Dataset Complexity Measures and Dataset Understanding

Dieudonne Ouedraogo

8/14/2019

Abstract

Usually, a successful machine learning problem will start with the business understanding followed by the data understanding. In the previous chapters, we argued that systems thinking is critical to reducing complexity. We proposed using systems thinking rather than business understanding because business understanding is included in the systems thinking. In this chapter, we also recommend a vision of data understanding that is different than the classical view. When a dataset is perceived as a system of interacting data points, researcher and practitioners could have a better understanding of the problem at hand. This systematic approach can lead to better modeling by assigning the right algorithm to the dataset efficient performance. The data is at the core of machine learning researcher and practitioners should pay close attention to understanding the complexity around the datasets. In this paper, we show how using network science metrics that reflect the structure of a dataset can help select the best classifier to be used on the same dataset.

I- Data Collection

The real-world problems required crafting datasets that can be used to solve business problems.

To craft or build datasets that machine learning systems need, organizations need to collect raw data from the system and from external systems that have potential action on the system.

An organization should perceive data as its information ingredients that could be used to tackle business problems. A good stock of ingredients can allow the organization to build different recipes and answer various business problems.

It is very crucial to monitor the internal system and external systems by recording and collecting all possible useful ingredients.

A business should collect as much information as possible.

Domain experts, transactional data, customers surveys could be useful in guiding data collection. The data collection could be automatic or manual. However, the quality of the data plays a vital role in the insights provided by machine learning systems. Data quality and integrity should be given a priority value.

II- Constructing the ABT (Analytical base table)

This stage is the most crucial in machine learning.

a- Features

A solution to a business problem through machine learning comes from selecting the right features, characteristics, or covariates that we call here ingredients to fed machine learning algorithms.

b- Observations

After selecting the right features to build our table or dataset, we record the values associated with those features. Each instance is called observation or data point or single example in machine learning standard terms. We refer it to in the paper as a picture.

c- dataset/ ABT

It is crucial to select or build the right features for the ML algorithms, so the solutions to the business problem are consistently predictable and inferable. The data is a representation of the process; a good description leads to the right answer. The collection of pictures or data points of our systems or process should be enough and sound to get insights and to generate the correct conclusions. A collection of data points is called dataset, set, or analytical based table.

III- Structure of an ABT/ dataset

After building an ABT, we will turn our attention toward the structure of the dataset. Very often, researchers and practitioners of machine learning overlook the structure of the ABT. This area is not well documented in the literature. However, a clear vision and understanding of that structure can help reduce computation time and build accurate inferences. To highlit, the importance of the structure of the ABT we will use a task of classification, and we will see that the ABT could be perceived as a system of data points.

In machine learning, the performance of a classifier is intrinsically related to the task. The structure of the dataset plays an important role. Even though new developments and improvements are done algorithmically, very little is done to understand the dataset structure. In many situations, it can take days even weeks to train and evaluate the models. If we have to try several models, this can be resource and time-consuming, and the current trial and error process for selecting the right algorithm is not efficient. We consider the case of a binary classification problem, but it is possible to generalize the findings into more than two classes problems. In this work we use network metrics to describe the complexity of a data set relative to a classification task. A dataset is transformed into a graph representation based on the ϵNN algorithm. A data point is a node and an edge exist between two points i,j if $d(i, j) < \epsilon$. A post-processing step is applied to the graph, pruning edges between examples of different classes. The structural information such as density, clustering coefficient, and hubs are extracted. Various data sets are collected, and their metrics are computed and used as a training dataset to build a predictive model where the outputs are the algorithms competing to be used as the best classifier on a dataset. The algorithms used in this study are Decision Trees, Naive Bayes, SVM, Logistic regression, Artificial Neural Networks,K-nearest neighbors

a- Prior work

A classification difficulty dramatically depends on the dataset. Understanding the characteristics associated with the dataset can help define and recommend the right algorithm to improve the performance of the task and to save time.

(Domingos, 2000) explained that the error of a predictor arises from three sources: a bias, from the difficulty of an algorithm to accurately model the relationship present in data; a variance, from the estimation of the correct parameters from the model due to imperfections from the sample used; a fundamental error referred to as noise.

