 5 clusters

Range: 0 to 1 (higher is better)

Pros: Penalizes splitting classes across multiple clusters

Cons: Doesn't measure if clusters contain mixed classes

---

4. V-Measure

Formula: V = 2 × (Homogeneity × Completeness) / (Homogeneity + Completeness)

What it means: Harmonic mean of homogeneity and completeness. Balances both concerns.

Example:

H = 0.8, C = 0.6
V-Measure = 2 × (0.8 × 0.6) / (0.8 + 0.6) = 0.686

Range: 0 to 1 (higher is better)

Pros: Single balanced metric, not biased toward many clusters

Cons: Requires ground truth labels

---

5. Rand Index (RI)

What it means: Proportion of point pairs with consistent assignments in true labels and clustering.

Calculation:

- Count pairs of points in same cluster AND same class (agreement)
- Count pairs in different clusters AND different classes (agreement)
- Compare to total pairs

Example:

Total pairs: 10 points = 45 pairs
Pairs that agree with true labels: 38
Rand Index = 38 / 45 = 0.844

Range: 0 to 1 (higher is better)

Pros: Considers both clustered and separated pairs

Cons: Can be high even with poor clustering due to random agreement

---

6. Adjusted Rand Index (ARI)

Formula: ARI = (RI - Expected RI) / (Max RI - Expected RI)

What it means: Rand Index adjusted for chance. Removes bias from random agreement.

Example:

Random clustering: ARI ≈ 0
Perfect clustering: ARI = 1.0
Poor clustering: ARI < 0

Range: -1 to 1 (higher is better, 0 means random)

Pros: Not biased by random agreement, interprets 0 as random baseline

Cons: More complex to understand than Rand Index

---

7. Normalized Mutual Information (NMI)

What it means: Information shared between true labels and cluster assignments, normalized.

Formula: NMI = 2 × I(Y; C) / (H(Y) + H(C))

Where I = mutual information, H = entropy

Intuition: How much knowing the true label reduces uncertainty about cluster assignment.

Example:

Perfect clustering: NMI = 1.0
Random clustering: NMI ≈ 0

Range: 0 to 1 (higher is better)

Pros: Symmetric, handles multiple clusters well, theoretically grounded

Cons: Requires ground truth

---

8. F-Measure (Fowlkes-Mallows Index)

Formula: FM = √(Precision × Recall)

For clustering:

- **Precision:** Of clustered pairs, what % are truly in same class
- **Recall:** Of true pairs, what % are clustered together

Example:

Precision = 0.9 (90% of clustered pairs are same class)
Recall = 0.8 (80% of true pairs clustered together)
F-Measure = √(0.9 × 0.8) = 0.849

Range: 0 to 1 (higher is better)

Pros: Combines precision and recall meaningfully

Cons: Requires ground truth, focused on pairwise agreement

Part 2: Metrics Without Ground Truth (Unsupervised)

When you have no labels, use these to evaluate cluster quality.

1. Silhouette Coefficient

What it means: How similar a point is to its own cluster vs. other clusters.

Formula for each point:

s = (b - a) / max(a, b)

Where:
a = average distance to other points in same cluster (cohesion)
b = average distance to points in nearest other cluster (separation)

Interpretation:

- s near 1: Point well-matched to its cluster
- s near 0: Point on cluster boundary
- s near -1: Point may be in wrong cluster

Example:

Point analysis:
- Distance to own cluster points: avg = 0.5
- Distance to nearest other cluster: avg = 3.0
- Silhouette = (3.0 - 0.5) / 3.0 = 0.833 (good)

Average Silhouette Score: Mean of all point silhouettes

Range: -1 to 1 (higher is better)

Pros: Intuitive, no labels needed, gives per-point scores

Cons: Computationally expensive for large datasets, sensitive to feature scaling

---

2. Davies-Bouldin Index (DBI)

What it means: Average similarity between each cluster and its most similar cluster. Lower is better.

