Unsupervised Learning of Visually-Observed Events

Jeffrey Mark Siskind
Department of Computer Science
University of Vermont
Burlington VT 05405 USA
802/656–2538
Qobi@EMBA.UVM.EDU
http://www.emba.uvm.edu/~qobi

People can describe what they see. Not only can they describe the objects that they see, they can also describe the events in which those objects participate. So if you were to see a person pick up a pen, you could describe that event by saying *The person picked up the pen.* In doing so you classify the two participant objects as a person and a pen respectively. You also classify the observed event as a picking-up event. Almost all recognition work in machine vision has focussed on object classification. In contrast, the work described in this abstract addresses the problem of event classification.

Siskind & Morris (1996) present a maximum-likelihood method for training an event classifier. This method operates as follows. Samples of a variety of different event types are enacted using coloured blocks in an ordinary desk-top environment and filmed with a video capture system. For each event type, a number of training instances are filmed. Then, tracking data is obtained for the objects that participate in each event enactment, i.e. the coloured blocks and the hand, using one of several different 2D tracking techniques. One way of performing 2D tracking is to use contour-fitting techniques (Kass, Witkin, & Terzopolous 1988). From this position, orientation, shape, and size information, a feature vector is computed that includes both relative and absolute information as well as first and second derivatives of that information. The tracker thus extracts a vector-valued time series from each training movie. A continuous-distribution hidden Markov model is then trained for each event type on the time series extracted from training movies for that event type. Once a set of models have been trained, new movies can be classified into existing classes by applying the tracker to extract a time series of feature vectors for the new movie, computing the likelihood that this time series was generated by each of the previously-trained models, and selecting the model that was most likely to have generated that time series. This technique has successfully classified 34 out of 36 unseen test movies depicting one of the six events *pick up*, *put down*, *push*, *pull*, *drop*, and *throw* from six training movies for each of the six event classes.

The techniques described by Siskind & Morris (1996) constitute supervised learning from labelled training data. The learner is presented with a set of training examples for each class to be learned. Such a technique might model a child learning the meanings of verbs by hearing utterances of those verbs while seeing events described by those verbs. A learner seeing two different events that both occur when the same verb is uttered might include those two event occurrences in the same class. The utterances would constitute the labels associated with the visual training data.

Intuitively, however, one imagines that children don't need linguistic experience to categorize the world into different classes. More likely, children independently form visual classes without benefit of linguistic labels and then subsequently map linguistic labels onto these classes. Such a process would require the learner to form event classes from visual input, in an unsupervised fashion, from unlabelled training data. This abstract explores work-in-progress that attempts to extend the techniques of (Siskind & Morris 1996) to perform unsupervised learning.

The general idea is to combine the EM algorithm (Dempster, Laird, & Rubin 1977) with the Baum-Welch reestimation procedure (Baum *et al.* 1970). Each movie in the training set is assumed to have been generated by a mixture of hidden Markov models. The combined algorithm simultaneously trains the mixture proportions and the parameters of the component hidden Markov models. When the algorithm converges, the desired labelling of the training set can be derived by quantizing the ownership matrix.

The combined algorithm operates as follows. Suppose we wish to classify L training movies into J classes. And suppose that each movie l has T_l frames. And suppose that each class j is generated by a U_j-state Markov process. Let $\mathbf{j}_l$ be a random variable denoting the class that generated movie l. And let $\mathbf{u}_{lt}$ be a random variable denoting the state of the Markov process that generated frame t of movie l. And let $\mathbf{x}_{lt}$ be a random variable denoting the feature vector generated for frame t of movie l. Let x_{lt} denote the observed feature vector computed by applying the tracker

to frame t of movie l. And let x_{ltk} denote the k^{th} entry in this vector. Suppose that $\mathbf{x}_{lt}$ is normally distributed with mean μ_{ju} and covariance Σ_{ju} when $\mathbf{j}_l = j$ and $\mathbf{u}_{lt} = u$. Let $\mathcal{N}(x, \mu, \Sigma)$ denote the normal multivariate probably density function of x with mean μ and covariance Σ. And let b_{ju} denote the probability that $\mathbf{u}_{l1} = u$, given that $\mathbf{j}_l = j$, for any l. Thus b_j denotes the initial state vector for class j. Let $a_{ju_1u_2}$ denote the probability that $\mathbf{u}_{lt} = u_2$, given that $\mathbf{j}_l = j$ and $\mathbf{u}_{lt-1} = u_1$, for any l and any $t > 1$. Thus a_j denotes the state transition matrix for class j.

