algorithms

MDPI

Article

The Study of Multiple Classes Boosting Classification Method Based on Local Similarity

Shixun Wang [1,2,*] and Qiang Chen [1]

[1] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China; cq734802349@163.com
[2] Engineering Lab of Intelligence Business & Internet of Things, Xinzheng 451191, China
* Correspondence: wangshixun@htu.edu.cn

Abstract: Boosting of the ensemble learning model has made great progress, but most of the methods are Boosting the single mode. For this reason, based on the simple multiclass enhancement framework that uses local similarity as a weak learner, it is extended to multimodal multiclass enhancement Boosting. First, based on the local similarity as a weak learner, the loss function is used to find the basic loss, and the logarithmic data points are binarized. Then, we find the optimal local similarity and find the corresponding loss. Compared with the basic loss, the smaller one is the best so far. Second, the local similarity of the two points is calculated, and then the loss is calculated by the local similarity of the two points. Finally, the text and image are retrieved from each other, and the correct rate of text and image retrieval is obtained, respectively. The experimental results show that the multimodal multi-class enhancement framework with local similarity as the weak learner is evaluated on the standard data set and compared with other most advanced methods, showing the experience proficiency of this method.

Keywords: multiclass boosting; local similarity; weak learning; loss function; retrieval accuracy

Citation: Wang, S.; Chen, Q. The Study of Multiple Classes Boosting Classification Method Based on Local Similarity. *Algorithms* **2021**, *14*, 37. https://doi.org/10.3390/a14020037

Academic Editor: Panagiotis Pintelas and Ioannis E. Livieris
Received: 15 December 2020
Accepted: 24 January 2021
Published: 26 January 2021

1. Introduction

With the rapid development of the Internet, more and more people are participating in the Internet. People's food, clothing, housing, and transportation are closely related to the Internet. The Internet has also brought a lot of convenience to our lives. For example, many things that are frequently used in daily life, such as online shopping, mobile phone positioning, search engines, etc., have been closely connected with our daily lives. Due to the diversity of things, we are easily dazzled when doing online shopping, for example, and do not know how to quickly choose what we want to buy. Over time different classification techniques have emerged. The traditional classification technique is a separate text classification, which uses the underlying characteristics of the text to show the characteristics of things. Using the classifier to classify data is a better method than the traditional classification method. Using local image features to generate a simple fuzzy classifier to distinguish known categories from other categories [1] represents a new target classification method. Boosting meta-learning is used to find the most representative local features. This method improves the classification accuracy and greatly shortens the learning and testing time. Additionally, by effectively combining a variety of deep neural networks for classification and learning deep internal features from image data sets it can produce the most advanced classification performance [2]. The combination of Boosting algorithm and neural network can use weights to constrain the classification performance of the neural network, thereby improving the performance of the classifier [3]. The Boosting method is not only used for classification, but can also be applied to medicine. The Multi-instance Learning Boost Algorithm (MIL-Boost) establishes a predictive model that can predict the deterioration of insulin resistance early based on the TyG index [4]. This algorithm is widely used in clinical decision support systems. According to the principle of Boosting, we can combine

several weak learners, or find a suitable classifier. Multi-class Boosting classifier-MBC adds a new multi-class constraint to the objective function [5], which can effectively mine different types of features in a unified classification framework. In addition, Boosting combined with weak learners, the number of enhanced iterations can be modeled as a continuous hyperparameter, and fitted with standard techniques to obtain an effective learning algorithm, which can be widely used in regression and classification [6]. AdaBoost, as a representative of the Boosting algorithm, constructs a globally optimal combination of weak classifiers on the basis of sample weights, which greatly improves the classification performance [7]. Given training data, weak learning algorithms (such as decision trees) can be trained to generate weak learners, and these weak learners only need to have better accuracy than random guessing. Training with different training data can get different weak learners. These weak learners act as members and make joint decisions. To obtain information from different weak learners, AdaBoost solves the following two problems: First, how to choose a group of weak learners with different advantages and disadvantages so that they can make up for each other's deficiencies. Second, how to combine the output of weak learners to obtain better overall decision-making performance.

To solve the first problem, AdaBoost allows each newly added weak learners to reflect some new patterns in the data. To achieve this, AdaBoost maintains a weight distribution for each training sample. That is, there is a distribution corresponding to any sample, which indicates the importance of this sample. When measuring the performance of weak learners, AdaBoost considers the weight of each sample. A misclassified sample with a larger weight will contribute a greater training error rate than a misclassified sample with a smaller weight. To obtain a smaller weighted error rate, weak classifiers must focus more on high-weight samples to ensure accurate predictions. By modifying the weight of the sample, the weak learner can be guided to learn different parts of the training sample. AdaBoost is divided into multiple rounds of training. In the first round of training, we update the weight of the sample and train a weak learner so that the learner produces the smallest weighted training error on the weight distribution. In the first round, all samples have the same weight. In each subsequent round, we increase the weight of misclassified samples and reduce the weight of accurately classified samples. In this way, we make each round of weak learners focus more on samples that are difficult to be accurately classified in the previous round.

Now that we have obtained a set of trained weak learners with different strengths and weaknesses, how do we effectively combine them so that mutual advantages complement each other to produce a more accurate overall prediction effect? Each weak learner is trained with different weight distribution. We can see that different weak learners are assigned different tasks, and each weak learner tries its best to complete the given task. Intuitively, when we want to combine the judgments of each weak learner into the final prediction result, if the weak learner performs well in the previous task, we will believe it more, on the contrary, if the weak learner in the previous task performs poorly, we believe it less. In other words, we will combine weak learners in a weighted manner, and assign each weak learner a value indicating the degree of credibility. This value depends on its performance in the assigned task. The better the performance, the greater the value, and vice versa. AdaBoost training will eventually have two situations: one is that the training error is reduced to zero. The other is that no matter how many times of training, there will be no overfitting problem. Each subsequent weak classifier has an accuracy of 0.5. However, these are limited to the same mode. If you want to better apply it in real-life, it is a more cross-modal retrieval application. Each source or form of information can be called a mode. For example, people have a sense of touch, hearing, sight, and smell; the medium of information includes voice; video; text; and a variety of sensors, such as radar, infrared, and accelerometer. Each of the above can be called a mode.