For Ho & Basu (2002), classification difficulty comes from three components: the complexity of the decision boundary, the sample size and dimensionality induced sparsity and the ambiguity of the classes.

Ho (2008), believed that data complexity analysis is essential when comparing algorithms performance in machine learning. Usage of dataset complexity can also be found in combinatorial optimization, Smith-Miles & Lopes, (2012).

Choosing a sufficiently diverse set of problems to explain both strengths and weaknesses of the analyzed algorithms is essential in determining the domain of competence of the algorithm.

Macià et al. (2013), who described how algorithm comparison might be biased by benchmark dataset selection, and showed how complexity measures might guide the choice. Characterizing problem space with some metrics makes it possible to estimate regions in which specific algorithms perform well as detailed by Luengo & Herrera, (2013). This leads to possibilities of meta-learning as described by Smith-Miles et al.,(2014).

Complexity measures could then be used not only as predictors of classifier performance but also as diversity measures capturing various properties of the datasets.

Ho & Basu (2002) introduced complexity measures which were also extended by Ho, Basu & Law (2006) and Orriols-Puig, Macià & Ho (2010). Those measures are often used for algorithm's evaluation as described by Macià et al., (2013) and Luengo & Herrera, (2013), they are also used in meta-learning (Diez-Pastor et al., 2015; Mantovani et al., 2015). Part of these measures focuses on the overlap of values of specific attributes: Fisher's discriminant ratio, the volume of the overlap region, the attribute efficiency, etc.

Another part is toward the class separability; in this section, we have measures such as the fraction of points on the decision boundary, the linear separability, the ratio of intra/interclass distance. Those measures focus on specific properties of the classification problem, measuring the shape of the decision boundary and the amount class overlap. We also have topological measures concerned with data sparsity, such as the ratio of attributes to observations.

Li & Abu-Mostafa (2006) defined dataset complexity using the general concept of Kolmogorov complexity. The measures proposed to use the number of support vectors in the support vector machine (SVM) classifier. They analyzed the problems of data decomposition and data pruning using the above methodology. They defined the representation of the dataset complexity called the complexity-error plot.

Smith, Martinez & Giraud-CARRIER(2013) tackled the problem of complexity in the dataset from a single instance point of view where they analyzed misclassified instances by various algorithms approach to data complexity is to explain. They devised local complexity measures calculated concerning the individual case and explored the correlations of those measures with the global data complexity measures of Ho & Basu (2002). They concluded that they are mainly related to class overlap.

Yin et al. (2013) used a feature selection based on Hellinger distance to described complexity by measuring the similarity between probability distributions. They chose features, which conditional distributions (depending on the class) have a minimal affinity. The authors demonstrated experimentally that, for the high-dimensional imbalanced data sets, their method is superior to popular feature selection methods using the Fisher criterion, or mutual information.

Complexity measures of a dataset

A-Measure of overlapping

The feature overlapping measures characterize how informative the available fea- tures are to separate the classes

F1 - Maximum Fisher discriminant ratio.

The measure gives the effectiveness of a single feature in separating the classes This measure computes the maximum discriminative power (Fisher ratio) of the attributes. The ratio is defined as

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

where μ_1,μ_2,σ_1^2, σ_2^2 are the means and variances of each class, respectively, in that feature. f is computed for each feature and maximum is taken as F1. A high value of F1 indicates that at least one of the attributes enables the learner to separate the examples of different classes with partitions that are parallel to an axis of the feature space. A low value of this measure does not imply that the classes are not linearly separable, but that they cannot be discriminated by hyperplanes parallel to one of the axis of the feature space.

F1v

Directional-vector maximum Fisher's discriminant ratio (F1v) complements F1 by searching for a vector able to separate two classes after the training examples have been projected into it.

