Online Learning of Discriminative Patterns from Unlimited Sequences of Candidates

Ilkka Autio
Department of Computer Science
University of Helsinki, Finland
iautio@cs.helsinki.fi

J.T. Lindgren
Department of Computer Science
University of Helsinki, Finland
jtlindgr@cs.helsinki.fi

September 1, 2006

Abstract

Recent research in object recognition has demonstrated the advantages of representing objects and scenes through localized patterns such as small image templates. In this paper we study the selection of patterns in the framework of extended supervised online learning, where not only new examples but also new candidate patterns become available over time. We propose an algorithm that maintains a pool of discriminative patterns and improves the quality of the pool in a disciplined manner over time. The proposed algorithm is not tied to any specific pattern type or data domain. We evaluate the method on several object detection tasks.

1 Introduction

One interesting approach to object detection is to represent scenes as collections of small image parts or other local models, which are then matched to stored templates (e.g. [10, 13, 4, 2, 12, 5, 11]). These *local patterns* in an image may correspond to intuitively understandable object parts (see figure 2), and make the resulting overall *hierarchical pattern* easily interpretable (i.e. certain spatial layouts of eye, nose and cheekbone templates indicate a face).

Previously these patterns have been selected from the data with batch learning [13, 2, 12, 11]. However, for many areas of interest (robotics, face recognition, etc) we will eventually need approaches that learn in more dynamic manner, allowing the system to adjust to fresh data.

Towards this goal, we present an online algorithm for maintaining a pool of discriminative patterns that are automatically selected from the data. The proposed algorithm is not tied to any particular way of representing or matching the patterns, and

basically any numerical feature could be used as a candidate pattern after thresholding. Due to the incremental nature of the algorithm, new training examples, patterns, representations and even ways to match them can be incorporated at any time.

In the online setting, candidate patterns cannot be known beforehand, and it eventually becomes infeasible to memorize all the previously seen inputs. Hence, the joint occurrence statistics of the patterns are unavailable and cannot be used for selection. Also, the statistics of different patterns are not directly comparable since the amounts of evidence are not the same. This so-called missing values problem cannot be addressed by the corresponding batch algorithms without explicitly quantifying the uncertainty of the point estimates. Due to these issues, we chose to develop the proposed online selection algorithm from scratch.

As far as we know, the selection problem in this form cannot be straightforwardly handled by methods presented in e.g. sequential analysis (for a review, see [9]), expert selection (e.g. [7]), or traditional feature selection (for a recent review, see [6]). Our setting resembles the infinite attribute model [3], with the main difference that we work under resource constraints.

2 The unlimited candidates setting

Let $\mathbf{x}_j \in \mathbb{R}^d$ denote a *vectorized input image* with associated class label $y_j \in \{0, 1\}$ and observation time t_j. We assume an unlimited sequence of patterns or functions $f_i : \mathbb{R}^d \to \{0, 1\}$. Patterns are revealed in time, just like input images. Because patterns are not necessarily forgotten like images, we denote the lifetime of a pattern by $[T_i, T_i']$. A pattern is *recognizable in* $\mathbf{x}_j$ if $t_j \in [T_i, T_i']$. A pattern is *present* (*absent*) in $\mathbf{x}_j$ iff $f_i(\mathbf{x}_j) = 1$ (0) and the pattern is recognizable in $\mathbf{x}_j$. When a pattern is recognizable, we have the program code for detecting if the pattern is present. For example, a pattern f_i may be a matching procedure for a local image patch extracted from $\mathbf{x}_j$ ($T_i = t_j$). If $t_j \notin [T_i, T_i']$, the pattern is *unrecognizable in* x_j and detection code is unavailable. The primary task is to predict the class label y_j given the patterns recognizable in $\mathbf{x}_j$. The class may be considered a *hierarchical* pattern $F(\mathbf{x}_j) = y_j$ such that the parameters of the detection code must be learned. The secondary task is to create and maintain a pool of recognizable patterns useful in the primary task.

Let A denote the (finite) set of patterns f_i recognizable prior to some finite but arbitrarily large horizon t_j ($T_i \leq t_j$). In this paper, we assume that the patterns in A are mutually independent given the class. It follows that the Naive Bayes (NB) classifier is optimal for the classification task. Denoting $P(f_i = 1|y = c) = p_{ic}$ and $P(f_i = 0|y = c) = n_{ic}$, the logarithmic NB classifier for $f_i \in A$ is

$$g_A(\mathbf{x}) = \sum_{f_i(\mathbf{x})=1} \log \frac{p_{i1}}{p_{i0}} - \sum_{f_i(\mathbf{x})=0} \log \frac{n_{i0}}{n_{i1}}. \quad (1)$$

The classifier thresholds $g_A(\mathbf{x})$ against the prior (here 0).