The main method of cross-modal retrieval is to find the relationship between different modes and map different modes to the same subspace, where we can measure the similarity between different modes. Specifically, first, we have some training real columns, each

instance is an image-text pair with a label; then, these real columns are divided into train set and test set, there are unrelated image–text pairs in each set, but all have the same category (that is, the training set and the test set do not have the same image–text pair, but the category is the same); then, during training, we learn a common semantic space from the train set; then, apply the common semantic space to the test set to generate a common feature representation for the instances in the test set; finally, use the common feature representation to compare The degree of similarity of the samples of the two modes is thus matched. Eventually, stronger learners of each modal can generate semantic vectors and apply logistic regression to obtain semantic vectors of images and texts. However, when features have high dimensions and sparse values, logistic regression cannot work effectively. Although image texts essentially represent the same set of semantic concepts, their manifestations differ greatly due to the large differences between different data modes. How to robustly represent the similarity between image and text and accurately measure the similarity is a tricky problem.

To deal with this problem, the existing methods can be divided into two main categories according to the different ways of modeling the corresponding relationship between image and text: one-to-one matching and many-to-many matching.

As shown in Figure 1A, we can see that it is a one-to-one matching method, usually extracting global feature representations of images and texts, such as desks, chairs, computers, illustrations, etc. In the picture, and then using the structure, the objective function of transformation or canonical correlation analysis projects their features into a common space, so that the distance between similar pairs of image texts in the space is close, that is, the similarity is high. However, this matching method is only a rough measure of the global similarity of image text and does not specifically consider which local content of the image text is semantically similar. Therefore, in some tasks that require accurate similarity measurement, such as fine-grained for cross-modal retrieval, the experimental accuracy is often low. In Figure 1B, the many-to-many matching method is shown. The many-to-many matching method is to try to extract multiple local instances from the image text, and then measure the local similarity for multiple paired instances. Fusion results in global similarity. We can see from the figure that there are many local instances extracted, but not all instances extracted by these methods depict semantic concepts. In fact, most of the instances are semantically meaningless and have nothing to do with the matching task. Only a few significant semantic instances determine the degree of matching. Those redundant examples can also be considered as noises that interfere with the matching process of a small number of semantic examples and increase the number of model calculations. In addition, existing methods usually need to explicitly use additional target detection algorithms or expensive manual annotations in the instance extraction process.

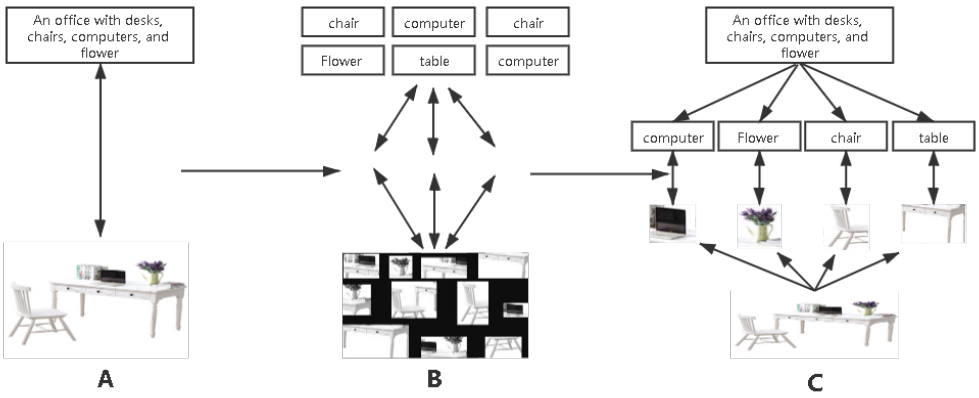

Figure 1. Correspondence between image and text. (**A**) One to one; (**B**) One to many; (**C**) Many to many.

Considering this situation, we propose a multimodal multi-class enhancement framework that uses local similarity as a weak learner. As shown in Figure 1C, its classification accuracy is improved compared to the previous method. To verify the effectiveness of the proposed local similarity as a weak learner multimodal multi-class enhancement framework, we tested the experimental performance of the framework. Moreover, compared with the current best method on two co-opened multimodal databases (Wiki and NUS-WIDE), showing good experimental results.

Our contributions mainly include the following.

1. On the basis of a simple multi-class enhancement framework using local similarity as a weak learner, it is extended to a multimodal multi-class enhancement framework. It can better analyze multi-modal semantic information, so as to obtain better cross-mode retrieval performance.
2. In the selection of weak learners, a suitable weak learner can be selected in each iteration. Specifically, each mode is transformed into a minimization optimization problem. Moreover, the algorithm provides corresponding formula derivation and theoretical analysis.
3. We conducted a large number of data experiments on the Wiki and NUS-WIDE datasets to verify the performance of the Boosting algorithm. Furthermore, compared with several latest methods. Experimental results show that our method has significant advantages for cross-mode retrieval.

The rest of the paper is arranged as follows, and the related work is introduced in Section 2. In Section 3, we introduced our framework and introduced the algorithm flow chart. In Section 4, we did a convergence experiment. It shows that continuously reducing losses will only improve the decision boundary without overfitting the data. The comprehensive experiment is shown in Section 5, and the final conclusion is summarized in Section 6.

2. Related Work

Among the classifiers, the Boosting method is one of the better methods. It is a general and effective method to generate strong learners by combining weak learners. AdaBoost [8] is such a method, which uses gradient descent to minimize the classification risk caused by exponential loss, and then select the if learner and its contribution coefficient for each iteration. Given training data and weak learners can be trained to generate different weak learners, and finally by combining weak learners, a classification model that is perfectly classified in the sample is generated. Weak learners generally use supervised machine learning algorithms, such as decision trees and support vector machines. Combining supervised machine learning algorithms with the boosting process can improve prediction efficiency [9]. The AdaBoost-CNN method integrates AdaBoost and Convolutional Neural Network (CNN), which can process large imbalanced data sets with high precision [10]. Supervised learning is for data that have been labeled. For incompletely labeled data, a semisupervised learning method [11] is required. This method can extract useful information from incompletely labeled data .

