

Лабораторная работа №1. МАР-адаптация GMM.

Кудашев Олег

22 августа 2014 г.

1 Цель работы

Целью работы является реализация и использование алгоритма МАР-адаптации GMM для создания голосовых моделей дикторов, а также построение на его основе простейшей системы распознавания дикторов по голосу.

2 Основные понятия и формулы

Как показывают последние достижения в области распознавания дикторов, наиболее эффективной генеративной моделью голоса диктора является GMM (Gaussian Mixture Model), описывающая распределение акустических признаков речевых сегментов фонограммы этого диктора. Как правило, в качестве акустических признаков используются MFCC (Mel-Frequency Cepstral Coefficients) вместе с их первыми и вторыми производными. Кроме этого, в качестве простейшего механизма компенсации канальных искажений используют метод CMN (Cepstral Mean Normalization).

Основным ключевым вопросом, определяющим надежность методов идентификации (расознавания) дикторов, является способ построения GMM. Как известно, функция распределения GMM представляет собой взвешенную сумму гауссовых распределений и задается формулой:

$$f(x|\pi, \mu, \Sigma) = \sum_{c=1}^M \pi_c N(x|\mu_c, \Sigma_c) = \sum_{c=1}^M \frac{\pi_c}{(2\pi)^{D/2} |\Sigma_c|^{1/2}} e^{-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1} (x-\mu_c)}, \quad (1)$$

где D - размерность случайного вектора x ; M - количество гауссoid; μ_c , Σ_c , π_c - среднее значение, ковариационная матрица и вес гауссoidы. Пусть у нас имеется некоторая выборка независимых значений $X =$

$\{x_1, x_2, \dots, x_N\}$. Тогда задача построения GMM для данной выборки сводится к оценке параметров μ_c, Σ_c, π_c .

Наиболее распространенным критерием оценки этих параметров является метод ML (Maximum Likelihood), суть которого можно выразить формулой:

$$\hat{\theta}_{\text{ML}}(X) = \arg \max_{\theta} f(X|\theta) = \arg \max_{\theta} \prod_{t=1}^T f(x_t|\theta) . \quad (2)$$

У этого критерия есть ряд недостатков:

- Не существует аналитического решения уравнения 2 для GMM, позволяющего получить точные оценки параметров гауссойд. Вместо этого применяется итерационный алгоритм ЕМ (Expectation-Maximization), позволяющий приблизительно найти один из локальных максимумов.
- Существует проблема overfitting-a, т.е. переобучения параметров на локальные данные.
- Как следствие, критерий ML обладает низкой робастностью (обобщающей способностью).

Наряду с критерием ML, существует не менее распространенный критерий МАР(Maximum A Posteriori) - оценки. Суть его заключается в использовании априорной информации о распределении искомых параметров. Используя теорему Байеса, критерий МАР-оценки сводится к следующей формуле:

$$\hat{\theta}_{\text{МАР}}(X) = \arg \max_{\theta} \frac{f(X|\theta) g(\theta)}{\int_{\vartheta} f(X|\vartheta) g(\vartheta) d\vartheta} = \arg \max_{\theta} f(X|\theta) g(\theta) . \quad (3)$$

Как правило, если в качестве априорного распределения используется сопряженное семейство распределений, то МАР-оценка параметров сводится к алгебраическому решению. Так, для семейства GMM, МАР-оценка параметров принимает вид (Reynolds):

$$\hat{\pi}_c = \frac{\alpha_c^{\pi} N_c / T + (1 - \alpha_c^{\pi}) \pi_c}{\sum_{k=1}^M \hat{\pi}_k} , \quad (4)$$

$$\hat{\mu}_c = \alpha_c^{\mu} F_c + (1 - \alpha_c^{\mu}) \mu_c , \quad (5)$$

$$\hat{\Sigma}_c = \alpha_c^{\Sigma} S_c + (1 - \alpha_c^{\Sigma}) (\Sigma_c + \mu_c \mu_c^T) - \hat{\mu}_c \hat{\mu}_c^T , \quad (6)$$

где N_c , F_c , S_c - статистики Баума-Уэлша нулевого, первого и второго порядков, вычисляемые по формулам:

$$N_c = \sum_{t=1}^T \gamma_c(t) , \quad (7)$$

$$F_c = \frac{1}{N_c} \sum_{t=1}^T \gamma_c(t) x_t , \quad (8)$$

$$S_c = \frac{1}{N_c} \sum_{t=1}^T \gamma_c(t) x_t x_t^T , \quad (9)$$

$$\gamma_c(t) = \frac{\pi_c N(x_t | \mu_c, \Sigma_c)}{\sum_{k=1}^M \pi_k N(x_t | \mu_k, \Sigma_k)} , \quad (10)$$

а α_c^ρ , $\rho \in \{\pi, \mu, \Sigma\}$ - коэффициенты адаптации, которые можно найти по формуле:

$$\alpha_c^\rho = \frac{N_c}{N_c + r^\rho} , \quad (11)$$

где r^ρ - фиксированная величина, т.н. *релевантность*, которая изменяется в диапазоне от 0 до ∞ .

Как правило, в качестве априорного распределения выступает GMM, обученная алгоритмом ЕМ на большой и представительной базе обучения и носит название UBM (Universal Background Model).

