

Identity inference

Shay Ohayon P. Perona

v1 - February 7, 2011 - SO

v2 - February 7, 2011 - PP

1 Introduction

This document summarizes the probability inference employed in the mouse tracking software.

2 Notation

- $(1, \dots, T)$ – The times of observation, until the current time T .
- M – The number of mice.
- $N = M!$ – The number of states in the HMM.
- S_t – The ‘state’ of the HMM at time t . It is a permutation assigning image ellipses (tracker trajectories at time t) to mouse identities.
- O_t – The ‘observation’ at time t . It is the set of pixels inside the four tracker ellipses.

3 Viterbi

This section would include the derivations for the Viterbi algorithm in our setting.

Viterbi finds the single best state sequence $S = \{S_1, S_2, \dots, S_T\}$ for the given observation sequence $O = \{O_1, O_2, \dots, O_T\}$.

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} P[s_1, s_2, \dots, s_t = i | O_1, O_2, \dots, O_t, \lambda] \quad (1)$$

where λ represents the model, which includes the transition matrix $A(t)$, the priors on the initial states, and the probabilities of observing the observations at any given state.

$\delta_t(i)$ represents the best score along a single path, at time t , which accounts for the first t observations and ends in state S_i .

Viterbi is based on the following induction rule:

$$\delta_t(j) = \max_i \delta_{t-1}(i) a_{ij} P[O_t|S_i] \quad (2)$$

Vitebi's algorithm has four stages: initialization, recursion, termination and backtracking.

3.1 Initialization

$$\begin{aligned} \delta_1(i) &= \pi_i P[O_1|S_i] \\ \psi_1(i) &= 0 \end{aligned} \quad (3)$$

$\psi_t(i)$ is used to keep track of the argument that is being maximized in equation 2.

3.2 Recursion

$$\begin{aligned} \delta_t(j) &= \max_i \delta_{t-1}(i) a_{ij} P[O_t|S_i] \\ \psi_t(i) &= \arg \max_i \delta_{t-1}(i) a_{ij} \end{aligned} \quad (4)$$

3.3 Termination

Finding the highest score at the last time point:

$$s_T^* = \arg \max_i \delta_T(i) \quad (5)$$

3.4 Backtracking

$$s_t^* = \psi_{t+1}(s_{t+1}^*) \quad (6)$$

4 Viterbi State inference

Some definitions.

We have $N = 4$ trackers. Each is assigned a unique identity $\{A, B, C, D\}$. Our story begins with a set of states $S_i, \leq i \leq 24$ (for four mice). Each state represents a permutation, or mapping, from trackers to true identities. That is, state $[1,4,3,2]$ corresponds to tracker 1 is A, tracker 2 is D, tracker 3 is C and tracker 4 is B.

What we would like to do is to compute the probability of being in a state given the observable data. This is put forth in the following equation:

$$P[S_i|D] = \frac{P[D|S_i] P[S_i]}{P[D]} = \frac{P[D|S_i] P[S_i]}{\sum_{j=1}^{24} P[S_j] P[D|S_j]} \quad (7)$$

That is, the probability of being in state i given the observable data D , and the expansion of the expression using bayes rule.

Our prior on the states is uniformly distributed, thus equation 7 becomes:

$$P[S_i|D] = \frac{\frac{1}{24}P[D|S_i]}{\frac{1}{24}\sum_{j=1}^{24}P[D|S_j]} = \frac{P[D|S_i]}{\sum_{j=1}^{24}P[D|S_j]} \quad (8)$$

Under the assumption that the data is independent (each ellipse is independent of the others), we obtain:

$$P[D|S_i] = P[D_1, D_2, D_3, D_4|S_i] = \prod_{j=1}^4 P[D_j|ID = S_i[j]] \quad (9)$$

That is, the probability of observing the data given state S_i is the multiplication of seeing each one of the small image patches D_j under the assumption that it belongs to identity $S_i[j]$.

The way we have modeled this distribution was by constructing a projection from the image space to 1D. The projection first extracts the HOG features from the image and then projects them to 1D using a mapping obtained from linear discriminant analysis which takes into account the class of interest. That is, we find the best mapping that separates $S_i[j]$ from all other identities.

We usually think about $P[D_j|ID = S_i[j]]$ in terms of the histogram of the projected HOG features. We had several ideas how to model this distribution (gaussians, t-distribution, non-parametric). Current implementation uses normal distribution.

5 Garbage collection idea

Lets consider the standard two class problem (say, class A and class not A). According to Bayes we get:

$$P[A|x] = \frac{P[x|A]P[A]}{P[x]} = \frac{P[x|A]P[A]}{P[x|A]P[A] + P[x|\neg A]P[\neg A]} \quad (10)$$

Pietro put forth the following idea. Let us consider a third class J (i.e., junk), and assume we have some way to model its distribution $P[x|J]$. Then, we can add its contribution equally to the existing two classes by computing:

$$P[A|x] = \frac{P[x|A]P[A]}{P[x]} = \frac{P[x|A]P[A] + \frac{1}{2}P[x|J]P[J]}{P[x|A]P[A] + P[x|\neg A]P[\neg A] + P[x|J]P[J]} \quad (11)$$

6 Current problems

I actually think I have done two mistakes in the implementation. Both are the same. I have erroneously replaced $P[Data|State]$ with $P[State|Data]$.

The first occurrence of this is in the viterbi derivation. I always thought I had to obtain the probability of being in a state given the data, while Vitervi actually requires the probability of the data given a state.

The same error occurred when I was doing the inference for the specific identity:

$$\prod_{j=1}^4 P[D_j | ID = S_i[j]]. \text{ I was actually taking } \prod_{j=1}^4 P[ID = S_i[j] | D_j]$$