Evolutionary Object Detection by Means of Naïve Bayes Models Estimation

Xavier Baró[1] and Jordi Vitrià[1,2]

[1] Computer Vision Center, Edifici O, Campus UAB, Bellaterra, Barcelona
[2] Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007 Barcelona, Spain
{xbaro,jordi}@cvc.uab.cat

Abstract. This paper describes an object detection approach based on the use of Evolutionary Algorithms based on Probability Models (EAPM). First a parametric object detection schema is defined, and formulated as an optimization problem. The new problem is faced using a new EAPM based on Naïve Bayes Models estimation is used to find good features. The result is an evolutionary visual feature selector that is embedded into the Adaboost algorithm in order to build a robust detector. The final system is tested over different object detection problems obtaining very promising results.

1 Introduction

The detection and classification of objects in images that have been acquired in unconstrained environments is a challenging problem because objects can occur under different poses, lighting conditions, backgrounds and clutter. This variation in the object appearance makes unfeasible the design of handcrafted methods for object detection. Although this problem has been the subject of research from the early beginning of the computer vision field, it has not been until the recent past years that researchers have developed generic object recognition systems for a broad class of real world objects. The key point for this achievement has been the use of a machine learning framework that makes use of very large sets of sample images to learn robust models: Given a training set of n pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the ith image and y_i is the category of the object present in $\mathbf{x}_i$, we would like to learn a model, $h(\mathbf{x}_i) = y_i$ that maps images to object categories.

One of the most extended approaches to object detection is based on *local methods*, which model an object as a collection of local visual features or "patches". Thus an image $\mathbf{x}_i$ can be considered to be a vector $(\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,m})$ of m patches. Each patch $\mathbf{x}_{i,j}$ has a feature-vector representation $F(\mathbf{x}_{i,j}) \in \Re^d$; this vector might represent various features of the appearance of a patch, as well as features of its relative location and scale. We can choose from a wide variety of features, such as the fragments-based representation approach of Ullman [1], the gradient orientation-based SIFT [2], or some forms of geometric invariant descriptors. One of the most successfully used *a priori* image feature, at least for a

M. Giacobini et al. (Eds.): EvoWorkshops 2008, LNCS 4974, pp. 235–244, 2008.

broad class of visual objects, is known as Haar feature. These features, which are related to the wavelet decomposition, were originally proposed in the framework of object detection by Viola and Jones [3] in their face detection algorithm.

The two-dimensional Haar decomposition of a square image with n^2 pixels consists of n^2 wavelet coefficients, each of which corresponds to a distinct Haar wavelet. The first such wavelet is the mean pixel intensity value of the whole image; the rest of the wavelets are computed as the difference in mean intensity values of horizontally, vertically, or diagonally adjacent squares.

Haar features present an interesting property in the context of object recognition: they can be computed very fast using the *integral image*. Integral image at location of x, y contains the sum of the pixel values above and left of x, y inclusive.

Viola and Jones [3] used the Adaboost algorithm [4] to learn a real time face detector with a very high classification performance. In their proposal, weak classifiers are threshold-based classification rules on the values of several Haar features of the image. In order to select the best features, they performed an exhaustive search on the whole set of Haar features for a window. Finally, they built a cascade of strong classifiers to achieve a very low level of false positives.

It is important to note that Haar features constitute an over-complete dictionary of the image and that there are more than 2^{18} different features for a small image window of 576 pixels (24x24 pixels). This fact imposes a high computational cost on the learning step of the Adaboost algorithm, which involves several rounds of exhaustive searches. From a practical point of view, the development of a high performance object detector represents, when using conventional hardware, a learning time of the order of several hundred hours.

The work of Viola and Jones was extended by Lienhart and Maydt [5], who showed that the use of a larger feature set may improve the convergence and the performance of the final classifier. The extension of the feature set was done by adding rotated versions of original Haar-like features, and thus adding a factor to the exponential relation between the feature set size and the training time.

