

Differentiable Dynamic Programming for Time Series Alignment

Тимур Гарипов, 517 группа
Татьяна Шолохова, 517 группа
Павел Коваленко, 517 группа
Саня Щербаков, 522 группа

9 июня 2018

Рассматривается задача выравнивания временных рядов.

Дана нотная запись музыкальной композиции и аудиозапись этой композиции. Требуется каждому моменту времени в аудиозаписи сопоставить ноту, играемую в этот момент.



В работе был использован датасет Bach 10, состоящий из 10 аудиозаписей фрагментов хоралов Баха, продолжительность фрагментов — от 25 до 40 секунд.

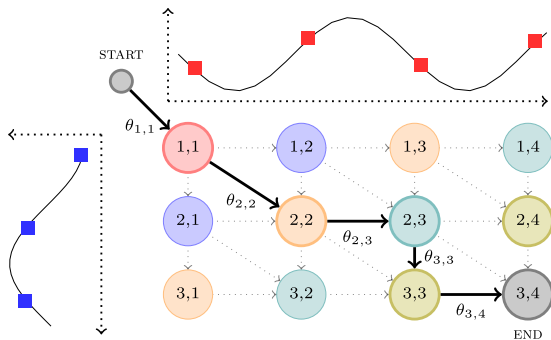
Каждая запись состоит из четырех дорожек, соответствующих четырем инструментам — скрипка, кларнет, саксофон и фагот. Есть как записи отдельных дорожек, так и сводная запись всех инструментов.

Для каждой дорожки дана ее идеальная нотная запись, однако фактическая игра от нее немного отклоняется. Также для всех дорожек дано правильное выравнивание аудио- и нотной записи. Для инструментов в выборке представлено от 15 до 25 различных нот.

Для обеих последовательностей — нотной и аудиозаписи — выделим с равными интервалами ключевые точки, для которых будем искать выравнивание.

Обозначим за N_A число нот в последовательности, а в аудиозаписи возьмем N_B точек с равными интервалами. Тогда выравнивание можно представить в виде бинарной матрицы $Y \in \{0, 1\}^{N_A \times N_B}$. Единица в позиции (i, j) означает, что в j -й момент времени проигрывалась i -я нота.

Предположим, что последовательность нот при игре не изменилась и ни одна из нот не была пропущена. Тогда выравнивание можно представить в виде пути в матрице Y из клетки $(1, 1)$ в клетку (N_A, N_B) , при этом разрешены перемещения только вправо, вниз и вправо-вниз. Пример выравнивания — на рисунке ниже.



Потребуем, чтобы N_B было больше N_A . Тогда можно потребовать, чтобы в каждый момент времени играла только одна нота, то есть для каждого момента времени требуется предсказать, какая нота сейчас играет.

Для матрицы это ограничение означает, что в каждом столбце может быть не больше одной единицы. Для пути в графе это ограничение равносильно запрету переходов вниз.

Метрика качества — *mean absolute deviation* — суммарное (по моментам времени) отклонение индекса предсказанной ноты от истинного индекса.

В статье предложено использовать следующие признаки для аудиодорожки:

- MFCC признаки — первые 5 коэффициентов.
- Root Mean Square Energy — энергия фрейма.
- Spectral Centroid — средняя частота спектра во фрейме.
- Spectral Bandwidth — разброс частот спектра во фрейме.

Были использованы реализации этих признаков из библиотеки librosa.

Разбиваем датасет на две части. На первой части обучаем классификатор по числу различных нот — предсказываем вероятность того, что в момент времени t играет нота i .

Для второй части датасета построим матрицу $\theta \in \mathbb{R}^{N_A \times N_B}$.

θ_{ij} соответствует отрицательному логарифму вероятности того, что в момент времени j играет нота номер i , то есть штрафу за предсказание ноты i для момента времени j . Этот штраф можно получить из классификатора.

Теперь требуется найти в матрице путь из клетки $(1, 1)$ в клетку (N_A, N_B) с наименьшим суммарным штрафом. Эту задачу можно решить за $N_A \times N_B$ операций при помощи динамического программирования.

Дана матрица $N_A \times N_B$ штрафов. Нужно найти путь из клетки $(1, 1)$ в клетку (N_A, N_B) с наименьшим суммарным штрафом. На пути из клетки можно перемещаться в ее соседа справа или справа-снизу.

Заведем матрицу D размера $N_A \times N_B$. D_{ij} равно минимальному штрафу, за который можно проложить путь из $(1, 1)$ в (i, j) . Будем заполнять эту матрицу по столбцам.