F2

Volume of overlap region,this measure computes the overlap of the tails of distributions defined by the instances of each class. Let $min(f_i, c_j)$ and $max(f_i, c_j)$ be, respectively, the minimum and maximum values of the feature f_i for class c_j. Then, the overlap measure is defined as

$$F2 = \prod \frac{MINMAX_i - MAXMIN_i}{MAXMAX_i - MINMIN_i}$$

where $MINMAX_i = min(max(f_i, c_1), max(f_i, c_2))$ $MAXMIN_i = max(min(f_i, c_1), min(f_i, c_2))$ $MAXMAX_i = max(max(f_i, c_1), max(f_i, c_2))$ $MINMin_i = min(min(f_i, c_1), min(f_i, c_2))$

A low value states that the attributes can discriminate the examples of different classes.

F4

Collective feature efficiency(F4) get an overview on how various features may work together in data separation. First the most discriminative feature according to F3 is selected and all examples that can be separated by this feature are removed from the dataset. The previous step is repeated on the remaining dataset until all the features have been considered or no example remains. F4 returns the ratio of examples that have been discriminated.

B- measures of neighborhood

Neighborhood measures characterize the presence and density of same or different classes in local neighborhoods. The Neighborhood measures analyze the neighborhoods of the data items and try to capture class overlapping and the shape of the decision boundary. They work over a distance matrix storing the distances between all pairs of data points in the dataset. To deal with both symbolic and numerical features, we adopt a heterogeneous distance measure named Gower distance.

N1

Fraction of borderline points (N1) computes the percentage of vertexes incident to edges connecting examples of opposite classes in a Minimum Spanning Tree (MST).

N2

Ratio of intra/extra class nearest neighbor distance (N2) computes the ratio of two sums: intra-class and inter-class. The former corresponds to the sum of the distances between each example and its closest neighbor from the same class. The later is the sum of the distances between each example and its closest neighbor from another class (nearest enemy).

N3

Error rate of the nearest neighbor(N3)classifier corresponds to the error rate of a one Nearest Neighbor (1NN) classifier, estimated using a leave-one-out procedure in dataset.

N4

Non-linearity of the nearest neighbor classifier (N4) creates a new dataset randomly interpolating pairs of training examples of the same class and then induce a the 1NN classifier on the original data and measure the error rate in the new data points.

T1

Fraction of hyper-spheres covering data (T1) builds hyper-spheres centered at each one of the training examples, which have their radios growth until the hyper-sphere reaches an example of another class. Afterwards, smaller hyper-spheres contained in larger hyper-spheres are eliminated. T1 is finally defined as the ratio between the number of the remaining hyper-spheres and the total number of examples in the dataset.

LSCAvg

Local Set Average Cardinality (LSCAvg) is based on Local Set (LS) and defined as the set of points from the dataset whose distance of each example is smaller than the distance from the examples of the different class. LSCAvg is the average of the LS.

C-Measures of linearity

The linearity measures try to quantify if it is possible to separate the classes by a hyper-plane. The underlying assumption is that a linearly separable problem can be considered simpler than a problem requiring a non-linear decision boundary.

L1

Sum of the error distance by linear programming (L1) computes the sum of the distances of incorrectly classified examples to a linear boundary used in their classification.

L2

Error rate of linear classifier(L2)computes the error rate of the linear SVM classifier induced from dataset.

L3

Non-linearity of a linear classifier (L3) creates a new dataset randomly interpolating pairs of training examples of the same class and then induce a linear SVM on the original data and measure the error rate in the new data points.

D- Measures of dimensionality

These measures give an indicative of data sparsity. They capture how sparse a datasets tend to have regions of low density. These regions are know to be more difficult to extract good classification models.

T2

Average number of points per dimension (T2) is given by the ratio between the number of examples and dimensionality of the dataset.

T3

Average number of points per PCA (T3) is similar to T2, but uses the number of PCA components needed to represent 95 variability as the base of data sparsity assessment.

T4

Ratio of the PCA Dimension to the Original (T4) it estimates the proportion of relevant and the original dimensions for a dataset.

E-Measures of class balance

These measures capture the differences in the number of examples per class in the dataset. When these differences are severe, problems related to generalization of the ML classification techniques could happen because of the imbalance ratio.

C1

The entropy of class proportions (C1) measure the imbalance in a dataset based on the pro- portions of examples per class.

C2

The imbalance ratio (C2) is an index computed for measuring class balance. This is a version of the measure that is also suited for multiclass classification problems.