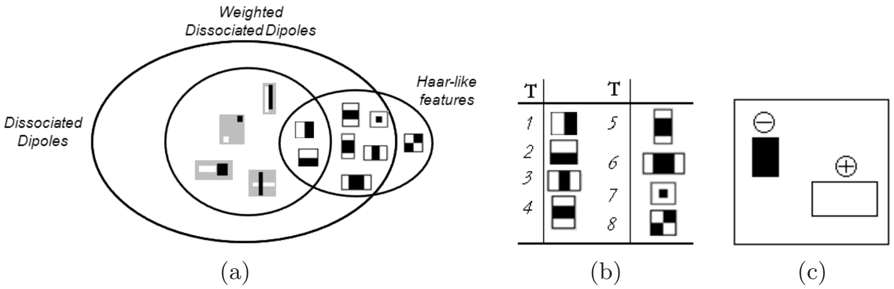
Another natural extension of the Haar features is the dissociated dipoles proposed by Balas and Sinha in [6] in the context of computational neuroscience. As in the case of Haar-like features, they are region based comparisons, but in this case we always have only two regions, which do not have to be adjacent regions. The main reason to consider these features is because while Haar-like features are local descriptors, the dissociated dipoles maintain this ability but also includes non-local descriptors of the images. In addition, they demonstrated that human visual system can perform that type of non-local comparisons, and that these comparisons benefits recognition systems. The proposal of Balas and Sinha, in spite of it can be easily adapted to Adaboost-based detectors, has not been used in any real world object detector due to computational limitations: there are more than 2^{28} different dissociated dipoles in standard image window of 24×24 pixels, what makes the use of the Adaboost algorithm unfeasible.

Using a parameterizable feature set as Haar-like features or dissociated dipoles, the object detection problem can be reformulated as an optimization problem, where we must find the best parameters to minimize an error function (i.e. the weighted classification error in the Adaboost algorithm or the number of miss-classified samples). The new problem corresponds to search in a large and sparse solution space in order to deal with the best parameters. Nowadays, an emerging field is growing up to deal with those type of problems, the Evolutionary Algorithms Based on Probabilistic Models (EAPM) [7]. EAPMs are a new paradigm in the evolutionary computing field. This paradigm was started with the publication of a simple algorithm named *Population Based Incremental Learning* (PBIL) [8], based on a simple univariate model, where all the variables are assumed to be independent. The best individuals of each generation are used to update these variables, and finally the model is sampled to obtain a new generation. In spite of its simplicity, these algorithms demonstrated to converge to good solutions for several problems. Few years later, Schmidth et al. [9] re-introduced the genetic operators to the PBIL algorithm improving significantly its performance. In spite of this return to the origins, the PBIL algorithm introduced an interesting view on evolutionary computation: the extraction of a statistical description of the promising solutions, in terms of a probability distribution is the base of EAPMs and the new systematic way to solve hard search and optimization problems that they represent.

In the literature we can find a wide variety of EAPMs, in which the most important difference is the used probability model. Taking into account the considered interactions between variables, we can classify the models within three main types: Univariate models where no interactions are considered, bivariate models with only pair-wise interactions and finally the models that allow multiple interactions. Once the most convenient probability model is selected, different estimation and sample strategies can be used, thus, we can find different algorithms that share the same type of model. The most known and used algorithms are the UMDA [10], PBIL [8] and cGA [11] for univariate models, MIMIC [12], COMIT [13] and BMDA [14] in the case of bivariate models and finally, considering models with multiple interactions the FDA [15], BOA [16] and EBNA [17]. The use of a complex model allows to better represent the features space, but it adds complexity to the estimation and sampling stages. In [7] a comparison between different EAPMs over several optimization problems suggests that for simple functions, where there are no interaction between the variables, the performance of the univariate and bivariate models perform as well as more complex models, but when we face more complex problems, a more sophisticated probability model is required. As a general rule, more complex models are more reliable but at the expense of bigger execution times.

2 Object Detection Formulation

To formulate the problem, in this section we first define the object detection strategy and the used features. At the end, a parametric definition which can be used in optimization approaches is obtained. The object detection used in this paper is based on the Adaboost algorithm and Haar-like features of Viola and Jones in [3], but in our case we do not use combinations of features and threshold



(a) (b) (c)

Fig. 1. *(a)* Graphical comparison between the different feature sets. *(b)* Values for the parameter T for the Haar-like features. *(c)* Excitatory and inhibitory poles.

to obtain the simple classifiers, only the sign is used instead. This type of features are called ordinal features, which have demonstrated to be robust against various intra-class variations and illumination changes [18]. In addition, we experiment with two larger feature sets than Haar-like features, which cannot be used with the classical Adaboost approach, but can be used with the Adaboost based on EAPMs. These features set are the *Dissociated Dipoles* introduced by Balas and Sinha in [6] and the *Weighted Dissociated Dipoles*, an extension of the dissociated dipoles that allows to represent most part of the Haar-like features (see Fig.1a).