Имея модель GMM диктора, мы имеем возможность проверять, принадлежит ли этому диктору тестовое произнесение. Существуют огромное количество способов проверки, но в данной работе мы ограничимся простейшим способом, а именно критерием Неймана-Пирсона:

$$\begin{aligned} \frac{1}{T} \log \frac{P(X | \hat{\pi}, \hat{\mu}, \hat{\Sigma})}{P(X | \pi, \mu, \Sigma)} = \frac{1}{T} \sum_{t=1}^T [& \log (\sum_{c=1}^M \hat{\pi}_c N(x_t | \hat{\mu}_c, \hat{\Sigma}_c)) - \\ & - \log (\sum_{c=1}^M \pi_c N(x_t | \mu_c, \Sigma_c))] , \end{aligned} \quad (12)$$

где π , μ , Σ , - параметры априорного распределения (UBM); $\hat{\pi}$, $\hat{\mu}$, $\hat{\Sigma}$ - MAP-оценка апостериорного распределения (GMM диктора).

3 Задание

3.1 Входные данные

Будем считать, что задача построения акустических признаков и выделения речевых сегментов фонограммы выполнена. Кроме этого, получены значения параметров априорного распределения - UBM. Таким образом, входными данными являются:

- Набор акустических признаков дикторов для построения голосовых моделей (файлы `*.features_bin` папки **models**). Файл `*.features_bin` имеет следующий бинарный формат:

```
[int, 1, размерность признаков - D]
[int, 1, количество признаков в файле - T]
[float, D, вектор признаков 1]
[float, D, вектор признаков 2]
...
[float, D, вектор признаков T]
```

Пример кода на Matlab, осуществляющий чтение файла признаков:

```
fileID = fopen(fileName);
D = fread(fileID, 1, 'int');
T = fread(fileID, 1, 'int');
features = fread(fileID, [D, T], 'float');
```

Пример кода на C++, осуществляющий чтение файла признаков:

```
#include <fstream>
#include <vector>
using namespace std;

ifstream featuresFile(fileName, ios::in | ios::binary);
int D = 0;
int T = 0;
featuresFile.read((char*)&D, sizeof(int));
featuresFile.read((char*)&T, sizeof(int));
vector<vector<float> > features(T);
for (size_t t = 0; t < T; t++)
{
    features[t].resize(D);
```

```

        featuresFile.read((char*)&features[t][0], sizeof(float) * D);
    }

```

- UBM (файл **ubm.gmm**). Для упрощения вычислений используется диагональная ковариационная матрица для каждой гауссойды. Формат файла *.gmm:

```

[int, 1, размерность признаков - D]
[int, 1, количество гауссойд - M]
[float, M, веса гауссойд]
[float, D * M , средние значения гауссойд]
[float, D * M , диагонали ковариационных матриц гауссойд]

```

Пример кода на Matlab, осуществляющий чтение файла *.gmm:

```

fileID = fopen(fileName);
D = fread(fileID, 1, 'int');
M = fread(fileID, 1, 'int');
pi = fread(fileID, M, 'float');
mu = fread(fileID, [D, M], 'float');
sigma = fread(fileID, [D, M], 'float');

```

Пример кода на C++, осуществляющий чтение файла *.gmm:

```

ifstream gmmFile(fileName, ios::in | ios::binary);
int D = 0;
int M = 0;
gmmFile.read((char*)&D, sizeof(int));
gmmFile.read((char*)&M, sizeof(int));
vector<float> pi(M);
vector<vector<float> > > mu(M);
vector<vector<float> > > sigma(M);
gmmFile.read((char*)&pi[0], sizeof(float) * M);
for (int c = 0; c < M; c++)
{
    mu[c].resize(D);
    gmmFile.read((char*)&mu[c][0], sizeof(float) * D);
}
for (int c = 0; c < M; c++)
{
    sigma[c].resize(D);
    gmmFile.read((char*)&sigma[c][0], sizeof(float) * D);
}

```

- Набор тестовых акустических признаков для определения, принадлежит ли высказывание искомому диктору (файлы `*.features_bin` папки **tests**)
- Протоколы target- и impostor- сравнений (папка **protocols**) для верификации и подсчета ошибки системы распознавания дикторов. Протоколы представляют из себя текстовые файлы, содержащие пары сравнений моделей дикторов и тестовых файлов, разделенных знаком табуляции.

3.2 Задание 1

Для каждого входного файла `*.features_bin` папки **models** получить апостериорные оценки параметров распределения GMM в соответствии с формулами 4, 5, 6, взяв в качестве априорного распределения параметров значения UBM из файла `ubm.gmm`. Величину r^p можно установить равную 20. Для упрощения вычислений, можно провести адаптацию только средних значений гауссойд, оставив веса и ковариационные матрицы без изменений.

Получившиеся постериорные оценки сохранить в соответствующий файл [имя входного файла без расширения].`.gmm` в строгом соответствии с форматом файла `*.gmm`

3.3 Задание 2

Для каждой пары сравнений target- и impostor- протоколов подсчитать среднее значение \log правдоподобия в соответствии с формулой 12, используя построенные GMM дикторов папки **models** и акустические признаки тестовых файлов папки **tests**.

Результаты записать в текстовые файлы `targets_answers.txt` и `impostors_answers.txt`, соблюдая порядок сравнений.

3.4 Задание 3

Для полученных оценок target- и impostor- сравнений подсчитать величину равновероятной ошибки EER (Equal Error Rate).

3.5 Выходные данные

Таким образом, выходными данными являются:

- Файлы *.gmm, построенные для каждого файла *.features_bin папки **models**.
- Файлы targets_answers.txt и impostors_answers.txt.
- Значение EER.

Все выходные данные необходимо присылать на e-mail kudashev@speechpro.com в zip-архиве [Фамилия магистранта]_lab1.zip

4 Контакты

Все интересующие вопросы можно отправлять на e-mail