In addition, AdaBoost-Cost [12] also combines multiple weak learners into a strong learner, but the requirements for weak learners are not high, and weaker weak learners can also meet the requirements. These are all single mode or two types of modes. Obviously, there are many defects in more than a dozen lives. For example, the computational complexity has not dropped too much; it is difficult to meet people's needs in terms of classification accuracy. Therefore, people began to study multimodal and multi-classification methods, but multimodal and multi-classification methods are more complicated and difficult to implement. TangentBoost [13] studied the design of a robust classifier, which can effectively deal with the typical noise data set and outlier data set in computer vision, and uses the probabilistic heuristic view of classifier design to identify such losses. A set of necessary conditions. Using these conditions, a new enhancement algorithm is obtained. Experiments using data from computer vision problems such as scene classification, target tracking,

and multi-instance learning show that TangentBoost is always better than the previous enhancement algorithm. REBEL [14] is a simple and accurate multi-class enhancement method. This method is quick to train and has less data. A new type of weak learner called local similarity is proposed. This framework can prove to minimize the training error of any data set at an exponential rate. Experiments were conducted on various synthetic and real data sets, which proved that there is a consistent tendency to avoid overfitting, but it is only used on a single mode and shows good results. Mobile robots and self-driving cars rely on multimodal sensor devices to perceive and understand the surrounding environment. In addition to accurate spatial perception, a comprehensive semantic understanding of the environment is essential for efficient and safe operation. A new deep neural network structure, LilaNet [15], is used for point-by-point, multi-category semantic annotation of semi-dense lidar data. An automatic process of large-scale cross-modal training data generation called automatic labeling is proposed to improve the performance of semantic labeling while keeping the workload of manual labeling low.

A direct optimization method for training multi-class boosting [16] is a high-accuracy predictor formed by Boosting combined with a set of weak classifiers with general accuracy. Compared with binary advancing classification, multi-class advancing is more valued. It proposes a new multi-type boost formula. Different from the previous multi-boosting algorithm that decomposes the multi-boosting problem into multiple independent binary boosting problems, it is a direct optimization method for training multiple types of boosting. In addition, by explicitly deriving the dual relationship of the original optimization problem, a completely revised boosting is designed using column generation technology in convex optimization. In each iteration, the weights of all weak classifiers are updated. Moreover, compared with the latest multi-class improvement, the recognition accuracy has been improved to a certain extent. MCBoost [17] proposed a new framework based on multi-dimensional codeword matrix and predictor, which minimizes risk through gradient descent in multi-dimensional function space, with one based on coordinate descent and one based on gradient descent. Both methods are very good multi-class enhancement methods. MMBoosting [18] is a multimodal multi-class enhancement framework that can simultaneously capture the semantic information within the modal and the semantic correlation between the modals, and using multi-class exponential functions and logistic loss function, it obtained two new versions of MMBoost, namely, MMBoost_exp [18] and MMBoost_log [18], which have made great contributions to the cross-modal retrieval enhancement method.

Although considerable progress has been made in the multimodal and multi-class enhancement method, the classification accuracy is not very high. Therefore, from the above development, we need to find a suitable weak learner to replace the existing. Some weak learners can improve the accuracy of multi-classification. Although there are many weak learners, we choose local similarity as the weak learner. The reason is that during each iteration, a suitable weak learner can be selected. In the next section, we mainly introduce the framework we constructed.

3. Our Framework

We define some of our symbols, introduce our enhancement framework, and describe our training process. x represents a scalar, and the bold x represents a vector, where $\mathbf{x} \equiv [x_1, x_2, ...]$, the logical expression $1 \in \{0, 1\}$, the inner product expression is $\langle \mathbf{x}, \mathbf{v} \rangle$, element-wise multiplication product is $\mathbf{x} \odot \mathbf{v}$. In a multi-class classification setting, a data point is represented as a feature vector $\mathbf{x}$ and is associated with a class label y. Each point is composed of d features. We use local similarity as a weak learner, because it supports binary weak learners [19], and its mathematical expression is simple (a closed solution that minimizes loss), and it has a strong empirical performance. To find the optimal predictor of different modes [20], construct a target risk function based on experiments:

$$[R[f, u] = 2\left\langle \sqrt{S_f^T \odot S_f^F}, \mathbf{1} \right\rangle + 2\left\langle \sqrt{S_u^T \odot S_u^F}, \mathbf{1} \right\rangle + \lambda J(f, u)] \quad (1)$$

The first part and the second part are training loss functions of images and text, respectively, and the third part represents the model complexity, and the optimal local similarity is that the loss function and model complexity are minimized at the same time. The problem is to find the minimum value of the objective function. Local similarity returns a vector value output **H**:

$$[\mathbf{H}(x) \equiv \sum_{t=1}^{T} f_t(x)\mathbf{a}_t] \tag{2}$$

The average misclassification error ε can be expressed as

$$\varepsilon \equiv \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}(F(x_n \neq y_n)) \tag{3}$$

Local similarity uses an exponential function to cap the average training misclassification error:

$$[\varepsilon \leq L \equiv \frac{1}{2N}\sum_{n=1}^{N} \langle \exp[y_n \odot \mathbf{H}(X_{\mathrm{n}}), \mathbf{1}]\rangle] \tag{4}$$

where $y_n \equiv 1 - 2\delta_{y_n}$, because it is an additive model [21], the parameters of the previous training are fixed. Each iteration is equivalent to jointly optimizing a new weak learner f and the accumulation vector a. Therefore, the training loss at iteration $I + 1$ can be expressed for

$$L_{I+1} \equiv L_f \equiv \left\langle S_f^T, \exp[-\mathbf{a}] \right\rangle + \left\langle S_f^F, \exp[\mathbf{a}] \right\rangle \tag{5}$$

where f is a given weak learner, and S_f^T and S_f^F are the sum of the classification weights [22] of right and wrong classification, respectively. In this form, it is easy to get the optimal accumulation vector $\mathbf{a}^*$:

$$[\mathbf{a}^* = \frac{1}{2}(In[S_f^T] - In[S_f^T])] \tag{6}$$

The training loss [23] can be optimized as

$$L_f^* = 2\left\langle \sqrt{S_f^T \odot S_f^F}, \mathbf{1} \right\rangle \tag{7}$$

