

Лабораторная работа №3. Алгоритм Витерби-пересегментации.

Кудашев Олег

11 ноября 2013 г.

1 Цель работы

Целью работы является реализация и исследование алгоритма декодирования Витерби, а также применение его к задаче пересегментации речевых сегментов дикторов на фонограмме.

2 Основные понятия и формулы

2.1 НММ, алгоритм Витерби

НММ (Hidden Markov Model) - распространенная статистическая модель, позволяющая моделировать последовательность наблюдаемых значений $X = \{x_1, \dots, x_N\}$ через так называемые «скрытые» состояния $Z = \{z_1, \dots, z_N\}$. При этом скрытые состояния представляют собой бинарные индекс-векторы, $z_t^k \in \{0, 1\}$, $\sum_{k=1}^K z_t^k = 1$.

Основная идея состоит в том, что текущее наблюдаемое значение x_t зависит только от скрытого состояния z_t . А текущее скрытое состояние z_t зависит от предыдущего состояния z_{t-1} и от вероятности перехода из состояния z_{t-1} в состояние z_t . Априорная вероятность состояния z_1 задается вектором π .

Таким образом, функция правдоподобия имеет вид:

$$p(X, Z|A, \pi, \phi) = p(z_1|\pi) \left[\prod_{n=2}^N p(z_n|z_{n-1}, A) \right] \prod_{m=1}^N p(x_m|z_m, \phi) \quad (1)$$

$$p(z_1|\pi) = \prod_{k=1}^K \pi_k^{z_1^k} \quad (2)$$

$$p(z_n|z_{n-1}, A) = \prod_{k=1}^K \prod_{l=1}^K a_{lk}^{z_n^k z_{n-1}^l} \quad (3)$$

$$p(x_n|z_n, \phi) = \prod_{k=1}^K p(x_n|\phi_k)^{z_n^k}, \quad (4)$$

где A – матрица вероятностей перехода; π – априорная вероятность состояния z_1 ; ϕ_k – параметры распределения наблюдаемых значений для скрытого состояния k .

Существует целый ряд задач, связанных с НММ:

- Подсчет функции правдоподобия (1) при известных величинах X, Z, A, π, ϕ .
- Нахождение наиболее вероятной последовательности скрытых состояний $\{z_1, \dots, z_N\}$, имея набор наблюдаемых значений $\{x_1, \dots, x_N\}$ и значения параметров A, π, ϕ .
- Обучение параметров НММ: A, π, ϕ .

В данной работе нас будет интересовать только одна задача, а именно задача нахождения наиболее вероятной последовательности скрытых состояний (т.н. декодирование). Для решения этой задачи применяется алгоритм Витерби, который является алгоритмом динамического программирования.

Пусть у нас есть некая НММ с параметрами A, π, ϕ а также последовательность наблюдаемых значений $\{x_1, \dots, x_N\}$. Тогда наиболее вероятная последовательность скрытых состояний $\{z_1, \dots, z_N\}$ задается рекуррентным соотношением:

$$\begin{aligned} V_1^k &= p(x_1|\phi_k) \cdot \pi_k \\ V_t^k &= p(x_t|\phi_k) \cdot \max_l (a_{lk} \cdot V_{t-1}^l), \end{aligned} \quad (5)$$

где V_1^k – значение функции правдоподобия (1) для наиболее вероятной последовательности скрытых состояний, оканчивающихся состояни-

ем k . Декодирование Витерби производится по формуле:

$$\begin{aligned} z_N &= \arg \max_k (V_N^k) \\ z_{t-1} &= Ptr(z_t, t), \end{aligned} \quad (6)$$

где $Ptr(z_t, t)$ - функция, возвращающая значение z_{t-1} , использованное для подсчета V_t^k в формуле (5), если $t > 1$, или k , если $t = 1$.

2.2 Пересегментация речевых сегментов дикторов на фонограмме

Пусть у нас есть некоторая начальная сегментация речевых сегментов дикторов на фонограмме. Зачастую, эта начальная сегментация получена достаточно грубым алгоритмом кластеризации и содержит ошибки. Поэтому, необходимо провести уточнение границ этих речевых сегментов. Для этих целей используется описанный выше алгоритм Витерби.

В качестве наблюдаемых значений выступает последовательность акустических признаков (MFCC). В качестве скрытых состояний выступают индексы дикторов, которым принадлежит соответствующий акустический признак. Априорные значения π_k для каждого диктора равны $1/S$, где S - число дикторов. Матрица перехода A фиксированна и, как правило, задается следующим образом: все диагональные элементы равны некоторому значению p , близкому к 1 (≈ 0.95), остальные элементы равны $(1-p)/(S-1)$. Функция распределения скрытых состояний задается в виде GMM:

$$\begin{aligned} p(x_t|\phi_k) &= N(x_t|w_k, \mu_k, \Sigma_k) = \\ &= \sum_{c=1}^M \frac{w_k^c}{(2\pi)^{D/2} |\Sigma_k^c|} e^{-\frac{1}{2}(x_t - \mu_k^c)^T \Sigma_k^{c-1} (x_t - \mu_k^c)}, \end{aligned} \quad (7)$$

где D - размерность случайного вектора x_t ; M - количество гауссoid; μ_k^c , Σ_k^c , w_k^c - среднее значение, ковариационная матрица и вес гауссойды c диктора k .

3 Задание

3.1 Входные данные

Для разработки и тестирования алгоритма Витерби-пересегментации подобрана база, состоящая из диалогов дикторов в телефонном канале.

Входными данными являются:

- Набор акустических признаков фонограмм (файлы *.features_bin папки **base**). 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);
}
```

- Файлы *.1.gmm и *.2.gmm папки **base**, содержащие параметры моделей дикторов. Для упрощения вычислений использовались диагональные ковариационные матрицы. 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);
}

```

- Файлы *.ref.indx папки **base**, содержащие информацию о принадлежности акустических признаков дикторам в соответствии с идеальной разметкой. Эти файлы необходимы для оценки качества и настройки алгоритма пересегментации. Файлы *.indx имеют следующий бинарный формат:

```
[int, 1, количество признаков в файле - K]
[int, K, индексы дикторов (от 0 до S-1)]
```

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

```
fileID = fopen(fileName);
K = fread(fileID, 1, 'int');
indexes = fread(fileID, K, 'int');
```

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

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

ifstream pointsFile(fileName, ios::in | ios::binary);
int K = 0;
pointsFile.read((char*)&K, sizeof(int));
vector<int> indexes(K);
pointsFile.read((char*)&indexes[0], sizeof(int) * K);
```

3.2 Задание

Реализовать алгоритм Витерби в соответствии с формулами (5, 6). Используя входные данные, выполнить декодирование Витерби и записать результат (последовательность скрытых состояний) в соответствующий файл *.sys.indx в строгом соответствии с форматом.

Примечание: вместо формул (5, 6) целесообразно использовать их логарифмы.

3.3 Выходные данные

Выходными данными являются файлы [имя файла без расширения].sys.indx.

Все выходные данные необходимо присылать на e-mail kudashev@speechpro.com в zip-архиве [Фамилия магистранта]_lab3.zip.

4 Контакты

Все интересующие вопросы можно отправлять на e-mail kudashev@speechpro.com.