База динамики

$$D_{11} = \theta_{11}, D_{1j} = +\infty$$

Шаг динамики

$$D_{ij} = \min(D_{i-1,j}, D_{i-1,j-1}) + \theta_{ij}$$

Рассмотрим $x \in \mathbb{R}^d$

$$\min(\mathbf{x}) = \min_{i=1,\dots,d} x_i$$

Пусть $\Omega(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ — сильно выпуклая функция

$$\min_{\Omega}(\mathbf{x}) = \min_{\mathbf{q} \in \Delta^d} \langle \mathbf{q}, \mathbf{x} \rangle + \Omega(\mathbf{q}).$$

Тогда $\min_{\Omega}(\mathbf{x})$ — гладкая функция:

$$\nabla_{\mathbf{x}} \min_{\Omega}(\mathbf{x}) = \arg \min_{\mathbf{q} \in \Delta^d} \langle \mathbf{q}, \mathbf{x} \rangle + \Omega(\mathbf{q})$$

Например:

$$\Omega(\mathbf{q}) = \gamma \sum_{i=1}^d q_i \log q_i, \quad \gamma > 0$$

$$\min_{\mathbf{q} \in \Delta^d} \Omega(\mathbf{x}) = \min_{\mathbf{q} \in \Delta^d} \sum_{i=1}^d q_i x_i + \gamma \sum_{i=1}^d q_i \log q_i$$

$$\hat{q}_i \propto \exp \left\{ -\frac{x_i}{\gamma} \right\}, \quad \hat{\mathbf{q}} = \text{softmax} \left(-\frac{\mathbf{x}}{\gamma} \right) = \text{softmax} \left(\frac{\mathbf{x}}{\gamma} \right)$$

Пусть $\theta \in \mathbb{R}^{N_A \times N_B}$.

- Алгоритм DTW:

- $\text{DTW}(\theta)$ — величина минимального пути;
- $Y(\theta) \in \{0, 1\}^{N_A \times N_B}$ — матрица выравнивания;

- Алгоритм DTW_Ω :

- $\text{DTW}_\Omega(\theta)$ — приближенная величина минимального пути;
- $Y_\Omega(\theta) = \nabla_\theta \text{DTW}_\Omega(\theta)$ — smoothed матрица выравнивания;

Оптимизируемый функционал:

$$\text{MAD}(\hat{Y}, Y) = \|L(Y - \hat{Y})^T\|_F^2,$$

где $L \in \mathbb{R}^{N_B \times N_B}$ — нижнетреугольная матрица, заполненная 1.

Compute $\text{DTW}_\Omega(\boldsymbol{\theta})$ and $\nabla \text{DTW}_\Omega(\boldsymbol{\theta})$

Input: Distance matrix $\boldsymbol{\theta} \in \mathbb{R}^{N_A \times N_B}$

▷ Forward pass

$v_{0,0} = 0; v_{i,0} = v_{0,j} = \infty, i \in [N_A], j \in [N_B]$

for $i \in [1, \dots, N_A], j \in [1, \dots, N_B]$ **do**

$v_{i,j} = d_{i,j} + \min_\Omega(v_{i,j-1}, v_{i-1,j-1}, v_{i-1,j})$

$\mathbf{q}_{i,j} = \nabla \min_\Omega(v_{i,j-1}, v_{i-1,j-1}, v_{i-1,j}) \in \mathbb{R}^3$

▷ Backward pass

$\mathbf{q}_{i,N_B+1} = \mathbf{q}_{N_A+1,j} = \mathbf{0}_3, i \in [N_A], j \in [N_B]$

$e_{i,N_B+1} = e_{N_A+1,j} = 0, i \in [N_A], j \in [N_B]$

$\mathbf{q}_{N_A+1,N_B+1} = (0, 1, 0); e_{N_A+1,N_B+1} = 1$

for $j \in [N_B, \dots, 1], i \in [N_A, \dots, 1]$ **do**

$e_{i,j} = \mathbf{q}_{i,j+1,1} e_{i,j+1} + \mathbf{q}_{i+1,j+1,2} e_{i+1,j+1} +$
 $\mathbf{q}_{i+1,j,3} e_{i+1,j}$

Return: $\text{DTW}_\Omega(\boldsymbol{\theta}) = v_{N_A,N_B}$

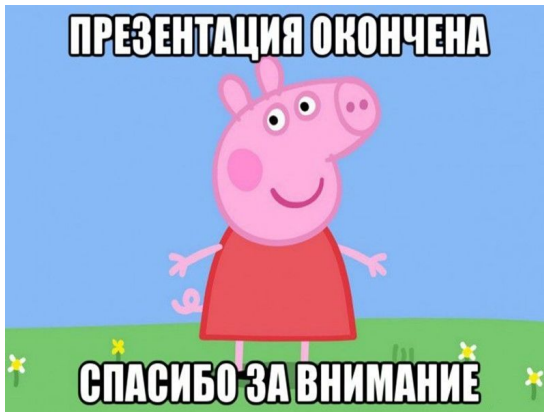
$\nabla \text{DTW}_\Omega(\boldsymbol{\theta}) = (e)_{i,j=1}^{N_A,N_B}$

Выборка была разбита на тренировочную и валидационную, по пять записей в каждой из частей.

Были рассмотрены три модели:

- 1 Простая модель с логистической регрессией в качестве базового классификатора.
- 2 End-to-end модель на основе DTW_{Ω} .

Инструмент	Pretrained	End-to-end
Скрипка	0.7804	0.4761
Кларнет	0.4527	0.2353
Саксофон	0.4930	0.4677
Фагот	0.6001	0.5650



Состав команды:

Тимур Гарипов, 517 группа
Татьяна Шолохова, 517 группа
Павел Коваленко, 517 группа
Саня Щербаков, 522 группа