

A. B. Kaygorodov

Применение математических моделей для распознавания динамических процессов в условиях ограниченных вычислительных ресурсов

A. V. Kaygorodov

Mathematical Models for Dynamic Processes Recognition with Limited Computational Resources

Рассматриваются основные проблемы распознавания динамических процессов, а также анализируются и адаптируются методы распознавания для реализации в условиях ограниченных вычислительных ресурсов. Исследуемые математические модели (метод опорных векторов и скрытая марковская модель) были адаптированы для непосредственного применения в области распознавания жестов с помощью устройств с акселерометром.

Ключевые слова: распознавание образов, жесты, машинное обучение, метод опорных векторов, марковские модели, оптимизация.

Введение. В ходе развития истории предлагались прототипы, начиная от языка интерпретации знаков (*sign language interpretation*) [1] и заканчивая общими принципами распознавания жестов. Использование Wii Remote и Nunchuk в игровой консоли от Nintendo позволило сделать консоль очень популярной, особенно в области спортивных игр.

Проведенный анализ популярных математических моделей распознавания динамических жестов: скрытых марковских моделей [2], нейронных сетей [3], байесовых сетей [4], показал, что оценки вычислительной сложности распознавания жестов с помощью этих моделей зависят от квадрата числа состояний (СММ), нейронов (нейросети) или вершин (байесовы сети), используемых для распознавания, умноженных на число символов наблюдаемой последовательности. Вследствие этого с ростом данных величин практическое использование для распознавания жестов в реальном времени указанных моделей из-за высоких затрат процессорного времени и памяти компьютера становится невозможным.

Основные этапы распознавания жестов

Сегментация. Большинство систем распознавания жестов используют сегментацию, основанную на нажатии кнопки (*button-based segmentation*), которая означает, что пользователь должен указать начало и конец жеста нажатием кнопки. Однако более естественным всё же будет использование систем сегментации без нажатия кнопки [5]. Роль сегментации заключается в определении части этого потока данных, где начинает происходить жест. Для того чтобы решить эту проблему, необходимо определить

This article touches main problems of dynamic processes recognition, and also analyzes and adapts some of recognition methods for using in limited computing resources. Investigated mathematic models (support vector machine and hidden Markov model) were adapted for applying in gesture recognition by devices with accelerometer.

Key words: pattern recognition, gestures, machine learning, support vector machine, hidden Markov model, optimization.

понятие жеста для нашей системы. В качестве примера может послужить следующее описание: жест начинается с быстрым ускорением, происходит непрерывное изменение направлений, заканчивается почти в устойчивом положении и длится более чем 0,8 секунды.

Для захвата жестов, в соответствии с определением, необходимо сделать некоторые предварительные обработки на трехмерном потоке данных. Прежде всего определение включает в себя заявления об изменении ускорения, поэтому необходимо считать производную по времени. Во-вторых, нужно вычислить размер вектора, потому что система должна быть инвариантной относительно порядка осей.

$$H_k = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2}. \quad (1)$$

Это значение может продемонстрировать изменение в ускорении и направлении тоже, но значение этой переменной может слишком быстро измениться на большое значение. Значит, необходимо использовать значение скользящей средней. Предлагается применение экспоненциальной скользящей средней как наиболее быстрой и менее ресурсоемкой.

$$EMA_{Hk} = \alpha H_k + (1 - \alpha) EMA_{Hk-1}. \quad (2)$$

Выход сегментации является переменной, но вектор данных — трехмерный, значит, имеем матрицу размером $3 \times k$. В режиме обучения она сохраняется в файл в системе с соответствующей пометкой. В режиме распознавания требуется решить, похожа ли матрица на один из жестов. Здесь есть два метода, которые позволяют это проверить.

Скрытая марковская модель. Скрытая марковская модель (СММ) — это инструмент статистического моделирования, который может быть применен для моделирования временных рядов с пространственной и временной изменчивостью. Как было упомянуто ранее, она часто используется для распознавания жестов, но также имеет многочисленные приложения в распознавании речи [5].

Скрытые модели Маркова являются статистическим методом, характеризующим наблюдаемые образцы дискретных данных. Однако прежде чем использовать этот метод, необходимо решить три основные проблемы: проблема оценки, проблема декодирования и задача обучения. Марковская цепь задается начальным распределением вероятностей $\rho^0(x)$ и вероятностями перехода $T(x';x)$. $T(x';x)$ — это распределение следующего элемента в цепи в зависимости от предыдущего; распределение на $(t+1)$ -м шаге равно

$$\rho^{t+1}(x') = \int T(x';x)\rho^t(x)dx. \quad (3)$$

В дискретном случае $T(x';x)$ — это матрица вероятностей переходов между состояниями $p(x'=i|x=j)$ [6].