3.1. Calculation of Local Similarity of Two Points

The traditional decision tree compares a single feature with a threshold and outputs +1 or −1. However, our weak learner (Local similarity) uses similarity measures [24] to compare points in the input space. Due to its simplicity and effectiveness, we use the negative squared Euclidean distance $-\left\|x_i - x_j\right\|^2$ as the similarity measure between x_i and x_j. For the local similarity of two points, at a given x_i and x_j, if x_i and x_j are more similar, the input space is positive, and it has the largest absolute effect near x_i and x_j. Therefore, the two local similarities are

$$[f_{ij}(\mathbf{x}) \equiv \frac{\langle \mathbf{d}, \mathbf{x} - \mathbf{m}\rangle}{4\|\mathbf{d}\|^4 + \|\mathbf{x} - \mathbf{m}\|^4}] \tag{8}$$

where $\mathbf{d} \equiv \frac{1}{2}[x_i - x_j], \mathbf{m} \equiv \frac{1}{2}[x_i + x_j]$ There are two modes of local similarity: single point and two points. The single point mode allows any single data point to be isolated, ensuring a basic reduction in loss [25]. However, it essentially leads to pure memory of the training data; imitating the nearest neighbor classifier. By providing the edge style function, the two-point mode increases the ability to better generalize. The combination of these two modes makes it flexible to handle a wide range of classification problems [26]. In addition, in any mode, the function of local similarity is easy to explain, that is, which of these fixed training points is more similar to a given query point.

3.2. Find the Appropriate Local Similarity

Assuming a data set with N samples, there are about N^2 possible local similarities, the following algorithm flowchart can effectively select the appropriate local similarity.

The purpose of the algorithm shown in Figure 2 is to select the appropriate local similarity, that is, to learn a set of predictors, so that the algorithm can mine the semantic information within the modal and the semantic correlation between the modal. Moreover, to make the optimization problem simple, we learn each predictor in turn [27]. First, we load the bimodal data set, initialize the inter-modal parameters, the parameters of the multi-class weak learner, and the number of iterations. The data set parameters include image set, text set and semantic vocabulary, and each semantic class is coded with a different unit vector, thereby representing the data point as a corresponding feature vector. At the beginning of the iteration, the predictors for images and text are initialized. Then, learn each predictor in turn, update the obtained image and text predictor to the latest predictor, increase the number of iterations by 1, and continue to execute the iterative loop until the loop condition is not satisfied. The learning of weak learners is mainly divided into the following steps:

(1) For the selection of the image weak learner, Formula (5) is used to calculate the basic loss L_1 of the weak learner f_1.

(2) Because each non-negative feature value is related to an independent feature vector, let $\mathbf{v}_n$ be the eigenvector corresponding to the nth eigenvalue λ, and define $\mathbf{f}$ as the row vector of the element $f(x_n)$. Therefore, $\mathbf{f}$ can be decomposed into

$$[\mathbf{f} = \langle \mathbf{f}, \mathbf{v}_1 \rangle \mathbf{v}_1 + \sum_{n=2}^{N} \langle \mathbf{f}, \mathbf{v}_n \rangle \mathbf{v}_n] \tag{9}$$

From the above formula, the feature vector $\mathbf{v}_1$ can be calculated, and all points are labeled according to their binarization class label b_n.

(3) Find the optimal local similarity f_i, that is, the point x_i that is closer to the threshold τ, and all the points that meet the conditions i is marked as +1, and the other points are marked as -1. Then, calculate the loss of f_i by Formula (8). Compared with f_i, the one with smaller loss is regarded as the best so far.

(4) Find the point x_j that is most similar to x_j, which can be obtained by the following formula,

$$[x_j = \arg \min_{b_j = -b_i} \{ \|x_i - x_j\|^2 \}] \tag{10}$$

(5) Use the two-point local similarity f_{ij} to calculate the loss. If its performance exceeds the previous best, store the new learner and update the best loss so far.

(6) Find all points that are sufficiently similar to x_j and remove them from the rest of the current iteration. In our implementation, we removed all x_n:

$$[f_{ij}(x_n) \leq f_{ij}(x_j)/2] \tag{11}$$

If all points are deleted, return the best local similarity so far; otherwise, re-find the point x_j that is most similar to x_i. After completing this step, ensure that the best local similarity [28] so far can be sufficient. For the selection of weak learner for text, an analogy to the selection of weak learner for the above image can be obtained.

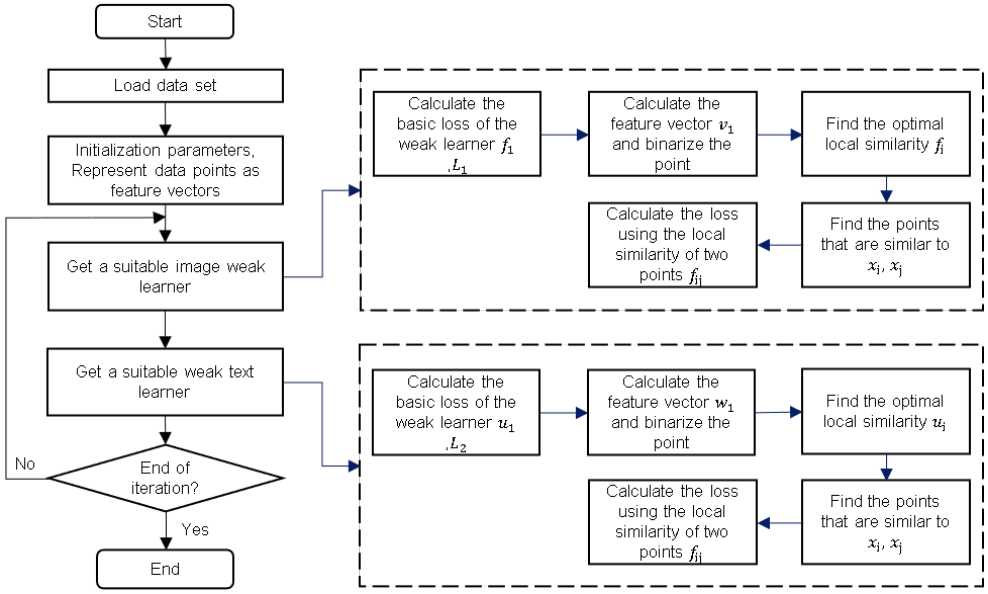

Figure 2. Algorithm flow chart.