Let α_{jltu} denote the probability that $\mathbf{x}_{l1} = x_{l1}, \ldots, \mathbf{x}_{lt} = x_{lt}$ and $\mathbf{lu}_t = u$, given that $\mathbf{j}_l = j$. These *forward* probabilities can be calculated with the following recurrence relation:

$$\alpha_{jltu} = \begin{cases} b_{ju}\mathcal{N}(x_{lt}, \mu_{ju}, \Sigma_{ju}) & t = 1 \\ \sum_{u'=1}^{U_j} a_{ju'u}\mathcal{N}(x_{lt}, \mu_{ju}, \Sigma_{ju})\alpha_{jlt-1u'} & \text{otherwise} \end{cases}$$

Let $\beta_{jltu_1u_2}$ denote the probability that $\mathbf{x}_{lt} = x_{lt}, \ldots, \mathbf{x}_{lT_l} = x_{lT_l}$ and $\mathbf{u}_l T = u_2$, given that $\mathbf{j}_l = j$ and $\mathbf{u}_{lt-1} = u_1$. These backward probabilities can be calculated with the following recurrence relation:

$$\beta_{jltu_1u_2} = \begin{cases} 1 & t = T_l + 1 \wedge u_1 = u_2 \\ 0 & t = T_l + 1 \wedge u_1 \neq u_2 \\ \sum_{u'=1}^{U_j} a_{ju_1u'}\mathcal{N}(x_{lt}, \mu_{ju'}, \Sigma_{ju'})\beta_{jlt+1u'u_2} & \text{otherwise} \end{cases}$$

Let γ_{jltu} denote the probability that $\mathbf{u}_{lt} = u$, given that $\mathbf{x}_{l1} = x_{l1}, \ldots, \mathbf{x}_{lT_l} = x_{lT_l}$ and $\mathbf{j}_l = j$. This value can be calculated as follows:

$$\gamma_{jltu} = \frac{\sum_{u_2=1}^{U_j} \alpha_{jltu}\beta_{jltuu_2}}{\sum_{u_1=1}^{U_j}\sum_{u_2=1}^{U_j} \alpha_{jltu_1}\beta_{jltu_1u_2}}$$

Let f_{jl} denote the probability that $\mathbf{x}_{l1} = x_{l1}, \ldots, \mathbf{x}_{lT_l} = x_{lT_l}$, given that $\mathbf{j}_l = j$. This value can be calculated as follows:

$$f_{jl} = \sum_{u=1}^{U_j} \alpha_{jlT_lu}$$

Suppose that the actual observed feature vectors were produced by a linear combination of the feature vectors generated by the various classes. Let π_j denote the coefficients of this linear combination. These constitute *mixing proportions*. Let z_{jl} denote the probability that $\mathbf{j}_l = j$. This *ownership matrix* can be calculated as follows:

$$z_{jl} = \frac{\pi_j f_{jl}}{\sum_{j=1}^{J} \pi_j f_{jl}}$$

Computing α, β, γ, f, and z constitute an E step. The following formulas can be used to reestimate the values $\hat{\pi}$, $\hat{b}$, $\hat{a}$, $\hat{\mu}$, and $\hat{\Sigma}$. These constitute an M step.

$$\hat{\pi}_j = \frac{\sum_{l=1}^{L} z_{jl}}{\sum_{j=1}^{J}\sum_{l=1}^{L} z_{jl}}$$

$$\hat{b}_{ju} = \frac{\sum_{l=1}^{L} z_{jl}\gamma_{jl1u}}{\left(\sum_{l=1}^{L} z_{jl}\right)\left(\sum_{u=1}^{U_j}\sum_{l=1}^{L} \gamma_{jl1u}\right)}$$

$$\hat{a}_{ju_1u_2} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} z_{jl}\gamma_{jltu_1}\gamma_{jlt+1u_2}}{\left(\sum_{l=1}^{L} z_{jl}\right)\left(\sum_{u_2=1}^{U_j}\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_{jltu_1}\gamma_{jlt+1u_2}\right)}$$

$$\hat{\mu}_{juk} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} z_{jl}\gamma_{jltu}x_{ltk}}{\left(\sum_{l=1}^{L} z_{jl}\right)\left(\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_{jltu}\right)}$$

$$\hat{\Sigma}_{juk_1k_2} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} z_{jl}\gamma_{jltu}(x_{ltk_1} - \hat{\mu}_{juk_1})(x_{ltk_2} - \hat{\mu}_{juk_2})}{\left(\sum_{l=1}^{L} z_{jl}\right)\left(\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_{jltu}\right)}$$

Work is underway to determine whether this approach can successfully learn to classify visuall-observed events in an unsupervised fashion.

References

Baum, L. E.; Petrie, T.; Soules, G.; and Weiss, N. 1970. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics* 41(1):164–171.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B* 39:1–38.

Kass, M.; Witkin, A.; and Terzopolous, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision* 321–331.

Siskind, J. M., and Morris, Q. 1996. A maximum-likelihood approach to visual event classification. In *Proceedings of the Fourth European Conference on Computer Vision*. Cambridge, UK: Springer-Verlag.