Скрытые модели Маркова представляют собой набор состояний, связанный переходами [5]. Каждое состояние характеризуется двумя наборами вероятностей: вероятностью перехода и дискретным распределением вероятности выхода или функция плотности вероятности, которая, учитывая состояние, определяет условную вероятность появления входного символа из конечного алфавита или непрерывного случайного вектора. Формально СММ определяется:

$O = \{o_1, o_2, \dots, o_M\}$ — алфавит на выходе. Наблюдаемые символы соответствуют физическим выходным данным моделируемой системы, в нашем случае трехмерный вектор со значением ускорения.

$\Omega = \{1, 2, \dots, N\}$ — множество состояний, представляющих последовательность. Эти состояния являются скрытыми и для многих практических приложений, которые часто являются некоторыми физическими значениями, связанными с состоянием модели;

$A = \{a_{ij}\}$ — матрица вероятностей перехода, где a_{ij} — вероятность перехода из состояния i в состояние j , т. е.:

$$a_{ij} = P(S_{t+1} = j | S_t = i), 1 \leq i, j \leq N, 1 \leq t \leq T-1. \quad (4)$$

$B = \{b_j(k)\}$ — матрица вероятностей выхода, где $b_j(k)$ — вероятность появления символа O_k , в состоянии j [6]:

$$b_j(k) = P(X_t = o_k | S_t = i), 1 \leq j \leq N, 1 \leq t \leq T, 1 \leq k \leq M. \quad (5)$$

$\pi = \{\pi_i\}$ — распределение начальных состояний, где

$$\pi_i = P(S_1 = i), 1 \leq i \leq N. \quad (6)$$

Так как есть величины-вероятности, то они должны отвечать следующим требованиям:

$$\sum_{j=1}^N a_{ij} = 1, 0 \quad \sum_{k=1}^M b_j(k) = 1, 0 \quad \sum_{i=1}^N \pi_i = 1, 0. \quad (7)$$

В дальнейшем будет использоваться запись вида $\Phi = (A, B, \pi)$.

Кроме того, следует отметить разницу между дискретными СММ и непрерывными в функциях вероятностей выхода. Первым делом требуется переопределить функцию плотности $b_j(x)$, которая теперь становится классификатором [2]. Используя гауссиан, получим:

$$b_j(x) = \sum_{k=1}^M c_{jk} N(x, \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^M c_{jk} b_{jk}(x), \quad (8)$$

где x — моделируемый вектор; μ_{jk} — вектор математического ожидания, вектор ковариации Σ_{jk} .

На практике модели зависят от источника сигнала. У акселерометра есть три оси, и данные посылаются, соответственно, тройкой вида

$$x = [x_t, y_t, z_t]. \quad (9)$$

Вектор математического ожидания

$$\mu = [\mu_x, \mu_y, \mu_z]. \quad (10)$$

Матрица ковариации

$$\Sigma = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yz}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 & \sigma_{zz}^2 \end{bmatrix} \quad (11)$$

Определитель матрицы ковариации:

$$\begin{aligned} |\Sigma| &= \sigma_{xx}^2 \sigma_{yy}^2 \sigma_{zz}^2 + \sigma_{xy}^2 \sigma_{yz}^2 \sigma_{zx}^2 + \sigma_{xz}^2 \sigma_{yx}^2 \sigma_{zy}^2 - \\ &- \sigma_{xz}^2 \sigma_{yy}^2 \sigma_{zx}^2 - \sigma_{yx}^2 \sigma_{xy}^2 \sigma_{zz}^2 - \sigma_{xx}^2 \sigma_{zy}^2 \sigma_{yz}^2. \end{aligned} \quad (12)$$

