Computer analysis of visual image similarity

Ksenia Zhagorina, Alexey Buslavyev

Ural Federal University, Institute of Mathematics and Computer Sciences,
Yekaterinburg, Russia
ksuhka-zhagorina@yandex.ru, buslavyewal@mail.ru

Abstract. This paper is a description of image analysis and machine-learning algorithms used for multiclass image classification in the process of our participation in the ImageCLEF 2012 competition. Our goal was to develop an application that could successfully determine the location of a mobile robot using the visual information provided by a camera placed on the robot. The resulting application uses machine-learning methods that improved on self-organizing Kohonen maps and classification algorithms based on probabilistic models. The result of our work was an application that was able to correctly classify 86 percent of input images presented in the ImageCLEF Robot Vision task.

Keywords: visual similarity analysis, image analysis, image search, classification problem, multiclass classification, computer vision, maximum likelihood method, probabilistic model, Kohonen network.

1 Introduction

In this paper we present the research performed in the context of our participation in the ImageCLEF 2012 competition (RobotVision task)[1]. The task is a multiclass image classification challenge: the goal is to assign each of the input images to a particular class.

The official definition of the task is as follows: "The fourth edition of the RobotVision challenge will focus on the problem of multi-modal place classification. Participants will be asked to classify functional areas on the basis of image sequences, captured by a perspective camera and a kinect mounted on a mobile robot within an office environment. The test sequence will be acquired within the same building and floor but there can be variations in the lighting conditions (sunny, cloudy, night) or the acquisition procedure (clockwise and counter clockwise).

These are all the rooms/categories that appear in the database: Corridor, Elevator Area, Printer Room, Lounge Area, Professor Office, Student Office, Visio Conference, Technical Room, Toilet."

[1] ImageCLEF – Image retrieval in CLEF (http://imageclef.org/)

For the test runs we only used images produced by the perspective camera. Each test sequence contains about 2500 images. Quality of classification algorithm we evaluated as the percentage of images correctly classified by this algorithm.

2 Use of Kohonen networks for image classification

2.1 Self-organizing Kohonen maps

Self-organizing Kohonen networks[2] or Kohonen maps are a type of artificial neural network that is trained using unsupervised learning. Training material for the network is a set of vectors in space $\mathbb{R}^n$. A self-organizing map consists of components called nodes or neurons. A reference vector of the same dimension as the input vectors is associated with each node. The typical arrangement of nodes is a regular spacing in a rectangular grid.

Kohonen networks are competitive (i.e. based on the "winner takes all" principle): the neuron whose reference vector is most similar to the input is considered the best matching unit (BMU) and is used during the training process and to produce the output signal of the network.

The map is initialized with random vectors for each neuron. For each iteration during the training phase, a BMU is determined. The reference vectors of the best matching neuron and its neighbors are modified according to the formula:

$$\vec{w}_{new} = \vec{w}_{old} + \varepsilon * \left(\vec{d} - \vec{w}_{old}\right) = (1 - \varepsilon) * \vec{w}_{old} + \varepsilon * \vec{d}$$

where $\vec{w}$ is the reference vector of a neuron, $\vec{d}$ is the input vector from the training sample. The variable ε depends inversely on the training step and the distance between the BMU and neuron being modified.

The output signal of the network is the Euclidean distance between the input vector and its BMU.

In order to use Kohonen networks for image classification, images needed to be presented as sets of vectors. For this for each image we generated a sample vector set based on the image's domains (overlapping areas of a certain size). Various domain parameters were used as the vectors' coordinates, for example: expected value of pixel color, dispersion of pixel color, standard deviation of pixel color from color of the domain's center, etc.

After training using a sample set of vectors built from one or more images a Kohonen map becomes representative for these images and can be used to determine the degree of similarity between a new input image and the images used in the training process. The degree of similarity (or distance between image and map) is defined as the arithmetic mean of the output signals obtained for each vector in the vector set built from this image.

---

[2] From Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/Kohonen_map)

Our test application used the following algorithm for classification: 1) Kohonen maps are trained on the provided set of already classified images - one map for each class. 2) A Kohonen map trained on all images of some class is considered representative for this class. 3) A new input image is assigned to the class that has the least distance between the class's representative map and the image.