The main idea of Adaboost is to maintain a weight distribution that reflects how many times each sample has been miss-classified by the previous added weak learners, and therefore, allowing to add new simple classifiers that concentrate on those samples which are systematically miss classified. At the end, the final classifier is a weighted combination of several simple classifiers. In Adaboost jargon, the simple classifier is named *Weak Classifier* and the final classifier is named *Strong Classifier*. The process of learning the best weak classifier using the weights distribution is named *Weak Learner* and is discussed in this section.

Given a training set $\langle(\mathbf{x}_1, y_1), ..., (\mathbf{x}_M, y_M)\rangle$, where $y_i \in \{-1, +1\}$ is the target value for sample $\mathbf{x}_i$, the goal of an object detection learning algorithm is to deal with the strong classifier $H(\mathbf{x}_i) = y_i$. In the boosting framework, we define a distribution $W = \{w_1, ..., w_M\}$ over the training set, where each w_i is the weight associated to the sample $\mathbf{x}_i$, and $H(\mathbf{x})$ corresponds to an additive model $H(\mathbf{x}) = \sum_t \alpha_t h_t(\mathbf{x})$ where the final decision is a combination of the decisions of several *weak classifiers* $h(\mathbf{x}) \in \{-1, +1\}$. In contrast to the strong classifier $H(\mathbf{x})$ where we expect a good prediction for any sample $\mathbf{x}_i$ in the training set, in the case of weak classifier we only expect they are better than a random decision.

Given $\mathcal{H}$ the set of all possible weak classifiers, $h^{\mathbf{s}} \in \mathcal{H}$ a certain weak classifier defined by parameters $\mathbf{s}$, W the weights distribution of the Adaboost and $\mathcal{E}(h^{\mathbf{s}}) = Pr_{i \sim W}[h^{\mathbf{s}}(\mathbf{x}_i) \neq y_i]$ the error function, the regression step consists on finding $\mathbf{s}^*$ that $\mathcal{E}(h^{\mathbf{s}^*}) \leq \mathcal{E}(h^{\mathbf{s}}) | \forall h^{\mathbf{s}^*}, h^{\mathbf{s}} \in \mathcal{H}$, where the complexity of finding $\mathbf{s}^*$ depends on the size of $\mathcal{H}$.

Using ordinal measures, Haar-like features can be parameterized by the upper-left position of one of the regions (X, Y) and their size (W, H), because the size and position of all the regions in Haar-like features is predefined for each type T (see Fig. 1b). Therefore, a Weak Classifier with ordinal Haar-like [5] features can be defined as:

$$\begin{aligned} &h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}. \\ \text{where} \quad &\mathbf{s} = (X, Y, W, H, T). \end{aligned} \tag{1}$$

To evaluate this feature, the mean intensity value of negative region (black one) is subtracted from mean value of positive region. The sign of this subtraction is used as the final class value. In the case of dissociated dipoles we can analogously parameterize the two regions as:

$$\begin{aligned} &h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}. \\ \text{where} \quad &\mathbf{s} = (X_e, Y_e, W_e, H_e, X_i, Y_i, W_i, H_i). \end{aligned} \tag{2}$$

where the subscript e refers to the excitatory pole (white in Fig. 1c) and the subscript i to the inhibitory pole (black).Finally, to deal with the weighted dissociated dipoles, we just add a weight parameter $RW \in \{1, 2\}$ to each pole:

$$\begin{aligned} &h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}. \\ \text{where} \quad &\mathbf{s} = (X_e, Y_e, W_e, H_e, RW_e, X_i, Y_i, W_i, H_i, RW_i). \end{aligned} \tag{3}$$

In addition to the mandatory parameters defined above, we can add extra parameters to improve the results. For instance, we add a polarity parameter which inverts the classification value. This parameter allows a fast step on the search process, inverting the regions. Other parameters can be added to extend the features to multichannel images, but the use of color images is out of the focus of this work.

Once the differences between the feature sets are described, we summarize the problem as finding the best instance of a set of random variables $\mathbf{s} = \{X_1, ..., X_K\}$, to minimize $\mathcal{E}(h^{\mathbf{s}}) = Pr_{i \sim W}[h^{\mathbf{s}}(\mathbf{x}_i) \neq y_i]$, where the only difference between each feature set is how to evaluate $h^{\mathbf{s}}$ and the dimension of $\mathbf{s}$.

