Int. J. Adv. Res. Sci. Technol. Volume 2, Issue 2, 2013, pp. 110-114.

IJARST

International Journal of Advanced Research in Science and Technology

journal homepage: www.ijarst.com

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

Study of Various Classification Algorithms using Data Mining

Smruti Ranjan Swain and Smruti Smaraki Sarangi*

Gandhi institute for Technology, Bhubaneswar, Odisha, India

****Corresponding Author's Email: smrutiranjan@gift.edu.in***

ARTICLEINFO

Article history:

Received	10 Sep. 2013
Accepted	26 Sep. 2013
Available online	02 Oct. 2013

Keywords:

Classification,
Naïve Bayes Classifier
Nearest Centroid Classifier.

ABSTRACT

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.).In the terminology of machine learning,[1] classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Introduction:

Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, regressors, etc.), and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the observations are often known as *instances*, the explanatory variables are termed *features* (grouped into a feature vector), and the possible categories to be predicted are *classes*. There is also some argument[*citation needed*] over whether classification methods that do not involve a statistical model can be considered "statistical". Other fields may use different terminology: e.g. in community ecology, the term "classification" normally refers to cluster analysis, i.e. a type of unsupervised learning, rather than the supervised learning

Challenges:

Classification and clustering are examples of the more general problem of pattern recognition, which is the assignment of some sort of output value to a given input value. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence; etc.

A common subclass of classification is probabilistic classification. Algorithms of this nature use statistical inference to find the best class for a given instance. Unlike other algorithms, which simply output a "best" class, probabilistic algorithms output a probability of the instance being a member of each of the possible classes. The best class is normally then selected as the one with the highest probability. However, such an algorithm has numerous advantages over non-probabilistic classifiers:

- It can output a confidence value associated with its choice (in general, a classifier that can do this is known as a *confidence-weighted classifier*).
- Correspondingly, it can *abstain* when its confidence of choosing any particular output is too low.

- Because of the probabilities which are generated, probabilistic classifiers can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of *error propagation*.

Frequentist procedures:

Early work on statistical classification was undertaken by Fisher,[2][3] in the context of two-group problems, leading to Fisher's linear discriminant function as the rule for assigning a group to a new observation.[4] This early work assumed that data-values within each of the two groups had a multivariate normal distribution. The extension of this same context to more than two-groups has also been considered with a restriction imposed that the classification rule should be linear.[4][5] Later work for the multivariate normal distribution allowed the classifier to be nonlinear:[6] several classification rules can be derived based on slight different adjustments of the Mahalanobis distance, with a new observation being assigned to the group whose centre has the lowest adjusted distance from the observation.

Bayesian procedures:

Unlike frequentist procedures, Bayesian classification procedures provide a natural way of taking into account any available information about the relative sizes of the sub-populations associated with the different groups within the overall population.[7] Bayesian procedures tend to be computationally expensive and, in the days before Markov chain Monte Carlo computations were developed, approximations for Bayesian clustering rules were devised.[8]

Some Bayesian procedures involve the calculation of group membership probabilities: these can be viewed as providing a more informative outcome of a data analysis than a simple attribution of a single group-label to each new observation.

Binary and multiclass classification:

Classification can be thought of as two separate problems – binary classification and multiclass classification. In binary classification, a better understood task, only two classes are involved, whereas multiclass classification involves assigning an object to one of several classes.[9] Since many classification methods have been developed specifically for binary classification, multiclass classification often requires the combined use of multiple binary classifiers

Feature vectors:

Most algorithms describe an individual instance whose category is to be predicted using a feature vector of individual, measurable properties of the instance. Each property is termed a feature, also known in statistics as an explanatory variable (or independent variable, although in general different features may or may not be statistically independent). Features may variously be binary ("male" or "female"); categorical (e.g. "A", "B", "AB" or "O", for blood type); ordinal (e.g. "large", "medium" or "small"); integer-valued (e.g. the number of occurrences of a particular word in an email); or real-valued (e.g. a measurement of blood pressure). If the instance is an image, the feature values might correspond to the pixels of an image; if the instance is a piece of text, the feature values might be occurrence frequencies of different words. Some algorithms work only in terms of discrete data and require that real-valued or integer-valued data be *discretized* into groups (e.g. less than 5, between 5 and 10, or greater than 10).

The vector space associated with these vectors is often called the *feature space*. In order to reduce the dimensionality of the feature space, a number of dimensionality reduction techniques can be employed.

Linear classifiers:

A large number of algorithms for classification can be phrased in terms of a linear function that assigns a score to each possible category *k* by combining the feature vector of an instance with a vector of weights, using a dot product. The predicted category is the one with the highest score. This type of score function is known as a linear predictor function and has the following general form:

where X_i is the feature vector for instance *i*, β_k is the vector of weights corresponding to category *k*, and score(X_i, *k*) is the score associated with assigning instance *i* to category *k*. In discrete choice theory, where instances represent people and categories represent choices, the score is considered the utility associated with person *i* choosing category *k*. Algorithms with this basic setup are known as linear classifiers. What distinguishes them is the procedure for determining (training) the optimal weights/coefficients and the way that the score is interpreted.