When the NB assumptions hold, discarding patterns can only result in loss of accuracy. Since using the set A is infeasible in practice and might result in overfitting, we need to select a subset of the available patterns. Hence, we define the optimal set

$B(t_j) \subset A$ of size K at time t_j as

$$B(t_j)^* = \underset{B(t_j)\subset A, |B(t_j)|=K}{\arg\min} E[|g_{B(t_j)}(\mathbf{x}_j) - g_A(\mathbf{x}_j)|], \tag{2}$$

where $f_i \in B(t_j) \Rightarrow T_i < t_j$.

According to criterion (2), optimality implies that the expected difference between the posterior probabilities produced by the two models is minimized. The optimal solution cannot include patterns available in the future, but we assume it may include any patterns in the past (forgetting the wrong patterns is suboptimal).

3 Deriving online approximations

Online optimization of (2) is non-trivial. We do not have access to true probabilities and point estimates are not comparable due to the f_i becoming available at different times.

Denote $B(t_j)$ by B, let $B \subset A$ and $\overline{B} = A \setminus B$. Denote by $D \subseteq A$ the patterns that would be present in $\mathbf{x}_j$ if A was recognizable. From (2) algebraic manipulation yields

$$\begin{aligned} E[|g_B - g_A|] &= E[|\sum_{i\in\overline{B}\cap D} \log\frac{p_{i1}}{p_{i0}} - \sum_{i\in\overline{B}\setminus D} \log\frac{n_{i0}}{n_{i1}}|] & (3)\\ &\leq \sum_{i\in\overline{B}} \max\{\log\frac{p_{i1}}{p_{i0}}, \log\frac{n_{i0}}{n_{i1}}\}. & (4)\end{aligned}$$

Now, instead of solving (2), we may attempt to minimize the upper bound (4) to get an approximate solution that is equivalent to

$$\begin{aligned} B^* &= \underset{B\subset A, |B|=K}{\arg\max} \sum_{i\in B} \max\{\log\frac{p_{i1}}{p_{i0}}, \log\frac{n_{i0}}{n_{i1}}\} & (5)\\ &= \underset{B\subset A, |B|=K}{\arg\max} \sum_{i\in B} m_i & (6)\\ &= \underset{B\subset A, |B|=K}{\arg\max} S(B), & (7)\end{aligned}$$

where m_i is defined as the *merit* of f_i and $S(B)$ as the *score* of set B.

We propose an algorithm based on Bayesian inference to create and maintain B in a disciplined manner. Suppose that at the instant t the algorithm maintains a pool $B(t)$ of recognizable patterns and a queue $Q(t) \subset \overline{B(t)}$ of recently observed (recognizable) patterns (applicants). According to (7), the algorithm can try to improve $S(B)$, i.e., $S(B(t+1)) > S(B(t))$, by exchanging a pattern f_j in $B(t)$ with a pattern f_i in $Q(t)$. As we cannot guarantee that a particular exchange increases $S(B)$, we choose the one that has the largest probability of doing so:

$$(i^*, j^*) = \underset{(i,j)\in Q\times B}{\arg\max}\ P(m_i > m_j | Data). \tag{8}$$

Further, we may impose a threshold on $P(m_{i^*} > m_{j^*}|Data)$ such that the algorithm abstains from making an exchange if even the best one is not likely to increase $S(B)$.

Because the merits m_i, m_j in (8) are functions of probabilities (5), we face the problem of estimating the probability of the underlying probabilities being such that $m_i > m_j$ is true. To make the solution more intuitive, we will first express the complex event $m_i > m_j$ in terms of more elementary events. Denote

$$\begin{array}{lcrcl} q_1 & \doteq & p_{i1}p_{j0} & > & p_{j1}p_{i0} \\ q_2 & \doteq & p_{i1}(1-p_{j1}) & > & (1-p_{j0})p_{i0} \\ q_3 & \doteq & (1-p_{i0})p_{j0} & > & p_{j1}(1-p_{i1}) \\ q_4 & \doteq & (1-p_{i0})(1-p_{j1}) & > & (1-p_{j0})(1-p_{i1}). \end{array}$$

Now simple algebraic manipulation yields

$$m_i > m_j \Leftrightarrow (q_1 \wedge q_2) \vee (q_3 \wedge q_4). \tag{9}$$

From above we see that the event $m_i > m_j$ is a logical function of four independent binomial distributions specified by the four independent parameters p_{i1}, p_{i0}, p_{j1} and p_{j0}. In principle we could now calculate the probability of $m_i > m_j$ by integrating the joint density $p(p_{i1}|Data)p(p_{i0}|Data)p(p_{j1}|Data)p(p_{j0}|Data)$ over the region of the parameter space where the logical expression holds. Here we solve the integration problem by simulation. For the simulation, we need just the cumulative distribution functions of the four parameters.