Формула обратной матрицы ковариации:

$$\begin{aligned} \Sigma^{-1} &= \frac{1}{|\Sigma|} \begin{bmatrix} \left| \begin{smallmatrix} \sigma_{yy}^2 & \sigma_{yz}^2 \\ \sigma_{zy}^2 & \sigma_{zz}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xz}^2 & \sigma_{xy}^2 \\ \sigma_{zz}^2 & \sigma_{zy}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{yy}^2 & \sigma_{yz}^2 \end{smallmatrix} \right| \\ \left| \begin{smallmatrix} \sigma_{yz}^2 & \sigma_{yx}^2 \\ \sigma_{zz}^2 & \sigma_{zx}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xx}^2 & \sigma_{xz}^2 \\ \sigma_{zx}^2 & \sigma_{zz}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xz}^2 & \sigma_{xx}^2 \\ \sigma_{yz}^2 & \sigma_{yx}^2 \end{smallmatrix} \right| \\ \left| \begin{smallmatrix} \sigma_{yx}^2 & \sigma_{yy}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xy}^2 & \sigma_{xx}^2 \\ \sigma_{zy}^2 & \sigma_{zx}^2 \end{smallmatrix} \right| & \left| \begin{smallmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{smallmatrix} \right| \end{bmatrix} \\ &= \frac{1}{|\Sigma|} \begin{bmatrix} \sigma_{yy}^2 \sigma_{zz}^2 - \sigma_{yz}^2 \sigma_{zy}^2 & \sigma_{xz}^2 \sigma_{zy}^2 - \sigma_{xy}^2 \sigma_{zz}^2 & \sigma_{xy}^2 \sigma_{yz}^2 - \sigma_{xz}^2 \sigma_{yy}^2 \\ \sigma_{yz}^2 \sigma_{zx}^2 - \sigma_{yy}^2 \sigma_{zz}^2 & \sigma_{xx}^2 \sigma_{zz}^2 - \sigma_{xz}^2 \sigma_{zx}^2 & \sigma_{xz}^2 \sigma_{yx}^2 - \sigma_{xx}^2 \sigma_{yz}^2 \\ \sigma_{yx}^2 \sigma_{zy}^2 - \sigma_{yy}^2 \sigma_{zx}^2 & \sigma_{xy}^2 \sigma_{zx}^2 - \sigma_{xx}^2 \sigma_{zy}^2 & \sigma_{xx}^2 \sigma_{yy}^2 - \sigma_{xy}^2 \sigma_{yx}^2 \end{bmatrix} \end{aligned} \quad (13)$$

Проблема оценки. Проблема оценки в СММ заключается в следующем: дана наблюдаемая последовательность $x = x_1 x_2 \dots x_T$ и модель $\Phi = (A, B, \pi)$. Как эффективно вычислить $P(x|\Phi)$ — вероятность появления наблюдаемой последовательности?

Предположим, что значение переменной $\alpha_t(i)$ определено следующим образом:

$$\alpha_t(i) = P(x_1 x_2 \dots x_t, S_t = i | \Phi). \quad (14)$$

т. е. вероятность наблюдения части последовательности $x_1 x_2 \dots x_t$, начиная с состояния i , в момент времени t , для модели Φ . Можно найти $\alpha_t(i)$ методом математической индукции:

$$\alpha_t(i) = \pi_i b_i(x_1), \quad 1 \leq i \leq N. \quad (15)$$

Индуктивно:

$$\alpha_{t+1} = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(x_{t+1}), \quad 1 \leq j \leq N, \quad 1 \leq t \leq T-1. \quad (16)$$

Решение:

$$P(x|\Phi) = \sum_{i=1}^N \alpha_T(i). \quad (17)$$

В общем-то в этой ситуации достаточно рассмотреть процедуру «вперед» для решения данной проблемы. Процедура «назад» будет частично применяться в разрешении проблемы обучения [6].

Аналогично написанному выше получим, что

$$\beta_t(i) = P(x_{t+1} x_{t+2} \dots x_T | S_t = i, \Phi), \quad (18)$$

т. е. вероятность наблюдения части последовательности $x_{t+1} x_{t+2} \dots x_T$, начиная с состояния i , в момент времени t , для модели Φ . Можно найти аналогично методом математической индукции:

$$\beta_T = 1, 0, \quad 1 \leq i \leq N. \quad (19)$$

Индуктивно:

$$\beta_t(i) = \sum_{j=1}^N \alpha_{ij} b_j(x_{t+1}) \beta_{t+1}(j), \quad T-1 \geq t \geq 1, \quad 1 \leq i \leq N. \quad (20)$$

Проблема декодирования. Дана наблюдаемая последовательность $x = x_1 x_2 \dots x_T$ и модель $\Phi = (A, B, \pi)$, как выбрать надлежащую последовательность состояний $Q = S_1 S_2 \dots S_T$, которая лучшим образом объясняет наблюдения?

