8th ICML Workshop on Automated Machine Learning (2021)

Entropy adjustment for model selection in automated clustering

I. Baimuratov BAIMURATOV.I@GMAIL.COM
ITMO University, Russia

Abstract

In this research, we consider the problem of model selection for automated clustering. We propose to avoid the theoretical results considering the impossibility of a universal clustering measure by evaluating clusterings only as a partition, not depending on data points. We consider entropy as such measure, but to prevent entropy-based model selection from trivial results we apply the method of correction for chance. We propose a new randomness model for entropy adjustment, as randomness models for adjusted mutual information are inapplicable to it. We illustrate experimentally that the optimum of adjusted entropy is not trivial and compare adjusted entropy with Silhouette and Davies-Bouldin scores. With relatively small true numbers of clusters adjusted entropy outperforms other metrics.

1. Introduction

Automated machine learning (AutoML) is intended to make machine learning technologies more available for ordinary users by automating model design. However, there is another limitation of machine learning consisting of demand in labeled data. Recently self-supervised learning approach became popular, as it weakens the demand for labeled data. However, automation of unsupervised learning that does not require labeled data at all remains understudied.

In this research, we consider the problem of model selection for automated clustering. There are theoretical results considering the impossibility of a universal clustering measure. We propose to avoid these limitations by evaluating clusterings only as a partition, not depending on data points.

We consider entropy as such a measure, but following it in selecting from all possible clusterings leads to the trivial result. To prevent entropy-based model selection from triviality we apply the method of correction for chance. We argue that randomness models for adjusted mutual information are inapplicable to entropy adjustment and propose a new randomness model for it.

We illustrate experimentally that, unlike entropy, optima of adjusted entropy varies depending on data. Also, we compare results of model selection based on adjusted entropy with ones based on Silhouette and Davies-Bouldin scores. With relatively small true numbers of clusters adjusted entropy outperforms other metrics.

2. Model selection problem for clustering

As far as we know, there is no general approach to clustering automation. We suppose it is caused by two facts.

First, unsupervised learning automation is insufficiently studied. Up to this day, the AutoML approaches consider mostly supervised learning. To illustrate this fact, we provide the formulation of AutoML as a CASH problem (Feurer et al., 2015):

$$A^*_{\lambda^*} \in \arg \min_{A^j \in A, \lambda \in \Lambda^j} \frac{1}{K} \sum_{i=1}^{K} L(A^j_\lambda, D^i_{train}, D^i_{valid}),$$

where A is an algorithm, λ is hyperparameters and D^i_{train}, D^i_{valid} are train and validation samples of i-th iteration of cross-validation. Here combined algorithm selection and hyperparameter optimization problem is approached with minimization of a loss function on a train set. This formulation is not applicable to clustering as there is no loss function nor train set.

Second, there are theoretical results considering the impossibility of a universal model or metric for clustering, which we would have been able to use as a loss function in supervised learning. This impossibility study started from Kleinberg's theorem (Kleinberg, 2002). Kleinberg considered clustering as function f that takes a distance function d on a data set S and returns a clustering partition C:

$$C = f(d(S)).$$

He proposed three requirements for ideal clustering:

- *Scale-Invariance.* For any distance function d and any $\alpha > 0$, we have $f(d) = f(\alpha \cdot d)$.
- *Richness.* $Range(f)$ is equal to the set of all partitions of S.
- *Consistency.* Let d and d' be two distance functions. If $f(d) = \Gamma$, and d' is a Γ-transformation of d, then $f(d') = \Gamma$,

and proved that they are incompatible. This result is in a sense outdated, but still, there is no general approach for model selection in clustering.

3. Clustering entropy

Our workaround for model selection in clustering consists of considering a function that evaluates clustering only as a partition and does not depend on numerical properties of data points, such as distances. By this we avoid scale-invariance and consistency requirements, therefore, it is possible to consider this function as universal. Also, as it would not depend on data points, such function would not suffer from the curse of dimensionality and would be computationally cheap.

As a starting point, we consider information entropy in this role. Entropy is the well-known information measure:

$$H(X) = -\sum_i P(x_i) log_2 P(x_i),$$

where X is a discrete random variable. Entropy has a long story of application to clustering. Clustering entropy can be defined as follows:

$$H(C) = -\sum_i \frac{|C_i|}{|C|} log_2 \frac{|C_i|}{|C|}, \tag{1}$$

where C is a clustering and C_i is a cluster.

