MP4: Audio-visual Person Identification

ECE 417 Fall 2017

Yuchen Liang

Zixu Zhang

October 24, 2017

1 Introduction

In this MP, we use a probability method to fuse both audio and visual trainning datas to conduct audio-visual person identification experiments. We constructed Gaussian Mixture Models (GMM) with audio ceptrum datas to estimate probability distrution of each person. Moreover, we also estimate posterior probability of person label given image principle components with 10 nearest neighbor (10-NN). Moreovre, we also tested identification accuracy of audio recognization GMM and visual recognization by 10-NN, and compared those results with audio-visual person identification experiments.

2 Method

2.1 Gaussian Mixture Model

If we assume that $\{y_i\}$ are set of classifier labels, and $\vec{x}$ is vector of data that we want to classifer, by Baye's rule, we will have posterior as:

$$P_{Y|X}(y_i|\vec{x}) = \frac{P(y_i \cap \vec{x})}{P_X(\vec{x})} = \frac{P_{X|Y}(\vec{x}|y_i)P_Y(y_i)}{P_X(\vec{x})}$$

A Bayesian classifer will have the following form:

$$\hat{y} = \arg\max_y P_{Y|X}(y|\vec{x}) = \arg\max_y \frac{P_{X,Y}(\vec{x}, y)}{P_X(\vec{x})} = \arg\max_y P_{X,Y}(\vec{x}, y)$$

In the same sense, we can take natural log of joint probabilty as the sum of the log likelihood $P_{X|Y}(\vec{x}|y_i)$ and log prior $P_Y(y_i)$. Thus, we will have our classifier as:

$$\hat{y} = \arg\max_y [\ln P_{X|Y}(\vec{x}|y_i) + \ln P_Y(y_i)]$$

Unlike the prior, which is deterministic by trainning data, we need use a good model for likelihood estimation to have a accurate classifer. Gaussian Mixture Model is one of great models to obtain likelihood of each class y_i. It is defined as the weighted sum of K different gaussian distributions with weight constraints to ensure that it is a valid distribution over probabilty space:

$$P_{X|Y}(\vec{x}|y_i) = \sum_{k=1}^{K} c_k \mathcal{N}(\vec{x}; \vec{\mu}_{y_i k}, \Sigma_{y_i k}), \quad c_k \geq 0, \quad \sum_{k=1}^{K} c_k = 1$$

In order to have an ideal GMM model, we will need to initialize the model with K Gaussian distributions with same covariance matrix Σ and evenly distributed weight c_i. In this MP, we use a model with 2 Gaussian distributions, and the initialization is done by `Matlab`'s `kmeans` function at line 90 of `gmm_train.m`.

2.2 Expectation Maximization

The above GMM of each class is found using the Expectation Maximization (EM) algorithm. EM algorithm is an iterative method to find the local maximum likelihood. More specifically, it maximizes the log likelihood of the independent observation data X in class of datum Y to find an optimal model Λ for GMM.

$$\mathcal{L} = \ln p(X|Y,\Lambda) = \ln \prod_{n=1}^{N} p_{X|Y,\Lambda}(\vec{x}_n|y_n,\Lambda) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} c_k \mathcal{N}(\vec{x}_n;\vec{\mu}_k,\Sigma_k) \right)$$

Evaluation of log likelihood is performed in function `gmm_eval.m`.

There are two major steps in each iteration of EM algorithm, both of which lie in `gmm_train.m`. The E step calculates the gamma probability $\gamma_{i,k}$, which is the posterior probability of the kth Gaussian Model, given the current observation $\vec{x}_i$. After this step, the current likelihood will be calculated to check if it has reached its local maximum. While the likelihood is still converging, the M step will use the calculated gamma probability to re-estimate the GMM model for next iteration of maximization. For E step, $\gamma_k(n)$ means the posterior probability of k^{th} distribution in GMM given n^{th} training data. It can be written as:

$$\gamma_k(n) = P_{K|X}(k|\vec{x}_n) = \frac{P(k)P(\vec{x}_n|k)}{P(\vec{x}_n)} = \frac{c_k \mathcal{N}(\vec{x}_n;\vec{\mu}_k,\Sigma_k)}{\sum_{\ell=k}^{K-1} c_\ell \mathcal{N}(\vec{x}_n;\vec{\mu}_\ell,\Sigma_\ell)}$$

$P(k)$ is the weight c_k of this distribution in current model. $\gamma_k(n)$ is calculated in function `estep`, and results in a $\mathbb{R}^{N\times K}$ matrix, where N is the number of training data, and K is number of distribution in GMM.

