Computer Vision Lab Class: Introduction to Pattern Recognition

Pierre-Yves Baudin, Régis Behmo

May 14, 2009

1 Introduction

The purpose of this lab class is to introduce a method for detecting patterns (i.e: objects) in images. We will focus on the star model [**] due to its simplicity and efficience.

We will focus on the problem of face detection. The pictorial structure (i.e: the face) is composed of five parts: the nose, the left eye, the right eye, the left corner of the mouth and the right corner of the mouth. Undirected connections are established between the nose and every other part.

2 Theory

Our model is represented by a undirected graph $G = (V, E)$, where $V = (v_1, v_2, v_3, v_4, v_5)$ are the vertices corresponding to each of the part and $E = \{(1, 2), (1, 3), (1, 4), (1, 5)\}$ are the (undirected) graph edges corresponding to the aforementioned connections between the nose (node 1) and the other parts. An instance of the object is given by a configuration $x = (x_1, x_2, x_3, x_4, x_5)$, where x_i is the location, in terms of pixel coordinates, of node v_i in the image. Let $I = \{1, 2, 3, 4, 5\}$ be the set of indices of the nodes and $\mathcal{N}(i)$ the set of indices of the neighbors of node i.

In order to match a pictorial structure to an image, we need to define a cost functional (also called energy) to be minimized. The cost of a configuration depends on how well each part matches the image data and how well the relative positions of the parts agree with the model. In a statistical framework, the *maximum a posteriori* criterion (MAP) is equivalent to such a cost functional. The posterior probability is $P(X|I, \theta)$, where I is the test image and θ is the set of parameters that define the model. The configuration X is now a vector of random variables. The MAP criterion is defined by:

$$\hat{x} = \arg\max_x P(x|I, \theta) \tag{1}$$

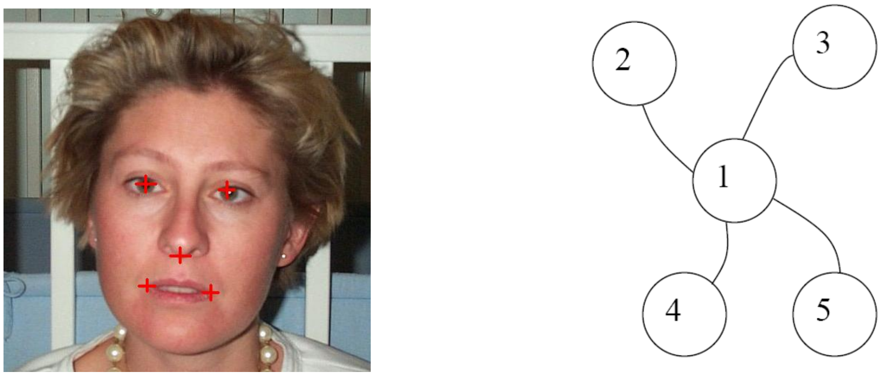

Figure 1: (left) Example from the training data base showing the location of the attributes we are trying to detect. (right) The structure of the learned model.

where $\hat{x}$ is the estimate of the unknown configuration $x = (x_1, x_2, x_3, x_4, x_5)$. It is shown that the posterior probability can be re-written in terms of unary and binary potentials:

$$P(X|I,\theta) = \frac{1}{Z} \prod_{k=1}^{5} \Phi_k(X_k) \prod_{(i,j)\in E} \Psi_{i,j}(X_i, X_j) \qquad (2)$$

$$E = \{(1,2),(1,3),(1,4),(1,5)\} \qquad (3)$$

where $\Phi_k(x_k)$ is the unary potential of the node k at coordinate x_k, and $\Psi_{i,j}(x_i, x_j)$ is the binary potential of pair of nodes (i,j), at coordinates x_i and x_j. The unary potentials represent how well the appearance of the image at location x_k matches the typical appearance of part k. The binary potentials are used to encode probabilistic location constraints ("the left eye is located above and left of the nose"...). We consider that both unary and binary potentials functions were computed in a previous training phase and are known.