4. Convergence Experiment

In this section, we will demonstrate that the continuous reduction in loss will only improve the decision boundary without overfitting the data. To better visualize and intuitively understand what the classifier is doing, we generated a two-dimensional synthetic data set. The synthetic data set is artificially synthesized through the method of clustering. Both numpy and scikit-learn provide the function of random data generation; here, we use makegaussian_quantiles, a subfunction of scikitlearn, to generate grouped multidimensional normal distribution data. The parameters are set as follows. The number of samples generated in the first group is 200, and the number of samples generated in the second group is 300. The other parameters of the two samples are the same, the feature mean is 3 and the covariance coefficient is 2, which generates a 2-dimensional normal distribution, the generated data is divided into 3 groups according to quantiles, then the randomly generated samples and the corresponding categories of the samples are connected to the arrays. The 500 points of data obtained are divided, 2/3 as the training set, and 1/3 as the test set. The statistical characteristics of the synthetic data set are shown in Table 1 below.

Table 1. The summary of Synthetic datasets.

Dataset	Size	Training Set	Testing Set	Classes
Synthetic dataset	500	333	167	3

To make the annotation clear and simplified, we abbreviate the local similarity as LS, and the convergence results are shown in Figure 3.

Figure 3a shows the training loss and training error of the classifier. Perform 500 iterations of training with the test error, and the final loss is 0.1355. Although the number of training is not sufficient, we can see intuitively that as the training error decreases, the test error and loss show a downward trend. Figure 3b performs 1000 iterations on the training loss, training error, and test error of the classifier, and the final loss is 0.0338. Figure 3c performs 1500 iterations on the training loss, training error, and test error of the classifier, and the final loss is 0.0137. Figure 3d performs 2000 iterations on the training loss, training error, and test error of the classifier, and the final loss is 0.0063. From the above four generalization experiments, we can see that even if the training error is reduced to zero, the test error will not increase. Our framework is in the case of different iteration times, when the training loss is in a downward trend, the test error is also such a downward trend, and as the number of training increases, when the error drops to zero, the test error

increases. The closer it is to zero, but not zero, it shows the authenticity of our experiment. In the actual test experiment, the error is certain. In all cases, the classification boundary is suitable for the training data, rather than overfitting, that is, each point is isolated. This can be observed more strictly from the training curve. As the number of iterations increases, the test error is quite stable when it reaches a minimum.

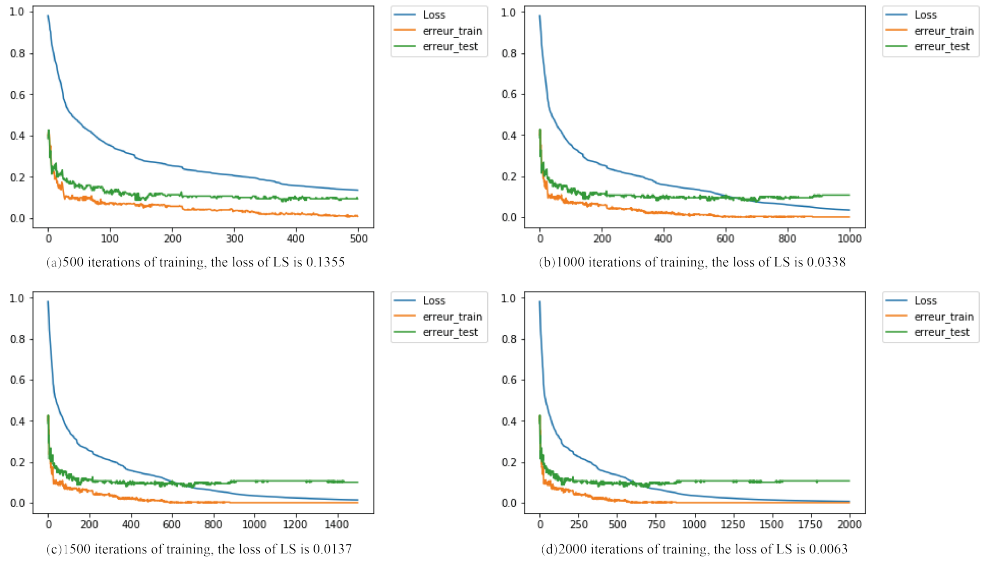

Figure 3. Training results of 500, 1000, 1500, and 2000 iterations, respectively, on a synthetic data set of 500 points.

5. Data Experiment

In this section, we will conduct extensive experiments on real data sets containing image modes and text modes, and compare the proposed local similarity framework with several existing unimodal cross-modal retrievals. The enhanced methods [29] are compared, where the query and retrieval objects come from different modes.

5.1. Experimental Settings

The benchmark dataset [30] contains two datasets: Wiki and NUS-WIDE. The Wiki dataset contains 2866 multimodal documents, all of which are image–text pairs combined from Wikipedia articles. For each document, the 128-dimensional codebook vector represents the underlying features of the image, and the LDA probability distribution of 10 topics represents the underlying features of the text. The data set is divided into a training set of 2173 documents and a test set of 693 documents. In the NUS-WIDE dataset, each example contains an image and its corresponding text label, and is associated with one or more labels from 81 semantic categories. We randomly selected 25,000 multimodal examples. For each example, a 500-dimensional feature vector and a 1000-dimensional label codebook vector were provided to represent images and text. We selected 5000 multimodal examples as the training set and 1250 multimodal examples as the test set. The statistical characteristics of the two data sets are shown in Table 2 below.

Table 2. The summary of Wiki and NUS-WIDE datasets.

Dataset	Size	Training Set	Testing Set	Classes
Wiki	1866	2173	693	10
NUS-WIDE	25,000	5000	1250	10

Then, we used MAP to evaluate the performance of the experimental results. MAP is a standard numerical metric for evaluating cross-mode retrieval performance. That is, the average value of AP; we need to calculate AP first, and then average it. The larger the

MAP value, the better the performance. Given a query and a set of w retrieval examples, the average accuracy is as below.

Where Precision (i) is the Precision value, and E is the number of relevant examples in the search. In the experiment, the value of w is set to 50 or the number of all retrieved samples, that is, w = 50 or w = all. From the above formula, MAP is obtained by averaging all the queried AP values. In addition, the Precision–Recall (PR) curve is also shown. For the experimental data of text and images, we use posterior probability to achieve the combination of the two. We obtain a strong predictor on the training set, then we use the learned predictor to obtain the semantic vector of the examples in the test set, and finally we use the centered normalized correlation [31] to sort the retrieved samples. It is worth noting that the query set and retrieval set are both test sets. Due to the randomness of data segmentation, we run the evaluation method 10 times and report the average result. In our local similarity framework, there is an important parameter that controls the impact of different risks on the objective function. In the learning process, our algorithm adopts a suitable optimization strategy, that is, alternate gradient descent of each mode to learn semantic information within and between modes. If the matching condition is too large, the optimization algorithm tends to reduce the cross-modal risk, thereby ignoring the semantic information of each modal. Then, use the above parameter settings in the next experiment.