The maximum entropy principle is applied for clustering optimization (Aldana-Bobadilla and Kuri-Morales, 2015). However, it works only for selection from clusterings with a fixed number of clusters. Selecting from clusterings with various numbers of clusters, the maximum entropy principle leads to trivial results. Given a set X, entropy $H(X)$ is maximal, when for all x_i holds $P(x_i) = \frac{1}{|X|}$, therefore, following the maximum entropy principle, in every case we should select clustering with the number of clusters $k = n$, but such clustering is useless.

4. Correction for chance

We propose to apply the correction for chance method to entropy for avoiding trivial optima. The idea of correction for chance, or adjustment, consists of subtracting from the value of a metric its expected value in order to assign lesser weights to models obtained solely due to chance. According to Hubert and Arabie (1985), given a measure s and its expected value $E_m[s]$ under some randomness model m, adjustment function is defined as follows:

$$\frac{s - E_m[s]}{s_{max} - E_m[s]}. \tag{2}$$

Vinh et al. (2010) proposed to correct mutual information for chance to evaluate clustering similarity, especially for small data sets. However, adjusted mutual information is an external measure, i.e. it evaluates clustering with respect to another label distribution. This approach is impractical as in real tasks it is not possible to evaluate clustering with respect to labeled data, or, having labeled data, it is more efficient to use supervised learning.

Moreover, it is not enough just to substitute entropy (1) as a measure s to the adjustment function (2) to get adjusted entropy $AH_m(C)$ of a clustering C under a randomness model m:

$$AH_m(C) = \frac{H(C) - E_m[H(C)]}{H(C)_{\max} - E_m[H(C)]}, \tag{3}$$

because the randomness models for mutual information are not applicable to entropy. Gates and Ahn (2017) discussed the following randomness models: permutations, fixed number of clusters and all clusterings. There are the one-sided variants considered also, but it is irrelevant in our case, as we do not have two sides for entropy.

Permutation model. We argue that the permutation model does not correct entropy for chance. The permutation model presupposes fixed cluster sizes, therefore, for every clustering C and every permutation $perm$ holds that $H(C) = H(perm(C)) = E_{perm}[H(C)]$, therefore, for every clustering C adjusted entropy $AH_{perm}(C)$ with the permutation model equals to zero:

$$AH_{perm} = \frac{H(C) - E_{perm}[H(C)]}{H(C)_{\max} - E_{perm}[H(C)]} = \frac{0}{H(C)_{\max} - E_{perm}[H(C)]} = 0.$$

Fixed number of clusters. Given a clustering C^k with a number of clusters k, applying the fixed number of clusters model num to get expected entropy $E_{num}[H(C)]$ requires calculating entropy for every clustering function that returns a partition with exactly k

subsets. The number of such partitions is described with a Stirling numbers of the second kind $S(n, k)$, which grows extremely fast. Therefore, though the fixed number of clusters model is theoretically applicable for entropy adjustment, practically it is useless.

All clusterings. The all clusterings model *all* for expected entropy $E_{all}[H(C)]$ requires iterating over all possible clusterings. The number of all clusterings is described with the Bell numbers B_n, which grows even faster than Stirling numbers of the second kind. Therefore, the all clusterings model is practically useless as well.

Thus, we need another randomness model m such that $H(C) \neq E_m[H(C)]$ for every clustering C and whose growth rate is at least polynomial.

5. Adjusted entropy

Given the adjusted entropy measure (3), the further question is what is the randomness model m for it. There are many possibilities for such a model, the $min - max$ model we propose consists of averaging minimum and maximum entropy for clusterings C^k with a fixed number of clusters k. The point is that for every C^k we can calculate minimal and maximal entropy without iterating over partitions. Therefore, the computational complexity of this model is constant.