3-Measures of Network and Methodology

The network measures represent the dataset as a graph and extract structural information from it. The transformation between raw data and the graph representation is based on the epsilon-NN algorithm. Next, a post-processing step is applied to the graph, pruning edges between examples of opposite classes.

Density

Average Density of the network (Density) represents the number of edges in the graph, divided by the maximum number of edges between pairs of data points.

ClsCoef

Clustering coefficient (ClsCoef) averages the clustering tendency of the vertexes by the ratio of existent edges between its neighbors and the total number of edges that could possibly exist between them.

Hubs

Hubs score (Hubs) is given by the number of connections it has to other nodes, weighted by the number of connections these neighbors have.

The methodology proposed here could broken into three parts as below

Part1

Creation of the graph from a dataset and extraction of metrics. each data point from the dataset is a node. We transform the dataset into a graph and we extract structural information from it. The transformation between raw data and the graph representation is based on the ϵNN algorithm.nodes i and j are connected by an edge, if the distance $d(i, j) < \epsilon$. The hyper-parameter ϵ controls the neighborhood radius. Next, a post-processing step is applied to the graph, pruning edges between examples of opposite classes.

Part2

Creation of the meta Dataset We pick datasets from UCI and Weka and Kaggle which cover a broad spectrum of characteristics for datasets. We create a meta feature target which value represents the best classifier reported so far by previous studies on that particular dataset. We use the characteristics of our networks as features for the meta-dataset (the dataset formed by the collection of datasets). The set of values of the characteristics of each dataset is an instance(observation) in our meta-dataset. A sample meta-dataset is below!

Part3

We run multiclass classification algorithms of our meta-dataset. From this classification, we extract the most predictive features from the network characteristics. We then build a predictive model based on the network characteristics related to datasets

Data Set	Density	ClsCoef	Hubs	Classifier
Data Set 1	0.12267744	0.66259662	0.19018967	Knn
Data Set 2	0.16689038	0.73260435	0.21731327	Knn
Data Set 3	0.18770169	0.61501807	0.31778641	SVM
Data Set 4	0.08261087	0.75370635	0.1407504	SVM
Data Set 5	0.12462275	0.46805337	0.42546745	DT
Data Set 6	0.15489668	0.55026316	0.37760867	SVM
Data Set 7	0.16296607	0.46244857	0.32698233	Knn
Data Set 8	0.10032129	0.54130901	0.27980588	SVM
Data Set 9	0.13096887	0.52377361	0.25825507	LDA
Data Set 10	0.18939032	0.6414742	0.3495547	DT
Data Set 11	0.11474631	0.8304949	0.16130215	NN
Data Set 12	0.19112739	0.60846942	0.37585083	Knn
Data Set 13	0.13113155	0.55996607	0.26080425	Knn
Data Set 14	0.02252658	0.84263901	0.0376098	DT
Data Set 15	0.16892021	0.64951819	0.21242524	Knn
Data Set 16	0.12711158	0.70428715	0.21882891	Knn
Data Set 17	0.04958247	0.88500332	0.21287715	NN