Denote the parameter of interest by p (e.g., p_{i1}). Suppose the status of the pattern (f_i) is known for n images (of the positive class c_1), and the pattern was present in k of the images. Assuming uniform prior density ($Beta(1,1)$), the posterior density of p (given n, k) is

$$P(p|n,k) = \frac{\binom{n}{k}p^k(1-p)^{n-k}}{\int_0^1 \binom{n}{k}p^k(1-p)^{n-k}dp} \tag{10}$$

$$= \frac{p^k(1-p)^{n-k}}{Beta(k+1, n-k+1)}. \tag{11}$$

Given the density, we sample from the Beta-distribution to do numerical integration. Simulating the parameter values, we can estimate the probability of (9) by a frequency count.

Hence, we propose the following algorithm.

4 Faster heuristic for pattern selection

As simulating equation (8) is computationally intensive, we invented a more simple algorithm for comparison. For this algorithm (called β-heuristic), we replace the merit measure defined in (5-6) with

$$m'_i = \sum_{t=T_i}^{T'_i} \alpha^{T'_i - t}(I(f_i(\mathbf{x}_t) = 1|c_1) - I(f_i(\mathbf{x}_t) = 1|c_0)), \tag{12}$$

where I is an indicator function and $\alpha \in (0, 1]$ a forgetting factor. The merit simply counts pattern occurrences in one class minus the other, weighted by age. If the

Algorithm 1 β-learn

Require: pool B, queue Q, constant s
  for $t \in 1, 2, 3, ...$ **do**
    Receive input $\mathbf{x}$ for iteration t
    Output prediction $\hat{y_t} = \begin{cases} 1 & \text{, if } g_B(\mathbf{x}) > 0 \\ 0 & \text{, otherwise} \end{cases}$
    Receive true label y_t of $\mathbf{x}$
    if $\mathbf{x}$ is the first input with $y_t = 1$ **then**
      fill B and Q with new patterns from $\mathbf{x}$
    else if $y_t = 1$ and $\hat{y_t} = 0$ **then**
      sample s new patterns from $\mathbf{x}$ to Q in FIFO fashion
    end if
    for all $f_i \in B \cup Q$ **do**
      update estimates used in eq. (1) and (7)
    end for
    if $y_t \neq \hat{y_t}$ **then**
      use eq. (8) to check if any pattern should be
      exchanged between B and Q.
    end if
  end for

class distribution is balanced, high values indicate both confidence and discriminability. When eq. (12) is evaluated in practice, T_i is the iteration when f_i was added to Q, and T_i' is the current iteration. We included the forgetting factor to discount older evidence, helping patterns from Q to more easily overtake patterns in B. Forgetting can be disabled with $\alpha = 1$.

In addition, the promotion criterion from equation (8) is changed to exchange the best pattern of Q with the worst pattern of B, if

$$\min_{f_i \in B} m_i' < \max_{f_j \in Q} m_j'. \tag{13}$$

These changes affect only a single line of the pseudocode of algorithm (1), where eq. 8 is evaluated.

The β-heuristic algorithm only requires maintaining a few numerical counters and does not require simulation. It is also computationally cheaper: the evaluation of eq. (8) costs $O(|B||Q|)$, whereas eq. (13) costs $O(|B|+|Q|)$. Updating the estimates used in eqs. (1) and (7) costs $O(|B| + |Q|)$ for both algorithms. In practice, the cost of evaluating the pattern presences in $\mathbf{x}$ may dominate.

5 Empirical evaluation

We evaluate our methods on the problems of detecting cars, faces, motorbikes, airplanes and houses. For cars, we use the dataset provided by Agarwal and Roth [1]. The bioidfaces data is from the BioID dataset [8]. The caltechfaces, motorbikes, airplanes and houses belong to the Caltech-datasets[1], as well as the backgrounds data, which was

[1]Downloaded from `http://www.robots.ox.ac.uk/~vgg/data.html`

Table 1: Achieved mean accuracies.

Online accuracy	β-learn	β-heuristic
rothcars	0.91 ± 0.01	0.90 ± 0.02
bioidfaces	0.94 ± 0.01	0.95 ± 0.00
caltechfaces	0.92 ± 0.01	0.92 ± 0.01
airplanes	0.89 ± 0.01	0.91 ± 0.01
motorbikes	0.88 ± 0.01	0.89 ± 0.01
houses	0.80 ± 0.01	0.82 ± 0.01
10-fold accuracy		
rothcars	0.95 ± 0.02	0.92 ± 0.03
bioidfaces	0.96 ± 0.02	0.97 ± 0.02
caltechfaces	0.94 ± 0.03	0.93 ± 0.02
airplanes	0.88 ± 0.03	0.92 ± 0.02
motorbikes	0.90 ± 0.04	0.92 ± 0.03
houses	0.81 ± 0.03	0.84 ± 0.04

used as negative examples on all problems. For each class (including the negative), we sampled 500 images, or the dataset size, if smaller. The images were converted to grayscale, and downscaled if the larger axis exceeded 200 pixels.