In this MP, we set up a threshold for log likelihood convergence. That is saying if the difference between current log likelihood and previous log likelihood is larger than the threshold, M step is called to optimize the current model. A general expression of M step is described as following:

$$c_k = \frac{\sum_{n=0}^{N-1} \gamma_k(n)}{N} \qquad \vec{\mu}_k = \frac{\sum_{n=0}^{N-1} \gamma_k(n)\vec{x}_n}{\sum_{n=0}^{N-1} \gamma_k(n)}$$

$$\Sigma_k = \frac{\sum_{n=0}^{N-1} \gamma_k(n)(\vec{x}_n - \vec{\mu}_k)(\vec{x}_n - \vec{\mu}_k)^T}{\sum_{n=0}^{N-1} \gamma_k(n)}$$

By assuming that D dimensions are independent in one training vector, we only consider diagonal elements of covariance matrix. While we do not have space to proof the convergence, a local maximum of log likelihood will be achieved after certain iterations. In our MP, this implemented from line 189 to 219 in function `mstep`. After each iteration, we will have a new model of GMM which consists a $D \times K$ matrix of means, a $D \times K$ matrix of diagonal elements of covariance matrix, and a $K \times 1$ vector of weight.

2.3 Audiovisual Fusion

In previous steps, we fit a GMM for each audio class using EM algorithm. Therefore, the probability of a test sample given the class C_i is given by:

$$P(X_{t,a}|C_i) = \prod_{n=1}^{N} P(\vec{x}_n|C_i) = \prod_{n=1}^{N} (\sum_{k=1}^{K} c_k \mathcal{N}(\vec{x}_n; \vec{\mu}_k, \Sigma_k))$$

where c_k, $\vec{\mu}_k$, and Σ_k are derived from the EM algorithm. (In this MP, we calculate the mean of log likelihood, which is the same as taking the square root of $P(X_{t,a}|C_i)$.) We also achieve the kNN posterior probability $P(C_i|X_{t,v})$ by counting the number of the k nearest neighbors, say n_i, for class C_i. So the posterior probability is given by:

$$P(C_i|X_{t,v}) = \frac{n_i}{k}$$

By Baye's Rule,

$$P(X_{t,a}, C_i|X_{t,v}) = P(X_{t,a}|C_i, X_{t,v})P(C_i|X_{t,v}) = P(X_{t,a}|C_i)P(C_i|X_{t,v})$$

since $X_{t,a}$ and $X_{t,v}$ are independent, and

$$P(X_{t,a}, C_i, X_{t,v}) = P(X_{t,a}, C_i|X_{t,v})P(X_{t,v}) \propto P(X_{t,a}, C_i|X_{t,v})$$

assuming the test videos and audios occur equally likely. Hence, we can classify the audiovisual combination using the following metric:

$$i = \arg\max_i P(C_i|X_{t,a}, X_{t,v}) = \arg\max_i P(C_i, X_{t,a}, X_{t,v}) = \arg\max_i P(X_{t,a}|C_i)P(C_i|X_{t,v})$$

In practice, we use a scaled version, namely $P(X_{t,a}|C_i)^w P(C_i|X_{t,v})^{1-w}$, because the former is a pdf while the latter is a pmf, so a weight is added to balance the two in case the pdf gets too large. Reasons for weight is that audio recognition treat pdf as its likelihood, and it can be any positive number, while visual recognition posterior use pmf, which ranges from 0 to 1. Since the determinant of covariance matrix of GMM in audio test is small, we will have a really large number for pdf in audio test, weight is important for audio-visual fusion.

3 Results

Recognition accuracies of audio, visual and audio-visual fusion are listed in Table 1, 2, and 3.

Table 1: Person ID Accuracy: Audio

%	A	B	C	D	Ave
Accuracy	50	70	60	90	67.5

Table 2: Person ID Accuracy: Visual

%	A	B	C	D	Ave
Accuracy	100	80	10	80	67.5

With results in Table 3, we can see that if we choose $w = 0.5$ we will the most accurate recognition in our test data.

Table 3: Person ID Accuracy: Audio + Visual

%	$w = 0.1$	$w = 0.2$	$w = 0.3$	$w = 0.4$	$w = 0.5$	$w = 0.6$	$w = 0.7$	$w = 0.8$	$w = 0.9$
A	97	97	95	94	93	88	86	82	75
B	74	74	76	75	74	73	75	72	70
C	22	32	50	62	65	68	68	67	67
D	97	99	99	99	99	99	99	99	97
Ave	72.5	75.5	80	82.5	82.75	82	82	80	77.25

4 Discussion

Comparing the result from mixed audio-visual with purely audio or visual, we find that at any chosen weight, the accuracy from audio+visual data is at least as high as the lower of the audio or the visual data, and many cases much higher than either of the two. This is because in the AV tests, the joint probability of audio and visual data rather than just one of them is calculated, providing more evidence of a given class/person/label. The AV accuracy gains its maximum accuracy at weight 0.5, and it gradually becomes lower towards either end. This is because as weight increases, a larger weight/reliability is given onto audio samples, and it becomes closer to pure audio tests (and the same happens for the other direction). An equal weight considers audio and visual samples equally, and aligns perfectly with the Bayes rule of HMM, thereby producing the maximum accuracy.