For a given C^k, entropy is minimal, if the partition is maximally imbalanced, i.e. if there is one big part and each other part have only one element, therefore, minimal entropy is calculated as follows:

$$H_{\min}(C^k) = -(\frac{n-k+1}{n} \log_2 \frac{n-k+1}{n} + \frac{k-1}{n} \log_2 \frac{1}{n}).$$

And entropy for a given C^k is maximal if all parts have the same size $\frac{n}{k}$:

$$H_{\max}(C^k) = -\log_2 k.$$

Thus, the expected entropy with $min - max$ model is defined as follows:

$$E_{min-max}[H(C^k)] = -\frac{1}{2}(\frac{n-k+1}{n} \log_2 \frac{n-k+1}{n} + \frac{k-1}{n} \log_2 \frac{1}{n} + \log_2 k). \tag{4}$$

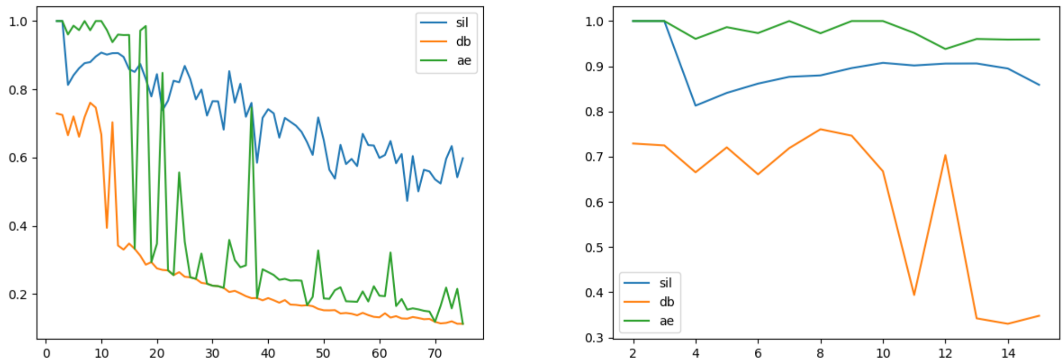
And adjusted entropy is

$$AH(C^k)_{min-max} = \frac{H(C^k) - E_{min-max}[H(C^k)]}{-\log_2 k - E_{min-max}[H(C^k)]}. \tag{5}$$

As for adjusted mutual information, for any clustering C holds $-1 \leq AH(C) \leq 1$. The best clustering has $AH(C) = 1$, while $AH(C) = -1$ is the worst option.

6. Experiments

With experiments, we are going to illustrate that the optimum of adjusted entropy, unlike unmodified entropy, varies depending on data and provides non-trivial results. To do that, we synthesized datasets with a fixed number of clusters $n = 150$ and dimensionality $d = 3$, like in the Iris dataset, but with various numbers of true clusters k_{true}.



(a) Accuracy for $k_{true} \leq 75$

(b) Accuracy for $k_{true} \leq 15$

Figure 1: Accuracy plots

For each dataset we searched for clusterings with optimal numbers of clusters k_{sil}, k_{db} and k_{ae} according to Silhouette, Davies-Bouldin and adjusted entropy scores respectively. Finally, we evaluated the accuracy of these clusterings with respect to k_{true} with Fowlkes-mallows score.

We synthesised datasets with $k_{true} \leq 75$. First of all, we point out that the optimum of adjusted entropy k_{ae} varies depending on a dataset, therefore, adjustment works as we expected. However, it appeared that adjusted entropy performs better than other metrics only for a relatively small number of true clusters, see Fig. 1a. Thus we provided the accuracy plot for $k_{true} \leq 15$ at Fig. 1b. The mean values are provided in Table 1. As we see, for $k_{true} \leq 15$ adjusted entropy has accuracy 98% and the advantage over the Silhouette score of 8%.

	k_{sil}	k_{db}	k_{ae}
$k_{true} \in 1, ..., 15$	0.9	0.61	0.98
$k_{true} \in 1, ..., 75$	0.72	0.26	0.41

Table 1: Mean accuracy

The code is presented at (Baimuratov, 2021). The experiments were performed with Intel Core i5-1035G1 CPU and 8 GB RAM.

7. Conclusion

In this research, we considered the problem of model selection for automated clustering. We argued that model selection for automated clustering is an open problem as, first, unsupervised learning is understudied in the AutoML field, and second, there are theoretical limitations considering the possibility of clustering optimization.