The sizes of the buffers B and Q were fixed to 32 and 512, respectively. Constant s was set to 64. β-learn made an exchange if the highest probability in eq. (8) exceeded 0.8. For the β-heuristic algorithm, α was set to 0.98. Here the used patterns were image patches sampled from the input images. The numeric parameters of the patches were sampled from uniform distributions, including selection location, patch size, acceptable matching region, and matching threshold. A pattern was taken as present in an image, if its normalized cross-correlation in the acceptable region exceeded the patch-specific threshold. All the details can be seen from the freely available source code package.

In our framework, a single run of an algorithm performs a single pass over the entire dataset. We define the percentage correct (or online accuracy) up to iteration n as follows,

$$Acc(n) = 1 - \frac{1}{n}\sum_{t=1}^{n} I(\hat{y}_t \neq y_t) = \frac{1}{n}\sum_{t=1}^{n} I(\hat{y}_t = y_t). \tag{14}$$

The accuracies over the whole datasets (i.e. n is the number of examples in the set) are shown in table 1. The results reported are averages of 10 individual runs of each algorithm where the ordering of the instances was randomly permuted on each run. The table also shows the more common 10-fold cross-validation accuracies. For both cases, the standard deviations are shown as well. Figure 1 shows the online accuracy development over time for the caltech faces dataset, averaged over the 10 runs. For other datasets, the plots had a similar form. Some of the selected patterns for the used datasets are shown in figure 2 for clarity.

The results seem acceptable considering the NB assumptions and the pattern type used (i.e. limited invariances and the rigidity of the template representations). We made control experiments to verify that these properties alone do not allow for good results in our setting. This was found to be true: without proper selection mechanism

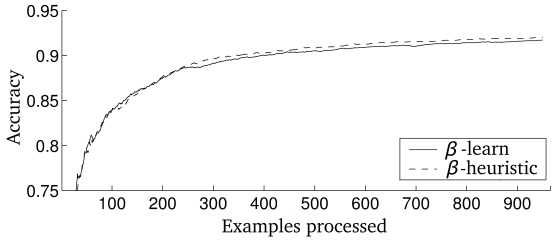

Figure 1: Behavior on the caltech faces.

Figure 2: Some selected patterns.

the accuracies remained close to chance level.

6 Conclusion

We presented an online algorithm for classification that selects patterns from unlimited sequences of candidates, and examined two criteria for selection. We showed preliminary results demonstrating the feasibility of the methods.

One direction of future work consists of providing a theoretical analysis for the proposed methods and the used learning model. This could result in improved learning algorithms for the unlimited candidates setting. For example, the feature independence assumed here cannot be met in practice. In the current work, filling Q conservatively (only on error) was designed towards keeping the selected patterns less dependent. Presently we are investigating how different kinds of patterns could be combined, and if it is feasible to build deeply hierarchical patterns such that classifiers could become input patterns for other classifiers.

To ensure replicable research, the source codes for the algorithms and experiments described in this paper have been made publicly available[2].

Acknowledgments. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

[2]See `http://www.cs.helsinki.fi/group/porr/`

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. of the 7th European Conference on Computer Vision*, volume 4, pages 113–130, 2002.

[2] I. Autio and J. T. Lindgren. Attention-driven parts-based object recognition. In *Proc. 16th European Conference on Artificial Intelligence*, pages 917–921, 2004.

[3] A. Blum. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9:373–386, 1992.

[4] S. Edelman and N. Intrator. Towards structural systematicity in distributed, statically bound visual representations. *Cognitive Science*, 27(1):73–109, 2003.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[7] M. Herbster and M. Warmuth. Tracking the best expert. *Journal of Machine Learning*, 32(2):151–178, 1998.

[8] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the Hausdorff distance. In *Proc. of the 3rd International Conference on Audio and Video based Person Authentication*, pages 90–95, 2001.

[9] T. L. Lai. Sequential analysis: Some classical problems and new challenges. *Statistica Sinica*, 11:303–408, 2001.

[10] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[11] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, volume 2, pages 994–1000, 2005.

[12] A. Torralba, K. P. Murphy, W. T., and Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, volume 2, pages 762–769, 2004.

[13] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002.