ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Incremental feature computation and classification for image segmentation

Guillaume Bernard[1,2], Michel Verleysen[2], and John A. Lee[1,2] *

1 - Molecular Imaging, Radiotherapy, and Oncology – IREC
Université catholique de Louvain, Belgium

2 - Machine Learning Group – ICTEAM
Université catholique de Louvain, Belgium

Abstract. Image segmentation problems can be solved with classification algorithms. However, their use is limited to features derived from intensities of pixels or patches. Features such as contiguity of two regions cannot be considered without prior knowledge of one of the two class labels. Instead of stacking various classification algorithms, we describe an incremental classifier that works in a space where features are progressively evaluated. Experiments on artificial images demonstrate the capabilities of this incremental scheme.

1 Introduction

Various approaches can address the problem of image segmentation. Histogram thresholding [1], pixel or patch clustering [2], gradient peak detection with active contours [3] or watersheds [4] are only a few of them. Many of these methods are unsupervised, though they can take into account some a priori information, such as the expected region shape, size, and edge smoothness. As a matter of fact, supervised segmentation raises less interest, mainly because usual classification algorithm can only deal with *intrinsic* features, such as pixel coordinates, pixel intensities, or patch textures. On the other hand, *extrinsic* features that describe the relationships between two or more classes in the image are difficult to take into account. For instance, let us consider a feature such as the spatial distance to the region of class Y in the image. When training a classifier, this feature can be trivially computed since the labels of all regions are known in a pre-segmented image. In contrast, in a test image, measuring this distance requires some region to be already given the label Y. A pragmatic solution to take benefit of extrinsic features consists in stacking at least two classifiers. The first one involves only intrinsic features. The resulting partial classification can then serve to compute a first batch of extrinsic features, which are fed into a second classifier, and so on. This incremental process has been investigated in [5], for example.

In this paper, we suggest a more generic approach, where the multiclass problem is first divided into several binary classification problems (one class versus all others). Binary classifiers based on the principle of the K-nearest neighbors (KNN) tackle these problems repeatedly in an iterative way. At each iteration, precomputed feature relevance factors that reflect the ability of a given

*J.A.L. is a Research Associate with the Belgian fund of scientific research F.R.S.-FNRS.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

feature to discriminate a given class are used to select a binary classifier. Hence, this incremental process attempts to solve first the simplest binary classification problems, in order to enrich as quickly as possible the pool of known extrinsic features. Eventually, at the end of this incremental process, a multiclass classifier is used with all features. The efficacy of the approach is demonstrated in a few image segmentation tasks.

The rest of this paper is organized as follows. Section 2 introduces the notations for intrinsic, extrinsic, and known features. Section 3 describes the incremental procedure for feature computation and partial classification, as well as the final multi-class classification, which improves the accuracy. Section 4 reports and discusses the experimental results. Finally, Section 5 draws the conclusions and sketches some perspective for future work.

2 Intrinsic, extrinsic, and known features

Let $\{\mathbf{X}, \mathbf{y}\}$ denote a data set where $\mathbf{X} = [x_{ij}]_{1 \leq i \leq D, 1 \leq j \leq N}$ contains the features and $\mathbf{y} = [y_j]_{1 \leq j \leq N}$ gives the corresponding labels. Labels y_j take their value in $\{Y_1, Y_2, \ldots, Y_C\}$, where C is the total number of classes.

In the training phase, the rows of data set $\mathbf{X}$ can be splitted into intrinsic and extrinsic features. As a reminder, intrinsic features are known at all times, independently of any classification, whereas extrinsic features require at least one class to be identified in the data set. Let $\mathcal{F}_{\text{in}}$, $\mathcal{F}_{\text{ex}}$ denote the non-intersecting sets of indices corresponding to intrinsic and extrinsic features. As extrinsic features represent relationships between objects of different classes, we assume that N is a multiple of C and that the data set consists of N/C groups of objects where all classes are instantiated once. Within the framework of image segmentation, this means that each image contains a single object of all kinds. This aasumption avoids undetermined or ambiguous relationships.

In the test phase, the unlabeled data set $\mathbf{X}'$ contains missing values for all extrinsic features. During the incremental classification process, blanks are filled in as soon as class labels are attributed. Let $\mathcal{F}_{\text{kn}}^{(t)}$ denote the set of indices corresponding to known features at iteration t of the incremental procedure. We have $\mathcal{F}_{\text{in}} = \mathcal{F}_{\text{kn}}^{(1)} \subseteq \mathcal{F}_{\text{kn}}^{(t)} \subseteq \mathcal{F}_{\text{kn}}^{(t+1)}$.