Formula:

$$DBI = (1/k) \times \Sigma \max(R_{i,j})$$

Where $R_{i,j} = (\sigma_i + \sigma_j) / d(c_i, c_j)$
σ = average distance within cluster
$d(c_i, c_j)$ = distance between cluster centers

Interpretation:

- DBI = 0: Perfect separation (unrealistic)
- Lower values: Better separated clusters
- Higher values: Overlapping clusters

Example:

Cluster 1 & 2: $\sigma_1 + \sigma_2 = 1.5$, center distance = 4.0, ratio = 0.375
Cluster 1 & 3: $\sigma_1 + \sigma_3 = 2.0$, center distance = 3.0, ratio = 0.667
Best ratio for cluster 1: 0.667

Average across all clusters: DBI = 0.54

Range: 0 to ∞ (lower is better)

Pros: No labels needed, fast computation, measures compactness and separation

Cons: Not normalized, interpretation depends on data

---

3. Calinski-Harabasz Index (CHI) / Variance Ratio Criterion

What it means: Ratio of between-cluster variance to within-cluster variance. Higher is better.

Formula:

```
CHI = (SSB / (k-1)) / (SSW / (n-k))

Where:
SSB = sum of squared distances between cluster centers and overall mean
SSW = sum of squared distances within each cluster
k = number of clusters
n = number of samples
```

Interpretation:

- Higher values: Better separation and compactness
- Related to F-statistic from ANOVA

Example:

```
Between-cluster variance: 50 (clusters well separated)
Within-cluster variance: 5 (points compact in clusters)
CHI = (50 / 4) / (5 / 96) = 240 (good)
```

Range: 0 to ∞ (higher is better)

Pros: Fast, considers both separation and compactness, higher is intuitively better

Cons: Not normalized, biased toward convex clusters

---

4. Gap Statistic

What it means: Compares within-cluster variance to what's expected in random data. Helps determine optimal number of clusters.

Formula:

```
Gap(k) = E[log(Wk)] - log(Wk)

Where:
Wk = within-cluster sum of squares for k clusters
E[log(Wk)] = expected value for uniform random data
```

How to use it:

1. Compute Gap statistic for k = 1, 2, 3, ..., K
2. Choose k where Gap(k) ≥ Gap(k+1) - se(k+1)
3. Typically pick smallest k satisfying this rule

Example:

```
k=1: Gap = 0.8
k=2: Gap = 1.2  ← largest gap
k=3: Gap = 1.1
k=4: Gap = 1.0

Choose k=2 (biggest improvement from 1 to 2)
```

Range: 0 to ∞ (larger gaps suggest better k)

Pros: Principled way to choose number of clusters, no labels needed

Cons: Computationally expensive (requires random sampling), complex interpretation

---

5. Dunn Index

What it means: Ratio of minimum separation to maximum compactness. Higher is better.

Formula:

```
Dunn = min(d(i,j)) / max(Δ(k))

Where:
d(i,j) = distance between cluster centers i and j
Δ(k) = diameter (max internal distance) of cluster k
```

Interpretation:

- High value: Clusters far apart and compact internally
- Low value: Clusters close or overlap

Example:

Smallest distance between cluster centers: 5.0
Largest cluster diameter: 1.5
Dunn = 5.0 / 1.5 = 3.33 (good)

Range: 0 to ∞ (higher is better)

Pros: Clear interpretation, considers cluster separation and compactness

Cons: Sensitive to outliers, computationally expensive for large datasets

---

6. Within-Cluster Sum of Squares (WCSS) / Inertia

What it means: Sum of squared distances from each point to its cluster center.

Formula:

$$WCSS = \Sigma \Sigma ||xi - cj||^2$$

Where:
xi = sample point
cj = center of cluster j

Interpretation:

- Lower WCSS: Tighter clusters
- Always decreases as k increases

Example:

k=2: WCSS = 25
k=3: WCSS = 18
k=4: WCSS = 14

Use the "Elbow Method":

- Plot WCSS vs. k
- Look for "elbow" where improvements level off
- Choose k at the elbow

Range: 0 to ∞ (lower is better)

Pros: Simple, computationally efficient, standard metric

Cons: Always decreases with more clusters (not standalone), hard to interpret absolute values

---

7. Calinski-Harabasz Score (Variance Ratio Criterion)

See section 3 above—same as CHI.