Classifier Chains:

Classifier chains is a machine learning method for problem transformation in multi-label classification. It combines computational efficiency of Binary Relevance method and possibility to use dependencies between labels for classification.[1]

Problem transformation:

Problem transformation methods transform a multi-label classification problem in one or more single-label classification problems.[2] In such a way existing single-label classification algorithms such as SVM and Naive Bayes can be used without modification. Several problem transformation methods exist. One of them is Binary Relevance method (BR). During this process the information about dependencies between labels is not

preserved. This can lead to a situation where a set of labels is assigned to an instance although these labels never co-occur together in the data set. Thus, information about label co-occurrence can help to assign correct label combinations. Loss of this information can in some cases lead to decrease of the classification performance.[3]

Other approach, which takes into account label correlations is Label Powerset method (LP). Each different combination of labels in a data set is considered to be a single label. After transformation a single-label classifier is trained where is thepower set of all labels in. The main drawback of this approach is that the number of label combinations grows exponentionally with the number of labels. For example, multi-label data set with 10 labels can have up to label combinations. This increases the run-time of classification. Classifier Chains method is based on the BR method and it is efficient even on a big number of labels. Furthermore, it considers dependencies between labels.

Method description:

Thus, classifiers build a chain where each of them learns binary classification of a single label. [11]The features given to each classifier are extended with binary values that indicate which of previous labels were assigned to the instance.

By classifying new instances the labels are again predicted by building a chain of classifiers. The classification begins by passing label information between classifiers through the feature space. Hence, the inter-label dependency is preserved. However, the result can vary for different order of chains. For example, if a label often co-occur with some other label only instances of one of the labels, which comes later in the label order, will have information about other one in its feature vector. In order to solve this problem and increase accuracy it is possible to use ensemble of classifiers.[4]

In Ensemble of Classifier Chains (ECC) several CC classifiers can be trained with random order of chains (i.e. random order of labels) on a random subset of data set. Labels of a new instance are predicted by each classifier separately. After that, the total number of predictions or "votes" is counted for each label. The label is accepted if it was predicted by a percentage of classifiers that is bigger than some threshold value.

Margin classifier:

Margin classifer is a classifier which is able to give an associated distance from the decision boundary for each example. For instance, if a linear classifier (e.g. perceptron or linear discriminant analysis) is used, the distance (typically euclidean distance, though others may be used) of an example from the separating hyperplane is the margin of that example.

The notion of margin is important in several machine learning classification algorithms, as it can be used to bound the generalization error of the classifier. These bounds are frequently shown using the VC dimension. Of particular prominence is the generalization error bound on boosting algorithms and support vector machines.[18]

Margin for boosting algorithms:

The margin for an iterative boosting algorithm given a set of examples with two classes can be defined as follows.

By this definition, the margin is positive if the example is labeled correctly and negative is the example is labeled incorrectly.

This definition may be modified and is not the only way to define margin for boosting algorithms. However, there are reasons why this definition may be appealing.[1]

Examples of margin-based algorithms:

Many classifiers can give an associated margin for each example. However, only some classifiers utilize information of the margin while learning from a data set.[16]

Many boosting algorithms rely on the notion of a margin to give weights to examples. If a convex loss is utilized (as in Ada Boost, Logit Boost, and all members of the Any Boost family of algorithms) then an example with higher margin will receive less (or equal) weight than an example with lower margin.[20] This leads the boosting algorithm to focus weight on low margin examples. In non-convex algorithms (e.g. Brown Boost), the margin still dictates the weighting of an example, though the weighting is non-monotone with respect to margin. There exists boosting algorithms that provably maximize the minimum margin (e.g. see [2]).

Support vector machines provably maximize the margin of the separating hyperplane. Support vector machines that are trained using noisy data (there exists no perfect separation of the data in the given space) maximize the soft margin. More discussion of this can be found in the support vector machine article. The voted-perceptron algorithm is a margin maximizing algorithm based on an iterative application of the classic perceptron algorithm.

Multi-label classification:

multi-label classification and the strongly related problem of multi-output classification are variants of the classification problem where multiple target labels must be assigned to each instance. Multi-label classification should not be confused with multiclass classification, which is the problem of categorizing instances into more than two classes. Formally, multi-label learning can be phrased as the problem of finding a model that

maps inputs x to binary vectors y, rather than scalar outputs as in the ordinary classification problem.[15]

There are two main methods for tackling the multi-label classification problem:[1] problem transformation methods and algorithm adaptation methods. Problem transformation methods transform the multi-label problem into a set of binary classification problems,[14] which can then be handled using single-class classifiers. Algorithm adaptation methods adapt the algorithms to directly perform multi-label classification. In other words, rather than trying to convert the problem to a simpler problem, they try to address the problem in its full form.