5.2. Experiments on the Wiki Dataset

Figure 4 shows the cross-modal retrieval performance of our proposed local similarity and comparison method on the Wiki dataset, including the mapping value [32] of the text retrieval image and its average value. From the figure below, we can intuitively see that the method we propose is always better than the comparison method in terms of retrieval results. The purpose of semantic information within a modal is to reflect the distinctive abstraction of each modal, and the focus of semantic association between modals is to maintain consistency between different models, so combining them is beneficial to improve retrieval performance. On the other hand, by complementing the corresponding modal semantic associations, the intra-modal semantic vector of low-quality object mapping can be enhanced.

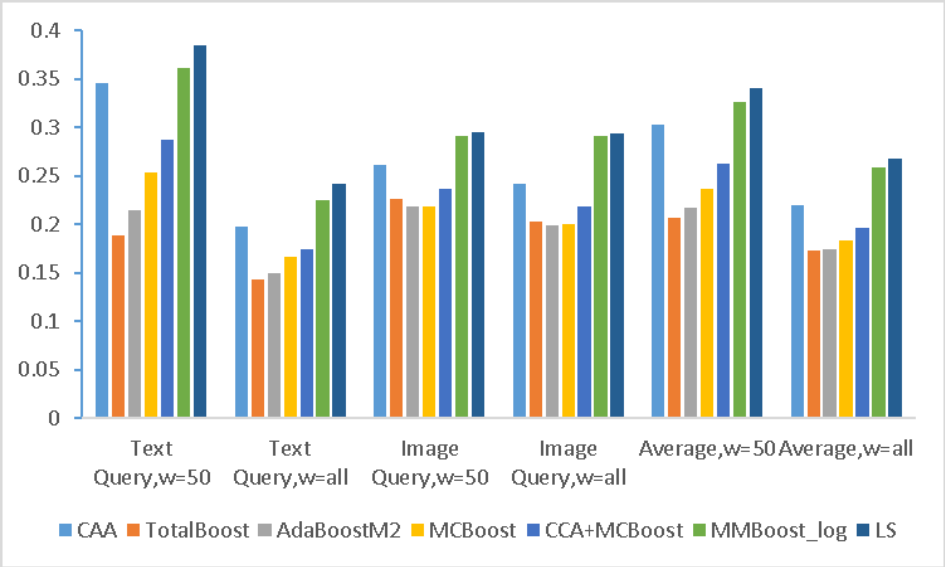

Figure 4. Performance comparison on the Wiki dataset.

For a more detailed analysis, the corresponding PR curves are shown in Figures 5 and 6. From the PR curve, we can see that in these two tasks, our proposed method once again surpasses similar methods, which is consistent with the results in Figure 4. Specifically, this improvement is substantial and occurs at most levels of memory, meaning better

accuracy and generalization capabilities. The accuracy curve can reflect the influence of the number of search objects on the search accuracy. Obviously, the accuracy curve of local similarity is always higher than the compared method, which means that when the number of retrieved objects is equal, more related objects can be returned. The experiment on the Wiki dataset. For text and image queries, the MAP scores of different methods are shown in Figure 4, and the corresponding curves are drawn in Figures 5 and 6. From the figure, we can see that the performance of our algorithm is better than the comparison method.

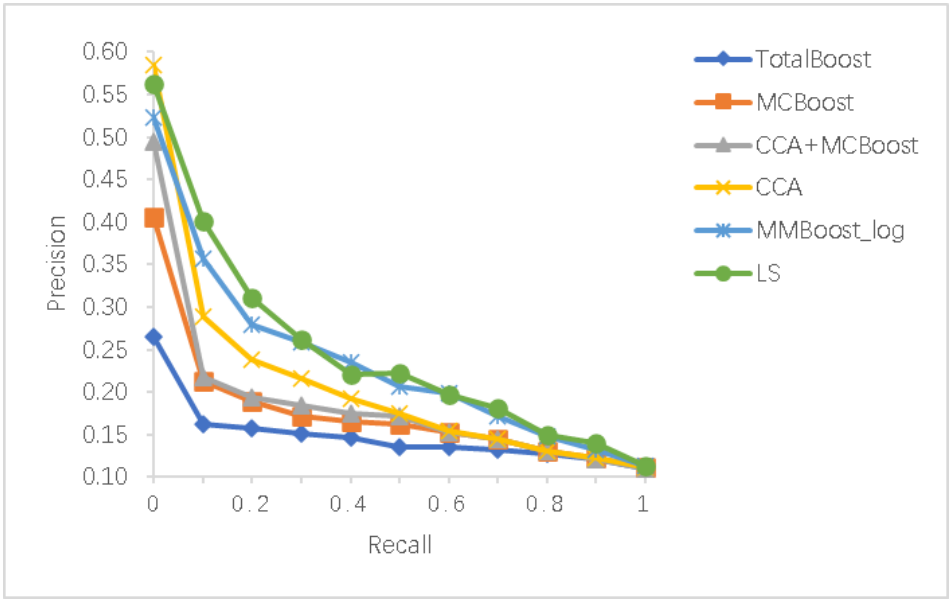

Figure 5. Precision–recall curves for text retrieval images.