2.2 Increasing Kohonen networks

It was determined that classical self-organizing Kohonen maps are not well suited for the task of image classification due to the nature of the data, primarily due to large amounts of training data. Standard fixed size Kohonen maps use a monotonically decreasing learning coefficient: this means that the maps actively train in the beginning of the training set. Each image has less effect on the state of the network than the previous one; therefore the network contains much less information regarding the final images of the training set. An alternative would be not using a decreasing learning coefficient, but this would lead to information belonging to the first images to be overwritten.

Additionally, because we did not have any need for visualization of data, the arrangement of nodes in a rectangular grid and the ability of neurons to affect each other were found not to be particularly useful.

In order to solve the problems identified earlier, we abandoned the traditional arrangement of nodes in favor of a linear one and added the ability to increase the number of neurons during training. This way a Kohonen map degenerated into a linear structure of reference vectors that grew during the training process. The basic principle remained the same – "winner takes all". During training each vector of the sample affected only the best matching unit (BMU).

This modification allowed us to process large amounts of training data, while avoiding loss of critical information.

The detailed training process for a growing Kohonen map is as follows:

1. At some stage of training a new vector $\vec{d}$ from the training sample is presented to the neural network.
2. Vector $\vec{w}$ is the best matching unit in network for input vector $\vec{d}$. In the capacity of distance between vectors Euclidean distance is used.
3. If the distance between $\vec{d}$ and $\vec{w}$ is greater than a certain threshold value, $\vec{d}$ is added to the network as new reference vector. The threshold value affects the accuracy of the Kohonen network.
4. Otherwise, vector $\vec{w}$ is modified according to the formula:

$$\vec{w}_{new} = \vec{w}_{old} + \varepsilon * (\vec{d} - \vec{w}_{old})$$

So after training a growing Kohonen network contains characteristic elements of the vector sample constructed from images of a certain class. Information from the training sample is not lost at any stage of training.

2.3 Hierarchical Kohonen networks.

Poor running time performance was another problem experienced when using standard Kohonen networks for image classification. Growing Kohonen networks solved the problem of training on large amounts of data but did not improve poor performance experienced during both training and classification stages.

An hierarchical Kohonen network is a structure that combines several growing Kohonen networks. Each node in the first level network is associated with another Kohonen network (called a second level network). For each iteration during the training phase, the first level network is trained on vector $\vec{x}$ that is characteristic of the whole image. After that, the second level Kohonen network associated with the best matching node of the first level network is trained on a vector sample set built form the same image.

It is also possible to modify the hierarchical Kohonen network so that both first and second level networks are trained on the same data, but the first level network in this case has a lower accuracy.

Using hierarchical Kohonen networks significantly improves the application's performance, but slightly reduces the efficiency (it reduces the percent of correctly classified images).

2.4 Summary of Kohonen networks

Kohonen networks were the main tool used for image classification and implemented in our submission for ImageCLEF 2012. As input vectors we used various characteristics of the images' domains. The best result of approximately 73% correctly classified images was obtained using growing Kohonen networks and the following characteristics as vector coordinates for the images' domains (in XYZ color space):

- Expected value of pixel color,
- Variance of pixel color,
- Standard deviation of pixel color from the color of the domain's center,
- Standard distance between the colors of neighboring pixels,
- Maximum horizontal color difference,
- Maximum vertical color difference.

3 Maximum-likelihood method

3.1 Principles

In statistics, maximum-likelihood estimation[3] (MLE) is a method of estimating an unknown parameter by maximizing the likelihood function.

[3] From Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/Maximum_likelihood)

Suppose we have a sample $x = (X_1, X_2, \dots, X_n)$ from distribution P_θ, where $\theta \in \Theta$ – is unknown parameter. Let $f(x\,|\theta) : \Theta \to \mathbb{R}^n$ is the likelihood function. Point estimate $\bar{\theta} = \bar{\theta}(x) = \arg\max_{\theta\in\Theta} f(x|\theta)$ is named the maximum likelihood estimate of the parameter θ. Thus the maximum likelihood estimate is an estimate, that maximizes the likelihood function for a fixed realization of the sample.