If we analyze the random variables, it is easy to discover that there are multiple dependencies between them, for instance, regions located near the right or bottom sides of the training window cannot have large sizes. The rest of the paper is concentrated on defining an evolutionary approach based on a novel EAPM which allows to solve this optimization problem taking into account these dependencies. From this point, we can forget about the fact we are in an object detection problem and the use of Adaboost. The problem now is how to find the best parameters of a feature given a weight distribution over the samples.

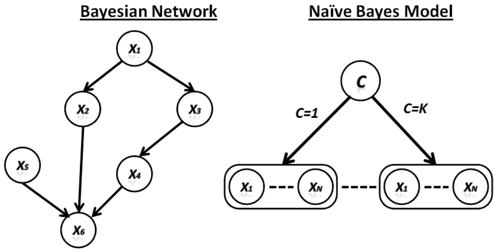


Fig. 2. Probabilistic models

3 Evolutionary Algorithm Based on Naïve Bayes Estimation

As we show before, the formulation derived from the object detection problem has multiple dependencies between the variables, and thus, we need to use a model that allows to represent all these interactions. In addition, as the optimization step must be repeated several times in order to obtain a good detector, we need a fast method. In this section we present a novel EAPM algorithm which accomplish both requirements. First we introduce the probability model, and lastly the evolutionary algorithm.

The most extended probabilistic model to capture multiple dependencies is the Bayesian network (or a belief network), represented as a probabilistic graphical model, specifically an acyclic directed graph, where each node corresponds to a variable (measured parameter, latent variable or hypothesis) and whose arcs encode the dependence between variables. Learning a Bayesian network from data is a two-fold problem: Structure learning and parameter estimation. Although there exist good methods to estimate the structure and parameters of a Bayesian network, because exact inference is $\#P$-complete and thus the existent methods are often too costly, approximate methods like Markov Chain Monte Carlo [19] and loopy belief propagation [20] must be used. The applicability of the Bayesian networks is limited by the fact that these methods have an unpredictable inference time and its convergence is difficult to diagnose.

An alternative to the Bayesian networks are the Naïve Bayes models, where with the "naive" assumption that all variables are mutually independent given a "special" variable C the model is simplified. Although the resulting model is very simple (see Fig. 2), in [21] Lowd and Domingos demonstrated from an empirical point of view that this simpler model has the same representation power than a Bayesian network. In addition to several experiments comparing both models, in this work they also propose the Naïve Bayes Estimation (NBE) algorithm to efficiently estimate Naïve Bayes Models from data. This algorithm consists of an Expectation Maximization (EM) method wrapped in an outer loop that progressively adds and prunes mixture components. Once the probability model is defined, and we have the NBE algorithm to estimate this model from a given data, the final algorithm is presented in Alg. 1.

Algorithm 1. EANBE learning algorithm

```
Input: Initial population $P_0$, evaluation function $\mathcal{E} : \mathbb{R}^D \mapsto \mathbb{R}$ (where $D$ is the dimension of each individual $I \in P_0$ and $L$ the number of individuals in $P_0$), the selection percentage $P_S$ and the hold-out percentage $P_H$.
Output: The best individual $I_{best}$ among all evaluated individuals.
  $L_S \Leftarrow round(P_S \times L)$
  $L_H \Leftarrow round(P_H \times L_S)$
  repeat
    Evaluate the population $P_i$ using the evaluation function $\mathcal{E}$
    Select the best individual $I_i$ in the population $P_i$
    Select the $L_S$ best individuals using the evaluation value
    if $\mathcal{E}(I_i)$ is better than $\mathcal{E}(I_{Best})$ or it is the first iteration then
      $I_{Best} \Leftarrow I_i$
    end if
    Divide the selected individuals into two random subsets $T$ and $H$ with $L_S - L_H$ and $L_H$ individuals respectively.
    Use the NBE algorithm to learn a model $M$ using $T$ as the training set and $H$ as the hold-out set.
    Sample the model $M$ to obtain a new population $P_{i+1}$ with $L$ individuals.
  until The maximum number of generations or some other stopping criteria are achieved
  return $I_{Best}$ and $\mathcal{E}(I_{Best})$.
```