Figure 1: Sample meta-dataset

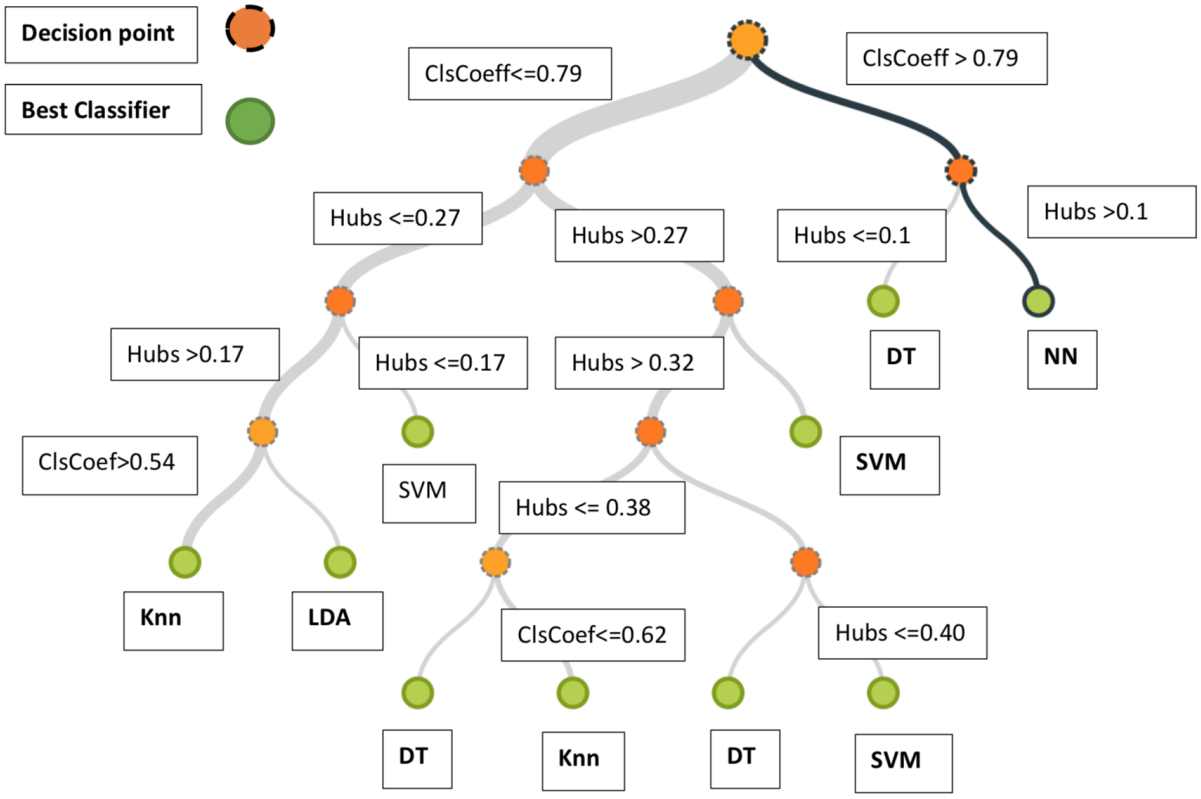

Figure 2: Decision Tree with network metrics as variables and algorithms as outputs

Results, future work

The results show that network metrics such as clustering coefficient, hubs, density are very informative in predicting the classifier to be used on the task. For clustering coefficient greater than 0.79 or hubs greater 0.1, neural network are the best performers; for hubs less than 0.1, using decision trees gives the best performance.

A decision tree algorithm is run on our meta-dataset formed by the set of network metrics, based on previous work the best classifier is known. The results show how the network metrics in the structure of the dataset could infer the best classifier. Machine learning pipelines involved long testing and comparison of performances of several algorithms with different parameters and steps. Our method is a meta-learning tool, to learn which machine learning algorithm to use without spending much time in testing. In this study we tackle classification, future research will address regression problems where the output is continuous; we will also explore unsupervised algorithms such as clustering. Issues with data collection are not addressed in this paper, we will be tackling the complexity surrounding data collection in future work.

Reference

1] E. Mansilla and T. K. Ho, "On classifier domains of competence," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, 2004, pp. 136–139 Vol.1.

[2] Tin K Ho and Mitra Basu. (2002). Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 3, 289–300.

[3] Enrique Leyva, Antonio Gonzalez, and Raul Perez. (2014). A Set of Complexity Measures Designed for Applying Meta-Learning to Instance Selection. IEEE Transactions on Knowledge and Data Engineering 27, 2, 354–367.

Lorena, A. C., Garcia, L. P. F., Lehmann, J., de Souto, M. C. P., and Ho, T. K. (2018). How Complex is your classification problem? A survey on measuring classification complexity. arXiv:1808.03591

Lorena, A. C., Maciel, A. I., de Miranda, P. B. C., Costa, I. G., and Prudêncio, R. B. C. (2018). Data complexity meta-features for regression problems. Machine Learning, 107(1):209-246.

Ana C Lorena, Ivan G Costa, Newton Spolaor and Marcilio C P Souto. (2012). Analysis of complexity indices for classification problems: Cancer gene expression data. Neurocomputing 75, 1, 33–42.