When applied to the task of image classification, the sample x is the set of all images in the sequence, the set Θ is the set of possible classes and the likelihood function $f(\theta)$ is determined by the choice of probabilistic model of the data. In the simplest case, when using only Kohonen networks, the function $f(\theta)$ is defined as

$$f(X_1, X_2, \dots, X_n\,|\theta = \theta_1, \theta_2, \dots, \theta_n) = \sum_{i=1}^{n} \frac{1}{\rho(X_i, \theta_i)}$$

where $\rho(X_i, \theta_i)$ is the distance between image X_i and representative network of class θ_i. Consequently, when defined in this way, the likelihood function reaches its' maximum at

$$\theta_i = \arg\min_{j=1,m} \rho(X_i, j)$$

i.e. image belongs to the class for which the representative network is nearest to this image.

3.2 Joining probabilistic model

One of the important features of the ImageCLEF RobotVision task is the fact that the input images that are to be classified are produced in sequence by a moving robot. This means that images depend on preceding ones; most importantly this means that the sample cannot contain a single image from a particular class. Every image has to belong to a sequence of multiple images of the same class that precede or follow it. This allows us to construct the following model.

We define a function $P: R^m \to [0;1]^m$, where m is the number of classes. The function P converts the distance from an image to representative networks of classes to the probability that this image belongs to a certain class.

$$P(d_{i1}, d_{i2}, \dots, d_{im}) = (p_{i1}, p_{i2}, \dots, p_{im}) \quad : \quad p_{ij} = \frac{u_{ij}}{\sum_j u_{ij}}$$

where

$$u_{ij} = \left(\frac{max - d_{ij}}{max - min}\right)^2,\ max = max_j\, d_{ij} + \varepsilon,\ min = min_j\, d_{ij}$$

ε - is a negligible quantity that is necessary so that no probability is equal to 0.

p_{ij} – is the probability that the i-th image belongs to j-th class regardless of which classes the neighboring images belong to.

$$D_{ij} = p_{ij} * p_{i-1,j} * p_{i-2,j} * \dots * p_{i-k,j}$$

– is the probability that the i-th image and k preceding it all belong to j-th class.

$$U_{ij} = p_{ij} * p_{i+1,j} * p_{i+2,j} * \dots * p_{i+k,j}$$

– is the probability that the i-th image and k following it all belong j-th class.

$$P_{ij} = \max(D_{ij}, U_{ij})$$

– is the probability that the i-th image belongs to j-th class, taking into account the probability that neighboring images also belong to the j-th class.

Thus the class containing the i-th image can be determined using the formula:

$$j^* = argmax(P_{ij})$$

This model allows us to avoid «wavelets» - small groups of incorrectly classified images.

3.3 Separating probabilistic model

This model incorporates additional information about how the images can be divided into groups and what classes can follow each other. In other words, this model takes into account information about the adjacency of rooms shown in the images. What rooms are adjacent can easily be determined from the training set, again using the continuity of the sequence of images.

Let us consider the more general case: suppose we have a matrix $\{A_{jk}\}_{j,k=1,m}$, where A_{jk} – prior probability that classes j and k are adjacent (in our case j and k are numbers corresponding to rooms).

The fact that adjacent images a and b belong to different classes j and k is equivalent to the following: images that precede a belong to the class j; images follow b belong to the class k; classes j and k are adjacent. Therefore the probability that images a and b belong to different classes j and k can be expressed as:

$$Diff(a, j, b, k) = D_{aj} * U_{bk} * A_{jk} \quad [4]$$

Let us assume that the transitions between classes occur in points of the finite set:

$$(a_1, b_1, (j_1 \to j_2)), (a_2, b_2, (j_2 \to j_3)), \dots, (a_s, b_s, (j_s \to j_{s+1}))$$

The probability of this is the probability that a_i and b_i images belong to different classes j_i and j_{i+1}, and that all images between neighboring transition points belong to the same class.

$$P = (\prod_{i=1}^{s} Diff(a_i, j_i, b_i, j_{i+1})) * (\prod_{c=0, c \neq a_i, b_i}^{N} (P_{cj})), \text{ where } j = j_i \text{ if } b_{i-1} < c < a_i$$

[4] Notation D_{aj}, U_{bk}, P_{cj} were described in section 3.2 Joining probabilistic model