4 Results

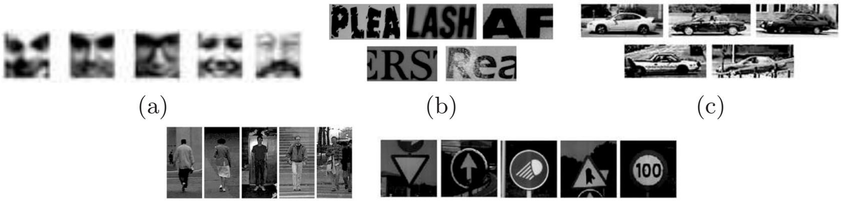
To validate the proposed object detection method, we have developed the *eapmlib*, a C++ based library that implements a general framework to work with EAPMs. The learning process is done by a Gentle Adaboost version in Matlab, using MEX files to communicate with the *eapmlib*. All the source codes for the library, MEX files and Matlab files are available in an online appendix at *www.cvc.uab.cat/~xbaro/eanbe/*.

In the first part of this section we describe the data sets used to perform the experiments, and after we describe the experiments. Finally, we analyze the obtained results.

4.1 Data Sets

In order to verify the usefulness of the presented methodology we have selected different real world object detection problems, using for each one a state of the art public database to facilitate the reproduction and comparison of the results. In the databases where background images are not provided, we create them using images from the *Corel Photo Libraries*. Data sets are described in the following:

Faces: We use the MIT-CBCL face database with a random selection of 1.000 face images and 3.000 non-face images. All the images correspond to frontal faces with several illumination changes.



(a) (b) (c)

(d) (e)

Fig. 3. Data set examples. *(a)* Faces *(b)* Text *(c)* Cars *(d)* Pedestrians *(e)* Traffic signs.

Text: We use the text location dataset from the *7th International Conference on Document Analysis and Recognition (ICDAR03)*. The images correspond to text regions over a wide set of surfaces and with several illumination changes. To obtain the training data, each text region has been split into overlapped subregions with the same size.

Cars: We use the UIUC cars database, with a total of 1.050 images containing 550 instances of lateral views of different cars in urban scenes and 500 images of background.

Pedestrians: We use the INRIA Person Dataset, with 2.924 images divided into 924 pedestrian instances and 2.000 background images.

Traffic Signs: In this case we use real images acquired in the context of a mobile mapping project provided by the Institut Cartogràfic de Catalunya[1]. The database consists on 1.000 images containing a traffic sign and 3.000 background images [22].

4.2 Measures

To compare the performance of each configuration of the system, the classification Balanced Error (BER) is used. This measure takes into account the different number of examples in each of the classes in order to face unbalanced problems. Given a samples set $\{(\mathbf{x}_i, y_i)|i = 1, ..., M\}$ with $y_i \in \{-1, 1\}$ and a certain hypothesis $h(\mathbf{x}) \mapsto \{-1, 1\}$ the balanced error is defined as:

$$BER = \frac{\frac{FN}{N^+} + \frac{FP}{N^-}}{2}. \tag{4}$$

where N^+ and N^- are the number of positive and negative samples respectively, and FP and FN the number of false positive and false negative samples.

4.3 Experiments

To validate the proposed schema and compare each selection of feature set and evolutionary algorithm, we perform a stratified 10-fold cross-validation. The population size for genetic algorithm and EANBE is set to 100 individuals, using $T = 200$ as the number of iterations in the Gentle Adaboost algorithm. In the

[1] Institut Cartogràfic de Catalunya (`www.icc.es`).

Table 1. Results obtained in the experiments

Data Set	**Dissociated Dipoles**		**Haar-like features**		**Weighted Diss. Dip.**	
	GA	*EANBE*	*GA*	*EANBE*	*GA*	*EANBE*
Cars	**79.84%**	79.67%	70.71%	**71.47%**	**95.95%**	95.82%
Faces	70.98%	**73.53%**	56.50%	**57.23%**	88.65%	**88.97%**
Text	55.42%	**58.98%**	48.17%	**49.82%**	**89.57%**	89.52%
Pedestrians	71.04%	**75.00%**	55.95%	**57.34%**	89.05%	**90.41%**
Traffic Signs	66.37%	**70.00%**	64.50%	**66.33%**	89.05%	**90.28%**

case of genetic algorithm, the individuals are encoded using binary gray code, while in the case of EANBE, the decimal code is used directly. The BER mean vales obtained in the experiments are shown in table 1.

