Multiclass classification

In machine learning, **multiclass** or **multinomial classification** is the problem of classifying instances into one of three or more classes (classifying instances into one of two classes is called binary classification).

While many classification algorithms (notably multinomial logistic regression) naturally permit the use of more than two classes, some are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies.

Multiclass classification should not be confused with multi-label classification, where multiple labels are to be predicted for each instance.

Contents

General strategies

The existing multi-class classification techniques can be categorized into (i) transformation to binary (ii) extension from binary and (iii) hierarchical classification.[1]

Transformation to binary

This section discusses strategies for reducing the problem of multiclass classification to multiple binary classification problems. It can be categorized into *one vs rest* and *one vs one*. The techniques developed based on reducing the multi-class problem into multiple binary problems can also be called problem transformation techniques.

One-vs.-rest

One-vs.-rest[2]:182, 338 (OvR or *one-vs.-all*, OvA or *one-against-all*, OAA) strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes are predicted for a single sample.[3]:182[note 1]

In pseudocode, the training algorithm for an OvR learner constructed from a binary classification learner L is as follows:

Inputs:

- L, a learner (training algorithm for binary classifiers)
- samples X
- labels y where $y_i \in \{1, \dots K\}$ is the label for the sample X_i

Output:

- a list of classifiers f_k for $k \in \{1, \dots, K\}$

Procedure:

- For each k in $\{1, \dots, K\}$
  - Construct a new label vector z where $z_i = y_i$ if $y_i = k$ and $z_i = 0$ otherwise
  - Apply L to X, z to obtain f_k

Making decisions means applying all classifiers to an unseen sample x and predicting the label k for which the corresponding classifier reports the highest confidence score:

$$\hat{y} = \underset{k \in \{1 \dots K\}}{\operatorname{argmax}} f_k(x)$$

Although this strategy is popular, it is a heuristic that suffers from several problems. Firstly, the scale of the confidence values may differ between the binary classifiers. Second, even if the class distribution is balanced in the training set, the binary classification learners see unbalanced distributions because typically the set of negatives they see is much larger than the set of positives.[3]:338

One-vs.-one

In the *one-vs.-one* (OvO) reduction, one trains $K\ (K - 1)\ /\ 2$ binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all $K\ (K - 1)\ /\ 2$ classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.[3]:339

Like OvR, OvO suffers from ambiguities in that some regions of its input space may receive the same number of votes.[3]:183

Extension from binary

This section discusses strategies of extending the existing binary classifiers to solve multi-class classification problems. Several algorithms have been developed based on neural networks, decision trees, k-nearest neighbors, naive Bayes, support vector machines and extreme learning machines to address multi-class classification problems. These types of techniques can also be called algorithm adaptation techniques.

Neural networks

Multiclass perceptrons provide a natural extension to the multi-class problem. Instead of just having one neuron in the output layer, with binary output, one could have N binary neurons leading to multi-class classification. In practice, the last layer of a neural network is usually a softmax function layer, which is the algebraic simplification of N logistic classifiers, normalized per class by the sum of the N-1 other logistic classifiers.

Extreme learning machines

Extreme learning machines (ELM) is a special case of single hidden layer feed-forward neural networks (SLFNs) where in the input weights and the hidden node biases can be chosen at random. Many variants and developments are made to the ELM for multiclass classification.

k-nearest neighbours

k-nearest neighbors kNN is considered among the oldest non-parametric classification algorithms. To classify an unknown example, the distance from that example to every other training example is measured. The k smallest distances are identified, and the most represented class by these k nearest neighbours is considered the output class label.

Naive Bayes

Naive Bayes is a successful classifier based upon the principle of maximum a posteriori (MAP). This approach is naturally extensible to the case of having more than two classes, and was shown to perform well in spite of the underlying simplifying assumption of conditional independence.

Decision trees

Decision tree learning is a powerful classification technique. The tree tries to infer a split of the training data based on the values of the available features to produce a good generalization. The algorithm can naturally handle binary or multiclass classification problems. The leaf nodes can refer to any of the K classes concerned.

Support vector machines

Support vector machines are based upon the idea of maximizing the margin i.e. maximizing the minimum distance from the separating hyperplane to the nearest example. The basic SVM supports only binary classification, but extensions have been proposed to handle the multiclass classification case as well. In these extensions, additional parameters and constraints are added to the optimization problem to handle the separation of the different classes.

Hierarchical classification

Hierarchical classification tackles the multi-class classification problem by dividing the output space i.e. into a tree. Each parent node is divided into multiple child nodes and the process is continued until each child node represents only one class. Several methods have been proposed based on hierarchical classification.

Learning paradigms

Based on learning paradigms, the existing multi-class classification techniques can be classified into batch learning and online learning. Batch learning algorithms require all the data samples to be available beforehand. It trains the model using the entire training data and then predicts the test sample using the found relationship. The online learning algorithms, on the other hand, incrementally build their models in sequential iterations. In iteration t, an online algorithm receives a sample, x_t and predicts its label $\hat{y}_t$ using the current model; the algorithm then receives y_t, the true label of x_t and updates its model based on the sample-label pair: (x_t, y_t). Recently, a new learning paradigm called progressive learning technique has been developed.[4] The progressive learning technique is capable of not only learning from new samples but also capable of learning new classes of data and yet retain the knowledge learnt thus far.[5]

See also

- Binary classification
- One-class classification
- Multi-label classification
- Multiclass perceptron

Notes

1. In multi-label classification, OvR is known as *binary relevance* and the prediction of multiple classes is considered a feature, not a problem.

References

1. Mohamed, Aly (2005). "Survey on multiclass classification methods" (https://www.cs.utah.edu/~piyush/teaching/aly05multiclass.pdf) (PDF). *Technical Report, Caltech*.
2. Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.
3. Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.
4. Venkatesan, Rajasekar; Meng Joo, Er (2016). "A novel progressive learning technique for multi-class classification". *Neurocomputing*. **207**: 310–321. arXiv:1609.00085 (https://arxiv.org/abs/1609.00085). doi:10.1016/j.neucom.2016.05.006 (https://doi.org/10.1016%2Fj.neucom.2016.05.006).
5. Venkatesan, Rajasekar. "Progressive Learning Technique" (http://rajasekarv.wixsite.com/rajasekar-venkatesan/progressive-learning-multi-class).