As the evaluation of yet unknown extrinsic features requires some precise class label to be attributed with reasonable certainty, it is more natural to use binary classifiers. Therefore, we must determine a running order for the binary classifiers. For this purpose, the already known features must be ranked according to their usefulness for binary classification. We suggest a ranking that assesses the overlapping of classes for a given feature. Let $\mathcal{N}_j^K(\mathbf{X})$ denote the set of indices corresponding to the K nearest neighbors of the jth column $\mathbf{x}_{\bullet j}$ of data set $\mathbf{X}$. Let $\mathcal{C}_k(\mathbf{y}) = \{p \text{ s.t. } y_p = Y_k\}$ be the set of indices associated with class Y_k. The usefulness of a certain feature to classify data inside or outside

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

class Y_k can be measured as

$$s_{ik} = \frac{1}{K|\mathcal{C}_k(\mathbf{y})|} \sum_{p \in \mathcal{C}_k(\mathbf{y})} |\{q \text{ s.t. } q \in \mathcal{N}_p^K(\mathbf{e}_i^T\mathbf{X}) \text{ and } y_q = Y_k\}| \ ,$$

where $|A|$ denotes the cardinality of set A and $\mathbf{e}_i$ is a vector of zeros everywhere except the ith element equal to 1. The value of s_{ik} can range from 0 to 1. The latter value indicates that class Y_k does not overlap with other classes along the axis of the ith feature. Each row of matrix $\mathbf{S} = [s_{ik}]$ can be computed quite efficiently by sorting vector $\mathbf{e}_i^T\mathbf{X}$ and sliding a $(2K+1)$-wide window. This leads to a computational complexity of $\mathcal{O}(N\ln(N) + NK\ln(K))$ for each feature.

3 Incremental feature computation and classification

The incremental procedure that we propose works as follows. First, we store the centered and normalized training set. Second, we compute matrix $\mathbf{S}$ and initialize $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{in}}$. Next, we start the incremental iterations. At each iteration we go through the following steps:

1. Compute $\ell = \max_k s_{ik}$ with $i \in \mathcal{F}_{\text{kn}}$; set $s_{i\ell} = 0$.
2. Train the ℓth binary classifier on the reduced data set $[x_{ij}]_{i \in \mathcal{F}_{\text{kn}}^{(t)}, 1 \leq j \leq N}$.
3. Attribute the class label Y_ℓ to the object in the test set having the highest probability to belong this class (make a random pick in case of a tie).
4. Compute all extrinsic features that involve a relationship with the object of class Y_ℓ and insert their index into $\mathcal{F}_{\text{kn}}^{(t)}$ to obtain $\mathcal{F}_{\text{kn}}^{(t+1)}$.
5. If $\mathcal{F}_{\text{kn}}^{(t+1)} = \mathcal{F}_{\text{in}} \cup \mathcal{F}_{\text{ex}}$, then stop, otherwise start a new iteration.

As features are ranked with matrix $\mathbf{S}$, the classifiers can be trained beforehand to increase the computational efficiency. At the end of the procedure, all features are known, but the classification might not be optimal. Once all features are known, we suggest the use of a multi-class classifier. This last step gives to the object to be classified their final class label. Moreover, this final global classification slightly improves the results, as shown in the experiments.

4 Experiments and results

As a proof of concept, we illustrate the principle of incremental classification with a simple image segmentation problem. The data set consists of artificial images of small worms or caterpillars. In each image, the caterpillar comprises a bright head and 5 dark, almost equidistant body segments (Fig. 1). The position, orientation, and twist of the caterpillars vary in each image. Beyond the rather easy segmentation of the patches corresponding to the caterpillar's head and body, the goal of the problem is to correctly label the first, second, ... and sixth

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

segments in spite of their identical color. Incremental classification provides a non-imperative way to solve the problem. The features for each segmented patch (head or body) are the gray level (intrinsic) and the distance to all other patches (extrinsic). The gray level allows the head to be identified. Knowing where the head is then allows some distances to be measured, which help to progressively distinguish the body segments. In practice, the data set contains 100 images.

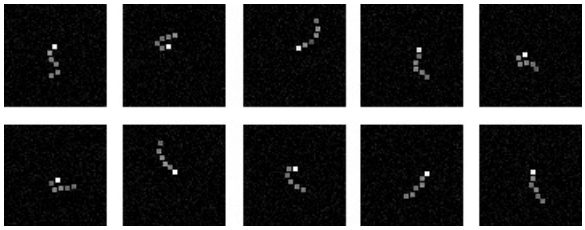

Fig. 1: Some images of the caterpillar problem.