Note that the results obtained using the Weighted Dissociated Dipoles outperform always the ones obtained with the smaller features sets. The results obtained by the other two feature sets, are not significantly different, but in the majority of the cases the dissociated dipoles outperform the Haar-like. Following the trend highlighted by Lienhart and Mayd in [5], the results improve with the size of the feature set.

5 Conclusions and Future Work

Although the tests must be extended to a larger set of databases, the results obtained with weighted dissociated dipoles in combination to the new evolutionary algorithm encourage to continue with this line. As a future work we want to extend this approach to multi-class learning and use a priori knowledge from the learning data in order to speed-up the selection process and obtain not only good features but meaningful features.

Acknowledgments. This work has been developed in a project in collaboration with the *Institut Cartogràfic de Catalunya* under the supervision of Maria Pla. This work was partially supported by MEC grant TIC2006-15308-C02-01 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

1. Ullman, S., Sali, E.: Object classification using a fragment-based representation. Biologically Motivated Computer Vision, 73–87 (2000)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of ICCV, Corfu., pp. 1150–1157 (1999)
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of the CVPR 2001, vol. 1, pp. 511–518 (2001)
4. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)

5. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. of the Int. Conf. on Image Processing, pp. 900–903 (2002)
6. Balas, B., Sinha, P.: Dissociated dipoles: Image representation via non-local comparisons. In: Annual meeting of the Vision Sciences Society, Sarasota, FL (2003)
7. Blanco, R., Lozano, J.: An Empirical Comparison of Discrete Estimation of Distribution Algorithms, pp. 167–180. Kluwer Academic Publishers, Dordrecht (2001)
8. Baluja, S., Caruana, R.: Removing the genetics from the standard genetic algorithm. In: The Int. Conf. on Machine Learning 1995, pp. 38–46 (1995)
9. Schmidt, M., Kristensen, K., Jensen, T.R.: Adding genetics to the standard PBIL algorithm. In: Proceedings of the 1999 Congress on Evolutionary Computation, vol. 2, pp. 1527–1534 (1999)
10. Muhlenbein, H.: The equation for response to selection and its use for prediction. Evolutionary Computation 5(3), 303–346 (1997)
11. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. IEEE-EC 3(4), 287 (1999)
12. de Bonet, J.S., Isbell Jr, C.L., Viola, P.: MIMIC: Finding optima by estimating probability densities. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.) Advances in Neural Information Processing Systems, vol. 9, p. 424. MIT Press, Cambridge (1997)
13. Baluja, S., Davies, S.: Using optimal dependency-trees for combinational optimization. In: Proceedings of the Fourteenth Int. Conf. on Machine Learning, pp. 30–38. Morgan Kaufmann, San Francisco (1997)
14. Pelikan, M., Mühlenbein, H.: The bivariate marginal distribution algorithm. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) Advances in Soft Computing - Engineering Design and Manufacturing, pp. 521–535. Springer, London (1999)
15. Mühlenbein, H., Mahning, T.: The factorized distribution algorithm for additively decomposed functions. In: Second Symposium on Articial Intelligence. Adaptive Systems, CIMAF 1999, La Habana, pp. 301–313 (1999)
16. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: BOA: The Bayesian optimization algorithm. In: Banzhaf, W., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference GECCO 1999, Orlando, FL, 13-17 1999, vol. I, pp. 525–532. Morgan Kaufmann, San Francisco (1999)
17. Etxeberria, R., Larrañaga, P.: Global optimization with bayesian networks. In: Second Symposium on Artifcial Intelligence(CIMAF 1999), Cuba, pp. 332–339 (1999)
18. Thoresz, K., Sinha, P.: Qualitative representations for recognition. Journal of Vision 1(3), 298–298 (2001)
19. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: Markov Chain Monto Carlo in Practice. Interdisciplinary Statistics Series. CRC Press, Boca Raton (1996)
20. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Generalized belief propagation. In: NIPS, pp. 689–695.
21. Lowd, D., Domingos, P.: Naive bayes models for probability estimation. In: ICML 2005: Proceedings of the 22nd international conference on Machine learning, pp. 529–536. ACM Press, New York (2005)
22. Baró, X., Vitrià, J.: Traffic sign detection on greyscale images. In: Recent Advances in Artificial Intelligence Research and Development, IOS Press, Amsterdam (2004)