

Лабораторная работа №2. Байесовский информационный критерий.

Кудашев Олег

24 апреля 2014 г.

1 Цель работы

Целью работы является реализация и исследование байесовского информационного критерия (Bayesian Information Criterion, BIC), а также применение его к задаче поиска точек смены дикторов на фонограмме.

2 Основные понятия и формулы

2.1 Алгоритм поиска точек смены дикторов на основе BIC

BIC позволяет производить выбор из набора моделей, описывающих распределение данных. Он является примером реализации принципа „бритвы Оккама”, суть которого состоит в том, что из набора гипотез, одинаково точно объясняющих какое-то явление, выбрать следует наиболее простую. Выпишем формулу BIC:

$$BIC(M) = \ln p(X|\theta_{MAP}) - \frac{\lambda}{2}\nu(M) \ln N, \quad (1)$$

где M - оцениваемая модель; X - данные; N - количество данных; θ_{MAP} - MAP-оценка параметров модели; $\nu(M)$ - количество параметров модели; λ - настраиваемое значение (порог срабатывания критерия).

Второе слагаемое в формуле (1) представляет собой штрафную функцию, которая уменьшает величину BIC при увеличении количества параметров. Таким образом, из набора моделей выбирается та, значение BIC для которой больше.

Применим BIC для задачи поиска точек смены дикторов на фонограмме. Пусть нам даны два речевых сегмента, содержащие данные X_1 и X_2 соответственно. Рассмотрим две гипотезы:

1. M_1 - данные из обоих речевых сегментов подчинены одному гауссову распределению (μ, Σ) .
2. M_2 - данные из первого речевого сегмента подчинены гауссову распределению (μ_1, Σ_1) , а данные из второго речевого сегмента подчинены гауссову распределению (μ_2, Σ_2) .

Рассмотрим величину:

$$\begin{aligned} \Delta BIC(M_1, M_2) &= BIC(M_1) - BIC(M_2) = \\ &= \ln \frac{L(X_1, X_2 | \mu, \Sigma)}{L(X_1 | \mu_1, \Sigma_1) L(X_2 | \mu_2, \Sigma_2)} - \frac{\lambda}{2} (\nu(M_1) - \nu(M_2)) \ln(N_1 + N_2), \end{aligned} \quad (2)$$

где N_1, N_2 - количество данных первого и второго сегментов соответственно.

Формула (2) может быть переписана в следующем виде:

$$\begin{aligned} \Delta BIC(M_1, M_2) &= \\ &= \frac{1}{2} (N_1 \ln |\Sigma_1| + N_2 \ln |\Sigma_2| - (N_1 + N_2) \ln |\Sigma_{1,2}| + \lambda \beta \ln(N_1 + N_2)), \end{aligned} \quad (3)$$

где $\Sigma_1, \Sigma_2, \Sigma_{1,2}$ - ковариационные матрицы, полученные на первом речевом сегменте, на втором и на обоих соответственно; β - количество параметров модели ($\beta = d$ при диагональной ковариационной матрице и $\beta = d(d+1)/2$ при полной ковариационной матрице).

Величина $\Delta BIC < 0$ означает, что наиболее предпочтительной является модель M_2 , следовательно, речевые сегменты принадлежат различным дикторам.

Простейший алгоритм поиска точек смены дикторов на основе BIC может выглядеть следующим образом:

1. Все акустические признаки речевых сегментов „сливаются” в один непрерывный массив данных. В дальнейшем работа идет только в рамках этого массива.
2. По всему массиву пробегается окно фиксированной длины с фиксированным шагом. Окно разбивается на две равные части, левой части окна соответствуют данные X_1 , правой - X_2 . Для точки, являющейся серединой окна, рассчитывается величина - ΔBIC по формуле (3).

3. Среди всех полученных таким образом значений выбираются локальные минимумы, меньшие нуля.
4. Последовательно среди всех полученных локальных минимумов выбираются наименьшие. При этом расстояние до уже выбранных локальных минимумов не должно превышать заранее заданной величины. Точки, соответствующие выбранным значениям и берутся в качестве точек смены дикторов.

Необходимо отметить, что описанный способ поиска точек смены дикторов начинает работать только если окна с данными X_1 и X_2 имеют размер не менее 1,5 - 2 секунд, что делает его неприменимым для задач, где происходит частая смена дикторов (активные телефонные диалоги, совещания и т.п.). Кроме этого, очевидным недостатком алгоритма является необходимость настраивания порогового значения λ .

2.2 Оценка качества системы

Для оценки качества алгоритма поиска точек смены дикторов используются значения :

$$PRC(Precision) = \frac{N_{correct}}{N_{sys}} \cdot 100\% \quad (4)$$

$$RCL(Recall) = \frac{N_{correct}}{N_{ref}} \cdot 100\% \quad (5)$$

$$F_{measure} = \frac{2 \cdot PRC \cdot PCL}{(PRC + PCL)} , \quad (6)$$

$N_{correct}$ - количество корректно определенных точек смены дикторов;
 N_{sys} - количество всех точек смены дикторов, выданных алгоритмом;
 N_{ref} - количество всех точек смены дикторов идеальной разметки.

3 Задание

3.1 Входные данные

В качестве тестового полигона алгоритма поиска точек смены дикторов будут использованы аудиозаписи радио „Свобода”, где смена дикторов происходит в среднем не чаще одного раза в 2-3 секунды. Будем считать, что задача построения акустических признаков и выделения речевых сегментов фонограмм выполнена. В качестве акустических признаков

использованы первые 20 коэффициентов MFCC (без C_0), построенные с шагом 10 мс и размером окна 20 мс.

Итак, входными данными являются:

- Набор акустических признаков фонограмм базы развития (файлы `*.features_bin` папки **dev**) для определения параметров алгоритма поиска точек смены дикторов. 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 `*.ref.pnts` папки `dev`, содержащие информацию о положениях точек смены дикторов (в отсчетах MFCC). Эти файлы необходимы для оценки качества и настройки алгоритма поиска точек смены дикторов на базе развития. Файлы `*.pnts` имеют следующий бинарный формат:

```
[int, 1, количество точек в файле - K]
[int, K, точки смены дикторов в отсчетах MFCC]
```

Пример кода на Matlab, осуществляющий чтение файла `*.pnts`:

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

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

```
#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> points(K);
pointsFile.read((char*)&points[0], sizeof(int) * K);
```

- Набор акустических признаков фонограмм базы тестирования (файлы `*.features_bin` папки `test`).

3.2 Задание 1

Реализовать алгоритм поиска точек смены дикторов на основе ВИС.

3.3 Задание 2

Используя базу развития (папка `dev`), осуществить настройку алгоритма таким образом, чтобы величина RCL (см. формулу (5)) по всей базе была не менее 70%. В качестве идеальной разметки использовать файлы `*.ref.pnts`. Основными настройками алгоритма являются: размер и шаг окна, значение порога λ , форма ковариационной матрицы (полная или диагональная).

3.4 Задание 3

Используя полученные настройки, произвести поиск точек смены дикторов на базе тестирования (папка `test`). Результат сохранить в файлы `[имя файла без расширения].sys.pnts` в строгом соответствии с форматом.

3.5 Выходные данные

Выходными данными являются файлы `[имя файла без расширения].sys.pnts` базы тестирования

Все выходные данные необходимо присылать на e-mail в zip-архиве `[Фамилия магистранта]_lab2.zip`.

4 Контакты

Все интересующие вопросы можно отправлять на e-mail