Six patches in each image can be segmented (hence, $N = 600$. Seven features characterize each patch (gray level plus the distance to the head, first body segment, etc.). Only the gray level is an intrinsic feature. Label 1 is associated with the head, label 2 with the first body segment, etc.

We randomly split this data set in two parts: 90 images serves as training set, whereas the remaining 10 images form the test set. The known features in the test set are centered and normalized by subtraction of the mean and division by the standard deviation computed on the training set. The binary and multi-class classifiers rely on the method of the large margin nearest neighbors (LMNN) [6], which combines a usual KNN with metric learning. In our incremental procedure, the Mahalonobis distance of the LMNN is optimized on the training set reduced to the features that are known at each iteration. The extrinsic features are inferred for each image individually. In each image the patch that belongs with the highest probability to the class we wish to identify is used to compute the distance to this class in the considered image. The whole classification procedure is repeated with 20 different training sets for various values of K. The same K is used both in the computation of matrix S and in all LMNNs. Accuracy is defined as the number of data well classified divided by the total number of data.

We analyzed the error rates at the end of the incremental procedure, just before the final multi-class classification, during each iteration and for each class. We also analyzed the order of feature extraction. Table 1 reveals no significant difference in final classification accuracy for the different values of K.

The final multi-class classification improves the accuracy (Table 1 and 2). Indeed, at the end of the incremental classification with binary LMNNs, some data might remain unlabeled and the final classification addresses this issue. Nevertheless, this last iteration cannot correct past classification mistakes.

Table 3 shows that the order of the features induced by $\mathbf{S}$ is stable across the 20 different test sets. For the first 4 iterations, it always extract the features

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

K	3	5	7	9	15
Accuracy	0.959	0.968	0.961	0.961	0.969
Std.	0.054	0.030	0.049	0.036	0.035

Table 1: Accuracy at the end of the classification.

k	3	5	7	9	15
Accuracy	0.952	0.957	0.954	0.953	0.967
Std.	0.054	0.026	0.044	0.037	0.036

Table 2: Accuracy before the last classification.

that involve the knowledge of class 1, 2, 4, and 1. We enabled the possibility to recompute a feature: in the experiment, the extrinsic feature depending on classes 1 and 4 are modified after a first evaluation. Such a class relabeling allows feature computation errors to be corrected in the incremental procedure. The second classification involves more features than the first one and is thus expected to be more reliable.

Class	1	2	3	4	5	6
Iteration 1	20					
2		20				
3				20		
4	20					
5			2			18
6					18	2
7				18	2	
8			18			

Table 3: Number of time a class is selected for binary classification in each iteration ($K = 15$). A blank cell means zero.

Table 4 shows that the third iteration is the less accurate. This iteration identifies class 4 (Table 3). Even with the lower accuracy at iteration 3, the next iterations yield a good accuracy. Table 5 reports that accuracy of the binary

Iteration	1	2	3	4	5	6	7	8
Accuracy	1.000	0.994	0.960	1.000	0.988	0.982	0.976	0.994
Std.	0.000	0.011	0.023	0.000	0.010	0.021	0.028	0.012

Table 4: Binary classifier accuracy at each iteration ($K = 15$).

classifiers at the class corresponding to the first body segment next to the head is has the best accuracy. The class 3 is well classified despite the fact that its classification occurred at the end of the feature extraction process. Figure 2 shows that the images that lead to big classification mistakes are those depicting

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Class	1	2	3	4	5	6
Accuracy	1	0.994	0.994	0.965	0.984	0.986
Std.	0	0.011	0.011	0.020	0.021	0.011

Table 5: Accuracy of the binary classifiers associated with each class ($K = 15$).

highly twisted or curled with large.

Fig. 2: Caterpillars for which we have classification issues.

5 Conclusion

This paper describes a procedure for incremental classification. It can deal with problems where the value of some features requires a partial classification to be already known. The process of incremental classification aims at refining the partial classification in an iterative way. The procedure is generic and can solve the subproblems in each iteration with various classification techniques (e.g. naives Bayes, KNN, SVM, etc.). The final multi-class classifier can be changed as well. Depending on the problem at hand, the procedure must be adapted with appropriate definitions of features and relevance factors. Failure to do so increases the risk of error propagation in the incremental process. Experiments on artificial images show that the procedure is effective.

In the future, we will investigate the possibility to resort to a single classifier that deals with all intrinsic and extrinsic features at all times, thanks to the use of adaptive relevance factors.

References

[1] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.

[2] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[3] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[4] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: minimum spanning forests and the drop of water principle. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1362–1374, 2009.

[5] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[6] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.