We proposed to avoid the theoretical limitations by considering a clustering metric that depends only on a label distribution, not on data points. First, we considered entropy in this role, but it appeared that following the maximum entropy principle in selecting from all possible clusterings leads to the trivial result.

In order to prevent entropy-based model selection from trivial results, we applied correction for chance method, like for adjusted mutual information. We analyzed the randomness models for adjusted mutual information (permutation, fixed number of clusters and all clusterings models) and they appeared to be inapplicable to entropy adjustment.

Thus we proposed a randomness model for entropy adjustment. This model consists of averaging minimum and maximum entropy for clustering with a fixed number of clusters. This model allows calculating expected entropy without iterating over partitions and has constant complexity.

We demonstrated experimentally that, unlike entropy, the optimum of adjusted entropy varies depending on data. To do it, we synthesized datasets with various true numbers of clusters and searched for the best numbers of clusters according to adjusted entropy. Also, we compared results of model selection based on adjusted entropy with ones based on Silhouette and Davies-Bouldin scores. It appeared that with relatively small true numbers of clusters adjusted entropy outperforms other metrics.

References

Edwin Aldana-Bobadilla and Angel Kuri-Morales. A clustering method based on the maximum entropy principle. *Entropy*, 17(1):151–180, 2015.

Ildar Baimuratov. Adjusted entropy. `https://github.com/ldrbmrtv/Auto2ML/tree/master/AdjustedEntropy`, 2021.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Alexander Gates and Yong-Yeol Ahn. The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, 1985.

Jon Kleinberg. An impossibility theorem for clustering. NIPS'02, page 463–470, Cambridge, MA, USA, 2002. MIT Press.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

Appendix A. Adjusted entropy code

```python
import numpy as np
from scipy.stats import entropy
from math import log

def part(y):
    value, counts = np.unique(y, return_counts = True)
    return counts

def inf(prob):
    return -log(prob, 2)

def entr(part):
    return entropy(part, base = 2)

def adj(part, mean_entr, max_entr):
    return (entr(part) - mean_entr)/(max_entr - mean_entr)

def adjusted_entropy(y):
    n = len(y)
    p = part(y)
    k = len(p)

    prob_max = (n-k+1)/n
    min_entr = prob_max*inf(prob_max) + (k-1)/n*inf(1/n)
    max_entr = inf(1/k)
    mean_entr = (max_entr + min_entr)/2
    return adj(p, mean_entr, max_entr)
```

Appendix B. Experiment code

```python
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score as sil
from sklearn.metrics import davies_bouldin_score as db
from sklearn.metrics import fowlkes_mallows_score as fm
import numpy as np
from adjusted_entropy import adjusted_entropy as ae
import matplotlib.pyplot as plt

def cluster(X, k):
    model = KMeans(n_clusters=k, random_state=0).fit(X)
    return model.labels_
```

```
def get_best_k(k_list, scores, minmax=max):
    if max:
        ind = np.argmax(scores)
    else:
        ind = np.argmin(scores)
    best_k = k_list[ind]
    return best_k

def evaluate(model, k_list, scores, score, score_name, minmax=max):
    labels = cluster(X, get_best_k(k_list, scores, minmax))
    model[score_name] = score(y, labels)

def cluster_and_evaluate(models, X, y, score):
    n = len(X)
    k_list = range(2, n)
    sil_list = []
    db_list = []
    ae_list = []
    for k in k_list:
        labels = cluster(X, k)
        sil_list.append(sil(X, labels))
        db_list.append(db(X, labels))
        ae_list.append(ae(labels))

    model = {}
    evaluate(model, k_list, sil_list, score, 'sil')
    evaluate(model, k_list, db_list, score, 'db', min)
    evaluate(model, k_list, ae_list, score, 'ae')
    models.append(model)

def present(models, k_list, score_name):
    score = [x[score_name] for x in models]
    print(np.mean(score))
    plt.plot(k_list, score, label=score_name)

n = 150
max_k = 16
models = []
k_list = range(2, max_k)
for k in k_list:
    print(k)
    X, y = make_blobs(n_samples=n, n_features=3, centers=k, random_state=0)
    cluster_and_evaluate(models, X, y, fm)

present(models, k_list, 'sil')
present(models, k_list, 'db')
present(models, k_list, 'ae')

plt.legend()
plt.show()
```