Comparison: External vs. Internal Metrics

Metric	Type	Requires Labels	What It Measures	Best Use
Purity	External	Yes	Cluster class dominance	Quick evaluation
Homogeneity	External	Yes	Class separation	Check class purity
Completeness	External	Yes	Within-class cohesion	Check class grouping
V-Measure	External	Yes	Balanced H + C	Overall external eval
Adjusted Rand Index	External	Yes	Pairwise agreement	Robust comparison
Silhouette	Internal	No	Point-cluster fit	Quality per sample
Davies-Bouldin	Internal	No	Separation/compactness	General quality
Calinski-Harabasz	Internal	No	Variance ratio	Fast evaluation
Gap Statistic	Internal	No	Optimal k selection	Find k
Dunn Index	Internal	No	Separation/compactness	Convex clusters
WCSS	Internal	No	Within-cluster distance	Elbow method

Choosing the Right Metric

If you have ground truth labels:

- **Quick check:** Use Purity
- **Balanced evaluation:** Use V-Measure or Adjusted Rand Index

- **Information-theoretic:** Use Normalized Mutual Information

If you don't have labels:

- **General quality:** Use Silhouette Coefficient or Davies-Bouldin Index
- **Fast evaluation:** Use Calinski-Harabasz Index
- **Find optimal k:** Use Gap Statistic
- **Track convergence:** Use WCSS with elbow method
- **Well-separated clusters:** Use Dunn Index

For choosing number of clusters (k):

- **Elbow method:** Plot WCSS vs. k
- **Silhouette method:** Plot average silhouette vs. k, pick highest
- **Gap statistic:** Most principled statistical approach
- **Domain knowledge:** Often the best approach

Code Examples

Silhouette Score

```python
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans

X = your_data
kmeans = KMeans(n_clusters=3).fit(X)
score = silhouette_score(X, kmeans.labels_)
print(f"Silhouette Score: {score:.3f}")  # Higher is better
```

Davies-Bouldin Index

```python
from sklearn.metrics import davies_bouldin_score

score = davies_bouldin_score(X, kmeans.labels_)
print(f"Davies-Bouldin Index: {score:.3f}")  # Lower is better
```

Calinski-Harabasz Index

```python
from sklearn.metrics import calinski_harabasz_score

score = calinski_harabasz_score(X, kmeans.labels_)
print(f"Calinski-Harabasz Index: {score:.3f}")  # Higher is better
```

V-Measure (with labels)

```python
from sklearn.metrics import v_measure_score

score = v_measure_score(true_labels, kmeans.labels_)
print(f"V-Measure: {score:.3f}")  # Higher is better
```

Elbow Method

```python
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k).fit(X)
    wcss.append(kmeans.inertia_)

import matplotlib.pyplot as plt
plt.plot(range(1, 11), wcss)
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
# Look for the "elbow" point
```

Common Pitfalls

Pitfall 1: Using only one metric

Solution: Combine multiple metrics for comprehensive evaluation

Pitfall 2: WCSS always decreases

Solution: Use elbow method or other metrics to choose k

Pitfall 3: Not scaling features

Solution: Standardize features before clustering and evaluation

Pitfall 4: Optimizing for the wrong metric

Solution: Understand what each metric measures and align with your goals

Pitfall 5: Ignoring domain knowledge

Solution: Use metrics as guidance but validate with domain expertise

---

Summary

External metrics (with labels): Purity, Homogeneity, Completeness, V-Measure, ARI, NMI

- Best for validation when ground truth is available
- Check if clustering matches known classes

Internal metrics (without labels): Silhouette, Davies-Bouldin, Calinski-Harabasz, Gap Statistic, Dunn Index, WCSS

- For real unsupervised evaluation
- No ground truth needed

Key insight: Always use multiple metrics. No single metric tells the whole story about cluster quality.