Now we only need to maximize the function P on all possible transition points. This task can be simplified considerably if we find the logarithm of the equation. The maximum of function P will be achieved in the same points as the maximum of function log(P).

$$log(P) = \sum_{i=1}^{s} (log(D_{aj}) + log(U_{bk}) + log(A_{jk})) + \sum_{c=0, c\neq a_i, b_i}^{N} log(P_{cj})$$

This problem can be reduced to the problem of finding the maximum weight path in a directed graph without cycles. The graph is constructed as follows: each image a is represented as m nodes $a_1, a_2, \dots, a_m$, where m is number of classes. Two special nodes are also added: node s is used as a start node in graph search and node t is a final node. The set of arcs contains follow arcs:

$$E = \{(a_j, b_k) : j, k = 1, .., m, \quad \text{where image } a \text{ precedes image } b\}$$

and also arcs from node s to all nodes a_j, formed from the first image in the sequence, and arcs from all nodes b_k, formed from the last image in the sequence to node t.

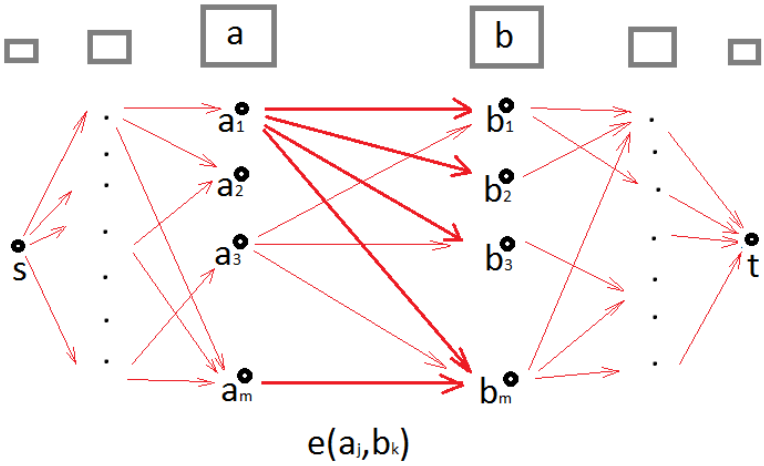

Fig. 1. The graph constructed from a sequence of images.

We introduce a weight function:

$$c(e) = c(a_j, b_k) = \begin{cases} 0.5 * (\log(P_{aj}) + \log(P_{bj})), & k = j \\ 0.5 * (log(D_{aj}) + log(U_{bk})) + log(A_{jk}), & k \neq j \end{cases}$$

$$c(s, a_j) = 0, \quad j = 1, .., m \quad a \text{ is the first image}$$

$$c(b_k, t) = 0, \quad k = 1, \dots, m \quad b \textit{ is } t\text{he last image}$$

In a given network without cycles of negative length, the maximum weight path from node s to node t can be determined using the Bellman-Ford algorithm for example. It should be noted that only the nodes located to the left of a_j (i.e. formed form images that precede the image a_j) can affect the maximum weight path to the node a_j. i.e. formed form images that precede the image a_j.

The maximum weight path in the graph determines the optimal classification of images.

3.4 Summary of probabilistic models

The correct choice of a probabilistic model can significantly influence the result rate in multiple classification tasks. The effect of applying probabilistic methods to a particular task can be significantly influenced by the precision of the function used as a probability estimate; the more precise the estimate is, the more effective the method in general is.

In the process of developing our test application, we had the opportunity to test both probabilistic models. The joining probabilistic model yielded a rate of 81.3% correctly classified images, while use of the separating probabilistic model allowed us to improve our result to 86% correctly classified images.

4 Conclusion

In the course of our research we investigated in detail, improved and applied in practice such methods of data processing as self-organizing Kohonen maps and their modifications and methods of classification based on the maximum-likelihood method.

We were able to create a test application that analyzed images and used the obtained visual information about the surrounding space (without building a map of the premises) to correctly determine the robot's location (in most cases).

The best result of 86% correctly classified images was achieved using growing Kohonen networks and a separating probabilistic model.

Our test application took part in the ImageCLEF-2012 contest (RobotVision task) and placed fourth. We are mostly satisfied with our result and believe that the approach presented in this paper is new and can be generalized and applied to various other image classification tasks.