Для того чтобы найти лучшую последовательность состояний Q для наблюдаемой последовательности x , требуется определить величину [5]:

$$\delta_t(i) = \frac{\max_{S_1 S_2 \dots S_{t-1}} P(S_1 S_2 \dots S_t = i, x_1 x_2 \dots x_t | \Phi), \quad (21)$$

т. е. фактически наибольшую вероятность на протяжении всего наблюдения, начинающегося в момент t и заканчивающегося состоянием i . Для получения последовательности состояний следует отслеживать аргумент, который максимален для предыдущего значения величины δ . Для этого потребуется переменная $\psi_t(i)$:

$$\delta_1(i) = \pi_i b_i(x_1), \quad \psi_1(i) = 0, 1 \leq i \leq N. \quad (22)$$

Индуктивно:

$$\delta_t(i) = \frac{\max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ij}] b_j(x_t); \quad (23)$$

$$\psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N.$$

Решение:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)], \quad S_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (24)$$

Откуда следует, что последовательность:

$$S_t^* = \psi_{t+1}(S_{t+1}^*), \quad T-1 \geq t \geq 1. \quad (25)$$

Проблема обучения. Пожалуй, самая сложная проблема СММ заключена в определении метода для подстройки параметров модели (A, B, π) и увеличения сходства вероятности наблюдения последовательности с моделью. Так как можно наблюдать только конкретные последовательности, нет единого способа определения параметров модели. Однако можно выбрать $\Phi = (A, B, \pi)$, чтобы локально максимизировать $P(x|\Phi)$, используя итеративные методы [5, 6]. Одним из них является алгоритм Баума-Велша.

Прежде чем начать описывать процесс переопределения (итеративного обновления и улучшения) параметров СММ, следует объявить $\xi_t(i, j)$ — вероятность нахождения в состоянии i в момент времени t и состоянии j , в следующий момент времени $t+1$ для исходной и наблюдаемой модели:

$$\xi_t(i, j) = P(S_t = i, S_{t+1} = j | x, \Phi). \quad (26)$$

Можно записать, как:

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_t(i) \alpha_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{P(x|\Phi)} = \\ &= \frac{\alpha_t(i) \alpha_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \alpha_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}. \end{aligned} \quad (27)$$

Также следует объявить $y_t(i)$ — вероятность нахождения в состоянии i в момент времени t наблюдаемой последовательности и модели. Для этого достаточно суммировать $\xi_t(i, j)$ по j :

$$y_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad (28)$$

Анализируя (28), можно выявить вспомогательные свойства:

$\sum_{i=1}^{T-1} y_t(i)$ — ожидаемое количество переходов из состояния i ;

$\sum_{i=1}^{T-1} \xi_t(i, j)$ — ожидаемое количество переходов из состояния i в состояние j .

Зная смысл этих значений, можно вывести формулы переопределения параметров СММ:

$$\hat{\pi} = y_1(i). \quad (29)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} y_t(i, j)}. \quad (30)$$

$$\hat{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j) \text{ т.к. } x_t = o_k}{\sum_{t=1}^T \gamma_t(j)}. \quad (31)$$

Непрерывно:

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j) \beta t(j)}{\sum_{j=1}^N \alpha_t(j) \beta t(j)} \right] \left[\frac{c_{jk} N(x_t, \mu_{jk}, \sum_{jk})}{\sum_{m=1}^M c_{jm} N(x_t, \mu_{jm}, \sum_{jm})} \right]. \quad (32)$$

$$\hat{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)}. \quad (33)$$

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) x_t}{\sum_{t=1}^T \gamma_t(j, k)}. \quad (34)$$

$$\hat{\sum}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) (x_t - \mu_{jk})(x_t - \mu_{jk})^t}{\sum_{t=1}^T \gamma_t(j, k)}. \quad (35)$$

Метод опорных векторов. Основная идея метода опорных векторов (далее — МОВ) — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей наши классы [7, 8]. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора. Чтобы использовать МОВ, необходимо преобразовать матрицы вида $3 \times k$ в вектор фиксированной длины. Эти статистические данные описывают направление и динамику жеста. Для того чтобы данные по всем измерениям были равны, требуется применить z -нормализацию.

Следует отметить, что МОВ используется для решения задач бинарной классификации, цель которых заключена в поиске гиперплоскости, которая бы максимально разделила два класса выборки в пространстве признаков.