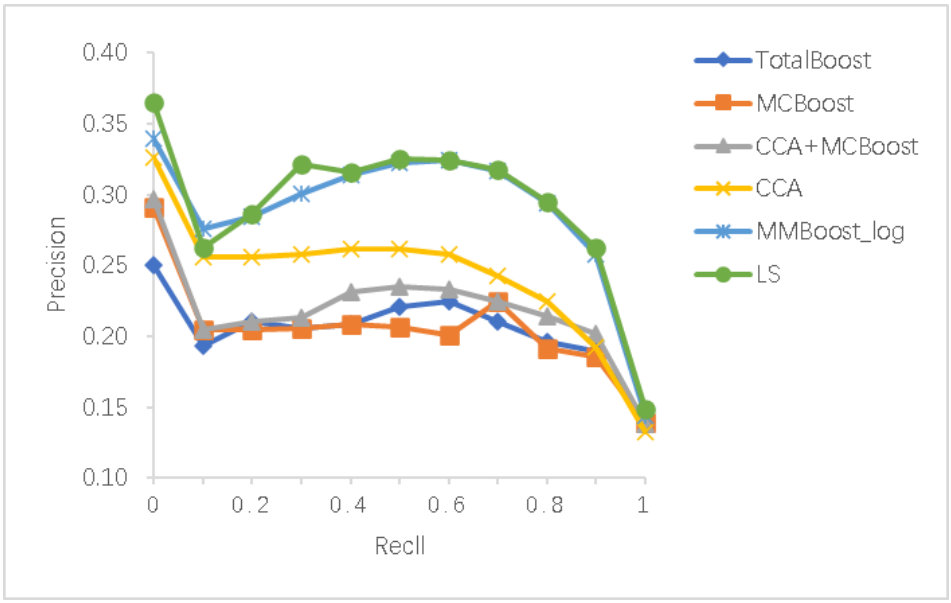

Figure 6. Precision–recall curves for image retrieval text.

5.3. Experiments on the NUS-WIDE Dataset

Figure 7 shows the performance comparison of several state-of-the-art multimodal methods on the NUS-WIDE data set. From the figure, we can see that in terms of performance, MCBoost and MMBoost_exp have a certain performance improvement, but compared with them our method, Local Similarity (LS) has better performance.

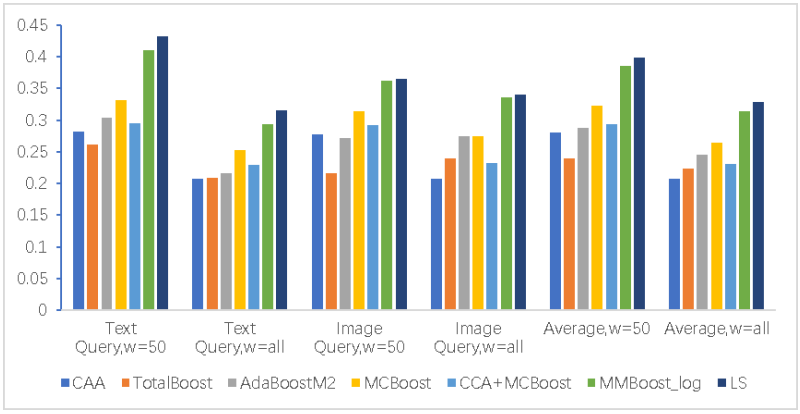

Figure 7. Performance comparison on the NUS-WIDE dataset.

To deal with out-of-sample data, we randomly select for text and image queries, the corresponding curves of MAP scores for different methods are shown in Figures 8 and 9. From the figure, we can see that the performance of our local similarity algorithm (LS) is still better than the comparison method. This is consistent with the Wiki dataset. Specifically, when w = all, our algorithm has an average MAP score of 0.328, which is about 4.5% higher than MMBoost_log. The above experiments show that the local similarity framework has certain advantages due to the combination of semantic information within modes and semantic associations between modes.

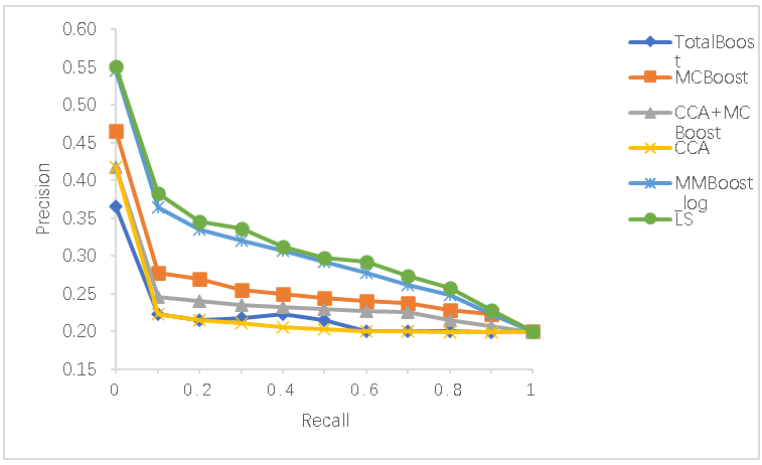

Figure 8. Precision–recall curves for text retrieval images.

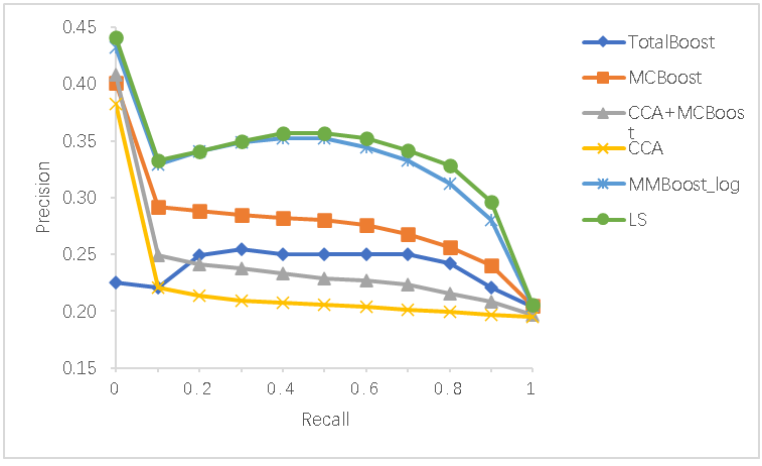

Figure 9. Precision–recall curves for image retrieval text.

6. Conclusions

So far, we have mainly focused on designing a multimodal multi-class enhancement framework for cross-modal retrieval, which utilizes a simple weak learner called local similarity. To achieve mutual retrieval of text and images, these learners have a clear and easy-to-understand function to test the similarity between query points and some predefined samples. In the selection of weak learners [33], a suitable weak learner can be selected for each iteration, and the use of a simple multi-type enhancement framework with local similarity as a weak learner in multimodality is realized. We have proved that the framework adheres to the theoretical guarantee that the training loss is minimized at an exponential speed and the training error is due to the upper limit of the loss [34]. Finally, we compared our method with several other state-of-the-art methods, and there is a certain improvement in most data sets. We compared the current best method on two co-opened multimodal databases (Wiki and NUS-WIDE). We can see that our method has steadily improved performance on different data sets.