Multiclass classification:

Multiclass or multinomial classification is the problem of classifying instances into more than two classes. While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies.[17] Multiclass classification should not be confused with multi-label classification, where multiple labels are to be predicted for each instance.

Naive Bayes classifier:

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes models are also known under a variety of names in the literature, including simple Bayes and independence Bayes.[1] All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method;[1] Russell and Norvig note that "[naive Bayes] is sometimes called a Bayesian classifier, a somewhat careless usage that has prompted true Bayesians to call it the idiot Bayes model."[2]:482

Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s,[2]:488 and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate preprocessing, it is competitive in this domain with more advanced methods including support vector machines.[3] It also finds application in automatic medical diagnosis.[4]

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[2]:718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Nearest centroid classifier:

Nearest centroid or nearest prototype classifier is a classification model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation.

When applied to text classification using tf*idf vectors to represent documents, the nearest centroid classifier is known as the Rocchio classifier because of its similarity to the Rocchio algorithm for relevance feedback.[1]

An extended version of the nearest centroid classifier has found applications in the medical domain, specifically classification of tumors.[2]

Conclusion:

This paper discuss about various classification technique considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

References:

1. Alpaydin, Ethem (2010). *Introduction to Machine Learning*. MI Press. p. 9. ISBN 978-0-262-01243-0.
2. Fisher R.A. (1936) " The use of multiple measurements in taxonomic problems", *Annals of Eugenics*, 7, 179–188
3. Fisher R.A. (1938) " The statistical utilization of multiple measurements", *Annals of Eugenics*, 8, 376–386
4. Gnanadesikan, R. (1977) *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley. ISBN 0-471-30845-5 (p. 83–86)
5. Rao, C.R. (1952) *Advanced Statistical Methods in Multivariate Analysis*, Wiley. (Section 9c)
6. Anderson,T.W. (1958) *An Introduction to Multivariate Statistical Analysis*, Wiley.
7. Binder, D.A. (1978) "Bayesian cluster analysis", *Biometrika*, 65, 31–38.
8. Binder, D.A. (1981) "Approximations to Bayesian clustering rules", *Biometrika*, 68, 275–285.
9. Har-Peled, S., Roth, D., Zimak, D. (2003) "Constraint Classification for Multiclass Classification and Ranking." In: Becker, B., Thrun, S., Obermayer, K. (Eds) *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, MIT Press. ISBN 0-262-02550-7
10. Peter Mills (2011). "Efficient statistical classification of satellite measurements". *International Journal of Remote Sensing*. doi:10.1080/01431161.2010.507795.
11. Read, Jesse; Bernhard Pfahringer; Geoff Holmes; Eibe Frank (2009). "Classifier Chains for Multi-label

Classification". *Proc 13th European Conference on Principles and Practice of Knowledge Discovery in Databases and 20th European Conference on Machine Learning* 2009.

12. Tsoumakas, Grigorios; Ioannis Katakis (2007). "Multi-label classification: An overview". *Int J Data Warehousing and Mining* 2007: 1–13.
13. Robert E. Schapire, Yoav Freund, Peter Bartlett and Wee Sun Lee.(1998) "Boosting the margin: A new explanation for the effectiveness of voting methods", *The Annals of Statistics*, 26(5):1651–1686
14. Manfred Warmuth and Karen Glocer and Gunnar Rätsch. Boosting Algorithms for Maximizing the Soft Margin. In the Proceedings of Advances in Neural Information Processing Systems 20, 2007, pp 1585–1592.
15. Tsoumakas, Grigorios; Katakis, Ioannis (2007). "Multi-label classification: an overview". *International Journal of Data Warehousing & Mining* 3 (3): 1–13.
16. Jesse Read, Bernhard Pfahringer, Geoff Holmes, Eibe Frank. Classifier Chains for Multi-label Classification. Machine Learning Journal. Springer. Vol. 85(3), (2011).
17. Vlahavas, Ioannis (2007). "Random *k*-labelsets: An ensemble method for multilabel classification". ECML.
18. Zhang, M.L.; Zhou, Z.H. (2007). "ML-KNN: A lazy learning approach to multi-label learning". *Pattern Recognition* 40 (7).
19. Madjarov, Gjorgji; Kocev, Dragi; Gjorgjevikj, Dejan; Džeroski, Sašo (2012). "An extensive experimental comparison of methods for multi-label learning". *Pattern Recognition* 45 (9).
20. Zhang, M.L.; Zhou, Z.H. "Multi-label neural networks with applications to functional genomics and text categorization". IEEE Transactions on Knowledge and Data Engineering, 18 (10) (2006), pp. 1338–1351.
21. Godbole, Shantanu; Sarawagi, Sunita (2004). "Discriminative methods for multi-labeled classification". Advances in Knowledge Discovery and Data Mining. pp. 22–30.