Предположим, что есть два жеста $G1, G2$, которые требуется классифицировать. Заметим, что обучающая выборка с количеством элементов n выглядит [7]:

$$\{(\tau_i, g_i)\}, i = 1, \dots, n, \quad (37)$$

где $\tau_i \in R^d$ представляет вектор состояний жестов (39) и [6]

$$g_i = \begin{cases} +1, & \tau_i \text{ является } G1 \\ -1, & \tau_i \text{ является } G2 \end{cases}$$

Возникает проблема оптимизации нескольких элементов:

$$W(\alpha) = \frac{1}{2} \sum_{i,j=1}^n g_i g_j \alpha_i \alpha_j K(\tau_i, \tau_j). \quad (39)$$

При: $\sum_{i=1}^n g_i \alpha_i = 0, \quad (40)$

$$0 \leq \alpha \leq C, i = 1, \dots, n, \quad (41)$$

где α_i — множитель Лагранжа; $K(\tau_i, \tau_j)$ — функция-ядро.

Тогда классифицирующая функция будет иметь вид:

$$f(x) = \text{sign}(h(x)) \quad (42)$$

$$h(x) = \sum_{i=1}^n \alpha_i^* g_i K(\tau_i, \tau_j) + b^*, \quad (42)$$

где α_i^*, b^* — оптимальное решение.

Однако распознавание жестов подразумевает распознавание более двух жестов. Для этого требуется реализация многоклассового МОВ. Допустимым решением здесь будет приведение проблемы многоклассового распознавания к ряду проблем бинарного распознавания. Здесь применимы два метода для построения многоклассового классификатора: один-против-одного (one-versus-one) и один-против-всех (one-versus-all) [8].

Выводы. Оба метода не являются ресурсоемкими во время распознавания и способны работать в условиях ограниченной рациональности, тем самым потребляя незначительные вычислительные способности устройства, на котором они будут использоваться. Для обучения данных алгоритмов не требуется применения специфических функций, реализация которых требует определенных конкретных требований, а значит, они могут быть реализованы кросс-платформенно. Достаточно сложное математическое обоснование алгоритмов скрытой марковской модели в рамках данной статьи было сведено к вычислению нескольких коэффициентов, а в методе опорных векторов выведено, что его суть сводится в оптимально подобранном ядре функции-классификатора. Тем самым данные методы не требуют больших вычислительных мощностей для непосредственной работы алгоритмов.

Библиографический список

1. Vogler C., Metaxas D. A framework for recognizing the si-multaneous aspects of American sign language // Computer Vision and Image Understanding. — 2001. — Vol. 81, № 3.
2. Pylv A. N., Ainen T. Accelerometer based gesture recognition using continuous HMMs // Lecture Notes in Computer Science (IbPRIA 2005 Proceedings). — 2005.
3. Tuulari E., Ylisaukko-Oja A. SoapBox: A platform for ubiquitous computing research and applications // Lecture Notes in Computer Science. — 2002. — Vol. 2414.
4. Rett J., Dias J. Gesture Recognition Using a Marionette Model and Dynamic Bayesian Networks // Lecture notes in computer science. — 2006. — Vol. 4142.
5. Hoffman F.G., Heyer P., Hommel G. Velocity profile based recognition of dynamic gestures with discrete hidden Markov models // Gesture and Sign Language in Human-Computer Interaction. — 1997.
6. Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi and Shijian Li. Gesture Recognition with a 3-D Accelerometer. — Springer-Verla; Berlin Heidelberg, UIC 2009. — LNCS 5585.
7. Bartlett P., Shawe-Taylor J. Generalization performance of support vector machines and other pattern classifiers // Advances in Kernel Methods. — Cambridge, 1998.
8. Burges C.J.C. Geometry and invariance in kernel based methods // Advances in Kernel Methods / Ed. by B. Scholkopf, C. C. Burges, A. J. Smola. — MIT Press, 1999.
9. Alon J., Athistos V., Yuan Q. and Sclaroff S. Simultaneous Localization and Recognition of Dynamic Hand Gestures // Proceedings of WACV MOTION'05. — Breckenridge, 2005. — Vol. 2.
10. Ong C. W., Ranganath S., Venkatesh Y. V. Understanding gestures with systematic variations in movement dynamics // Pattern recognition. — 2006. — Vol. 39, № 9.