In the future, we will work on the following two aspects: First, promote semisupervised or unsupervised learning methods to improve recognition accuracy. Second, find a more suitable weak learner, to improve the accuracy of the forecast.

Author Contributions: Conceptualization, both authors; methodology, Q.C.; software, validation, Q.C.; formal analysis, all authors; writing—original draft preparation, Q.C.; writing review and editing, Q.C.; visualization, Q.C.; project administration, S.W. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Natural Science Foundation of China (Nos. 11702087 and U1904123), the Natural Science Foundation of Henan Province (No.162300410177), and the Key Program of Higher Education Institutions of Henan Province (No. 17A520040).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Korytkowski, M.; Rutkowski, L.; Scherer, R. Fast image classification by boosting fuzzy classifiers. *Inf. Sci.* **2016**, *327*, 175–182. [CrossRef]
2. Zhang, F.; Du, B.; Zhang, L. Scene classification via a gradient boosting random convolutional network framework. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1793–1802. [CrossRef]
3. Ioannis, E.L.; Pintelas, P. On ensemble techniques of weight-constrained neural networks. *Evol. Syst.* **2020**, 1–13. [CrossRef]
4. Bernardini, M.; Morettini, M.; Romeo, L.; Frontoni, E.; Burattini, L. Early temporal prediction of type 2 diabetes risk condition from a general practitioner electronic health record: A multiple instance boosting approach. *Artif. Intell. Med.* **2020**, *105*, 101847. [CrossRef] [PubMed]
5. Zhang, B.; Yang, Y.; Chen, C.; Yang, L.; Han, J.; Shao, L. Action recognition using 3d histograms of texture and a multi-class boosting classifier. *IEEE Trans. Image Process.* **2017**, *26*, 4648–4660. [CrossRef] [PubMed]
6. Aravkin, A.Y.; Bottegal, G.; Pillonetto, G. Boosting as a kernel-based method. *Mach. Learn.* **2016**, *108*, 1951–1974. [CrossRef]
7. Nagahashi, H.; Wu, S. Analysis of generalization ability for different adaboost variants based on classification and regression trees. *J. Electr. Comput. Eng.* **2015**, *2015*, 835357. [CrossRef]
8. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
9. Praveena, M.; Jaiganesh, V. A literature review on supervised machine learning algorithms and boosting process. *Int. J. Comput. Appl.* **2017**, *169*, 32–35. [CrossRef]
10. Taherkhani, A.; McGinnity, T.M. AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing* **2020**, *404*, 351–366. [CrossRef]
11. Livieris, I.; Kiriakidou, N.; Kanavos, A.; Tampakas, V.; Pintelas, P. On ensemble ssl algorithms for credit scoring problem. *Informatics* **2018**, *5*, 40. [CrossRef]
12. Mukherjee, I.; Schapire, R.E. A theory of multiclass boosting. *J. Mach. Learn. Res.* **2011**, *14*, 437–497.
13. Masnadi-Shirazi, H.; Mahadevan, V.; Vasconcelos, N. On the design of robust classifiers for computer vision. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
14. Appel, R.; Perona, P. A simple multi-class boosting framework with theoretical guarantees and empirical proficiency. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
15. Piewak, F.; Pinggera, P.; Schfer, M.; Peter, D.; Schwarz, B.; Schneider, N.; Enzweiler, M.; Pfeiffer, D.; Zollner, M. Boosting lidar-based semantic labeling by cross-modal training data generation. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.

16. Shen, C.; Hao, Z. A direct formulation for totally-corrective multi-class boosting. In Proceedings of the Computer Vision & Pattern Recognition, Providence, RI, USA, 20–25 June 2011.
17. Saberian, M.; Vasconcelos, N. *Multiclass Boosting: Theory and Algorithms*. In proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–14 December 2011.
18. Wang, S.; Dou, Z.; Chen, D.; Yu, H.; Li, Y.; Pan, P. Multimodal multiclass boosting and its application to cross-modal retrieval. *Neurocomputing* **2019**, *357*, 11–23. [CrossRef]
19. Allwein, E.L.; Schapire, R.E.; Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. *J. Mach. Learn. Res.* **2000**, *1*, 113–141.
20. Coxeter, H.S.M. *Regular Polytopes*; Dover Publications: New York, NY, USA, 1973.
21. Dietterich, T.G.; Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell. Res.* **1995**, *2*, 263–286.
22. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; Wiley: Hoboken, NJ, USA, 2004.
23. Günther, E.; Pfeiffer, K.P. Multiclass boosting for weak classifiers. *J. Mach. Learn. Res.* **2005**, *6*, 189–210.
24. Guermeur, Y. Vc theory of large margin multi-category classifiers. *J. Mach. Learn. Res.* **2007**, *8*, 2551–2594.
25. Wiley, R. *An Introduction to Derivatives*; Harcourt Brace College Publishers: San Diego, CA, USA, 1999; ISBN 9780030244834.
26. Mason, L.; Baxter, J.; Bartlett, P.; Frean, M. *Boosting Algorithms as Gradient Descent*. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999; pp. 512–518.
27. Mease, D.; Wyner, A. Evidence contrary to the statistical view of boosting. *C4 Programs Mach. Learn.* **2008**, *9*, 131–156.
28. Saberian, M.J.; Masnadi-Shirazi, H.; Vasconcelos, N. TaylorBoost: First and second-order boosting algorithms with explicit margin control. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011.
29. Schapire, R.E.; Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* **1999**, *37*, 297–336.
30. Vapnik, V.N. Statistical learning theory. *Encycl. Ences Learn.* **1998**, *41*, 3185.
31. Sun, Y.; Todorovic, S.; Li, J.; Wu, D. Unifying the error-correcting and output-code AdaBoost within the margin framework. Machine Learning. In Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, 7–11 August 2005.
32. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
33. Shen, C.; Lin, G.; Hengel, A.V.D. Structboost: Boosting methods for predicting structured output variables. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2089–2103. [CrossRef] [PubMed]
34. Appel, R.; Burgos-Artizzu, X.; Perona, P. Improved Multi-Class Cost-Sensitive Boosting via Estimation of the Minimum-Risk Class. *arXiv* **2016**, arXiv:1607.03547.