However, brute-force maximization - i.e. directly maximizing the MAP with respect to x_1, x_2, x_3, x_4, x_5- is generaly impossible. Thus the need for appropriate algorithms. The Max-product algorithm provides an efficient and elegant way of computing:

$$B_i(x_i) = \max_{x_j,\, j\in\mathcal{N}(i)} P(x_1, x_2, x_3, x_4, x_5) \; i \in I$$

For instance, $B_1(x_1)$ can then be expressed as:

$$
\begin{aligned}
B_1(x_1) = \max_{x_2,x_3,x_4,x_5} P(x|I,\theta) \propto \Phi_1(x_1) \times \max_{x_2} \Phi_2(x_2)\Psi_{1,2}(x_1,x_2) \times \max_{x_3} \Phi_3(x_3)\Psi_{1,3}(x_1,x_3) \\
\times \Phi_4(x_4)\Psi_{1,4}(x_1,x_4) \times \max_{x_5} \Phi_5(x_5)\Psi_{1,5}(x_1,x_5) \\
= \Phi_1(x_1) \prod_{k \in \mathcal{N}(1)} \max_{x_k} \Phi_k(x_k)\Psi_{k,1}(x_k,x_1) \qquad (4) \\
= \Phi_1(x_1) \prod_{k \in \mathcal{N}(1)} \mu_{k \to 1}(x_1) \qquad (5)
\end{aligned}
$$

with

$$
\mu_{j \to 1}(x_1) = \max_{x_j} \Phi_j(x_j)\Psi_{1,j}(x_1,x_j) \qquad (6)
$$

where $\mu_{j \to 1}(x_1)$ is called the *message* sent from node j to node 1. We can now compute the estimate $\hat{x}_1$ of x_1 by finding the $\arg\max_{x_1}$ in equation 6.

$$
\hat{x}_1 \quad = \quad \arg\max_{x_1} B_1(x_1) \qquad (7)
$$

To compute the estimate $\hat{x}_j$ of the leaf x_j, the same scheme applies ($\forall j \neq 1$):

$$
\begin{aligned}
\hat{x}_j &= \arg\max_{x_j} B_j(x_j) \\
B_j(x_j) &= \Phi_j(x_j) \times \max_{x_1} \Phi_1(x_1)\Psi_{1,j}(x_1,x_j) \\
&\times \prod_{k \in \mathcal{N}(1)\backslash j} \max_{x_k} \Phi_k(x_k)\Psi_{k,1}(x_k,x_1) \\
&= \Phi_j(x_j)\mu_{1 \to j}(x_j)
\end{aligned}
$$

where

$$
\mu_{1 \to j}(x_j) = \max_{x_1} \Phi_1(x_1)\Psi_{1,j}(x_1,x_j) \prod_{k \in \mathcal{N}(1)\backslash j} \mu_{i \to 1}(x_1) \ (j \neq 1) \qquad (8)
$$

For instance, when node 1 (nose) sends a message to node 2 (left eye) it combines the messages μ_3, μ_4, μ_4 it has received from all other nodes in the graph (right eye, left mouth, right mouth) with its own observation potential, $\Phi_1(X_1)$, and with the compatibility $\Psi(X_1, X_2)$ of node 1 with node 2 locations.

The good thing is we already computed the $\mu_{i \to 1}(x_1)$, $(i \in I\backslash 1)$ when computing $\hat{x}_1$, so a large part of the job is already done. Notice that we now have a general form for the message:

$$
\mu_{i \to j}(x_j) = \max_{x_i} \Phi_i(x_i)\Psi_{i,j}(x_i,x_j) \prod_{k \in \mathcal{N}(i)\backslash j} \mu_{k \to i}(x_i) \qquad (9)
$$

which was already true for $\mu_{j \to 1}(x_1)$, except that it was simpler (since j has no other neighbor than 1).

3 Work to do

The unary and binary potential functions were previously computed, so you do not need to learn the appearance and compatibility models. Your job consists in computing the optimal locations $\hat{x}_i$ of all nodes:

1. Evaluate the messages sent from the leaf nodes to the root (equation 6)
2. Evaluate the messages sent from the root to the leaves (equation 8)
3. Infer the positions $\hat{x}_j$ of the all nodes (equations 7 and **??**)

You should send us back a report containing .... The report should be handed back by ?/?. Afterwards, the solutions are going to be available and no report will be accepted.

3.1 Data

We provide you with a series of test images (which were not used for learning the potentials), and corresponding potentials. The unary potentials (files ...) are given as images of the same size as their respective test image. There are five unary potential images for each test image, one for each facial attribute. The binary potentials (file ...) are given as four gaussian distributions (one per graph edge).

3.2 Structure of the code

We provide you with a fully written main() (in “main.cpp”) function that instanciates a c++ class called MaxProduct (in “maxprod.cpp”). A few key member functions of this class are not written yet and it is your job to do so.

- xxx : complete this function to make the program do ...

References

[1]