Gleison Morais and Ronaldo C Prati. (2013). Complex Network Measures for Data Set Characterization. In 2nd Brazilian Conference on Intelligent Systems (BRACIS). 12–18.

Luis P F Garcia, Andre C P L F de Carvalho and Ana C Lorena. (2015). Effect of label noise in the complexity of classification problems. Neurocomputing 160, 108–119.

Ajay K Tanwani and Muddassar Farooq. (2010). Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets. Learning Classifier Systems 6471, 127–144.

Tin K Ho and Mitra Basu. (2002). Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 3, 289–300.

Albert Orriols-Puig, Nuria Macia and Tin K Ho. (2010). Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull.

Enrique Leyva, Antonio Gonzalez and Raul Perez. (2014). A Set of Complexity Measures Designed for Applying Meta-Learning to Instance Selection. IEEE Transactions on Knowledge and Data Engineering 27, 2, 354–367.

R. Leite, P. Brazdil, and J. Vanschoren, "Selecting classification algorithms with active testing," in MLDM, ser. Lecture Notes in Computer Science, P. Perner, Ed., vol. 7376. Springer, 2012, pp. 117–131.

P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, Metalearning: applications to data mining. Springer, 2009.

T. K. Ho, M. Basu, and M. H. C. Law, Data Complexity in Pattern Recognition. Springer, 2005, ch. Measures of Geometrical Complexity in Classification Problems.

J. M. Sotoca, R. A. Mollineda, and J. S. Sanchez "A meta-learning framework for pattern classification by means of data complexity measures," Inteligencia Artificial, vol. 10, no. 29, pp. 31–38, 2006.

T. K. Ho and H. S. Baird, "Pattern classification with compact distribution maps," Computer Vision and Image Understanding, vol. 70, no. 1, pp. 101 – 110, 1998.

F. Smith, "Pattern classifier design by linear programming," Computers, IEEE Transactions on, vol. C-17, no. 4, pp. 367–372, 1968.

A. Hoekstra and R. Duin, "On the nonlinearity of pattern classifiers," in Proceedings of the 13th International Conference on Pattern Recognition, vol. 4, 1996, pp. 271–275.

L. Frank and E. Hubert, "Pretopological approach for supervised learn- ing," in Proceedings of the 13th International Conference on Pattern Recognition, vol. 4, 1996, pp. 256–260.

E. Mansilla and T. K. Ho, "On classifier domains of competence," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, 2004, pp. 136–139 Vol.1.

S.-W. Kim and B. J. Oommen, "On using prototype reduction schemes to enhance the computation of volume-based inter-class overlap measures," Pattern Recognition, vol. 42, no. 11, pp. 2695 – 2704, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320309001642

X. Zhu, "Semi-supervised learning with graphs,"Ph.D. thesis, Carnegie Mellon University, 2005, http: //pages.cs.wisc.edu/ jerryzhu/pub/thesis.pdf.

N. Ganguly, A. Deutsch, and A. Mukherjee, Eds., Dynamics On and Of Complex Networks Applications to Biology, Computer Science, and the Social Sciences. Springer, 2009.

E. D. Kolaczyk, Statistical Analysis of Network Data: Methods and Model. Springer, 2009.

A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," SIGKDD Explorations, vol. 11, no. 1, pp. 10–18, 2009.

R. C. Prati, "Combining feature ranking algorithms through rank aggre- gation," in IJCNN. IEEE, 2012, pp. 1–8.

D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," J. Artif. Intell. Res. (JAIR), vol. 6, pp. 1–34, 1997.

G. E. A. P. A. Batista and D. F. Silva, "How k-nearest neighbor parameters affect its performance," in Argentine Symposium on Artificial Intelligence, 2009, pp. 1–12.

C. Soares, "Uci++: Improved support for algorithm selection using datasetoids," in PAKDD, ser. Lecture Notes in Computer Science, T. Theeramunkong, B. Kijsirikul, N. Cercone, and T. B. Ho, Eds., vol. 5476. Springer, 2009, pp. 499–506.

A. Ben-David, "Comparison of classification accuracy using Cohen's Weighted Kappa," Expert Syst. Appl., vol. 34, no. 2, pp. 825–832, 2008.