О некоторых математических методах распознавания речи

Ермилов А.В.

В данной статье приводится описание некоторых математических методов, которые применяются при распознавании речи и идентификации диктора. Дается детальное описание скрытых марковских моделей (СММ) и метода опорных векторов. Приведены основные особенности применения скрытых марковских моделей для описания динамики речевого сигнала, а также метода опорных векторов для идентификации диктора.

• распознавание речи • идентификация диктора • скрытые марковские модели • SVM.

In this article presented several mathematical methods, which are used in speech recognition and speaker identification. Given thorough description of Hidden Markovs Models and Support Vector Machines. Application of Hidden Markov Models to the description of speech signal's dynamics are provided as well as application of Support Vector Machines to speaker identification.

• speech recognition • speaker identification • HMM • SVM

1. ВВЕДЕНИЕ

В современном мире все большее значение уделяется интерфейсам, использующим речевой ввод и вывод для взаимодействия между пользователем и компьютером. Поэтому всё большее многообразие в голосовых сообщениях приходиться принимать во внимание разработчику систем распознавания речи, реализующих акустический интерфейс.

Задача распознавания речи (во многих своих проявлениях: от транскрибирования слитной речи до верификации и идентификации диктора) в настоящее время является крайне актуальной. Свидетельством этому служит растущее число публикаций и конференций по данной тематике (таких как ICASSP, INTERSPEECH), а также то, что в крупнейших транснациональных корпорациях (таких как Microsoft, Google, IBM) открываются департаменты, ориентированные на исследования по данной тематике.

Исследовательские усилия в сфере речевых технологий привели к появлению большого числа коммерческих систем распознавания речи. Такие компании, как Nuance, IBM, ScanSoft, предлагают большой набор программных решений как для серверных, так и для десктопных приложений.

Улучшение существующих систем распознавания речи позволило существенно упростить взаимодействие человека с компьютером в том случае, когда использование классических интерфейсов невозможно (например, при управлении автомобилем или в сложных условиях, таких как ликвидация последствий чрезвычайных ситуаций) или затруднено (например, людям,

обладающим слабым зрением, или с ограниченными физическими возможностями), а также сделать работу с компьютером или иной техникой более комфортной. Также следует отметить, что применение систем распознавания речи весьма велико в работе правоохранительных служб (например, при идентификации говорящего или в системе защиты свидетелей).

В данной статье дается попытка описать некоторые математические методы, применяемые как для распознавания (транскрибирования) речи, так и для идентификации говорящего. В статье приводятся основные результаты, связанные с применением скрытых марковских моделей для траскрибирования речевого сигнала, а также особенности применения метода опорных векторов для идентификации диктора.

Следует отметить, что в статье не приводятся методы предварительной обработки сигнал (усиление, нарезка на фреймы и т.д.), способы выделения признаков речевого сигнала (такие как методы вычисления кепстральных коэффициентов) и другие темы, касающиеся цифровой обработки сигналов. Кроме того, остается за кадром применение таких современных моделей, как Deep Neural Networks, которые дают значительное увеличение точности распознавания.

2. СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ

Процессы, протекающие в реальной жизни, обычно характеризуются наблюдениями, которые можно рассматривать как сигналы. Эти сигналы могут быть как дискретными (например, символы какого-либо алфавита), так и непрерывными (музыка, температура, речь). Сигналы могут быть стационарными (то есть их статистические свойства не меняются во времени) или нестационарными. Сигналы могут быть чистыми (например, приходящими строго от одного источника) или могут быть испорчены каким-либо иным источником сигнала (шумом) или искажениями при передаче, реверберациями и т.д.

Для теоретического описания системы строится модель прохождения сигнала. Существует несколько причин, из-за которых применение таких моделей представляется удобным:

1. Такая модель может использоваться для обработки сигнала с целью получения желаемого результата. Например, если пользователи заинтересованы в улучшении качества речевого сигнала, который был испорчен шумом и/или искажениями при передаче. В этом случае можно использовать модель прохождения сигналов для создания системы, которая уменьшит шум и искажения оптимальным образом.

2. Модель прохождения сигналов позволяет определить характеристики источника сигнала при отсутствии самого источника. Это свойство особенно важно, когда получение сигнала непосредственно из источника очень дорого, например, сопровождено с большими затратами денег или требует большого количества времени. В этом случае представляется возможным построить модель и с помощью симуляций выяснить свойства источника.

3. Модели прохождения сигналов хорошо работают на практике, а следовательно, позволяют эффективно создавать важные с практической точки зрения предсказательные, распознающие и идентифицирующие системы.

Существуют несколько способов выбора модели прохождения для описания характеристик данного сигнала. Выделяют два основных типа моделей: детерминистические и стохастические.

Детерминистические модели обычно используют некоторые известные свойства сигнала, например, представление сигнала синусоидальной волной или

суммой экспонент и т.д. В этом случае спецификация модели достаточно проста: необходимо лишь оценить параметры сигнала – амплитуду, частоту, фазу и т.д.

Стохастические модели пытаются описать только статистические свойства сигнала. Примерами подобных моделей могут служить Гауссовы процессы, Пуассоновские процессы, марковские процессы (в том числе и скрытые). В основе стохастических моделей лежит предположение о том, что сигнал может быть хорошо описан как параметрический случайный процесс и что его параметры могут быть оценены достаточно точно.

Скрытая марковская модель (HMM – Hidden Markov model) определяется как двойной случайный процесс. Лежащий в основе случайный процесс представляет собой однородную марковскую цепь с конечным числом состояний. Последовательность состояний не наблюдается и поэтому называется скрытой. Эта цепочка состояний влияет на другой случайный процесс, который и производит последовательность наблюдений. Скрытые марковский модели представляют собой важный класс моделей, которые успешно используются во многих отраслях знаний, например, при моделировании речи. Базовая теория по скрытым марковским моделям будет дана ниже.

Можно выделить следующие преимущества использования скрытых марковских моделей при использовании в задаче распознавания речи:

- HMM обладают простой математический структурой.
- Структура HMM позволяет моделировать сложную цепочки наблюдений.
- Параметры модели могут быть автоматически выбраны таким образом, чтобы описать имеющийся набор данных для обучения.

В системах распознавания речи скрытые марковские модели обычно применяются для представления фонем или целых слов. Каждое скрытое состояние представляет часть фонемы или слова. В каждый момент времени состояние, в котором находится система, может быть изменено в соответствии с набором переходных вероятностей, связанных с данным состоянием. Схематично это представлено на рис. 1.

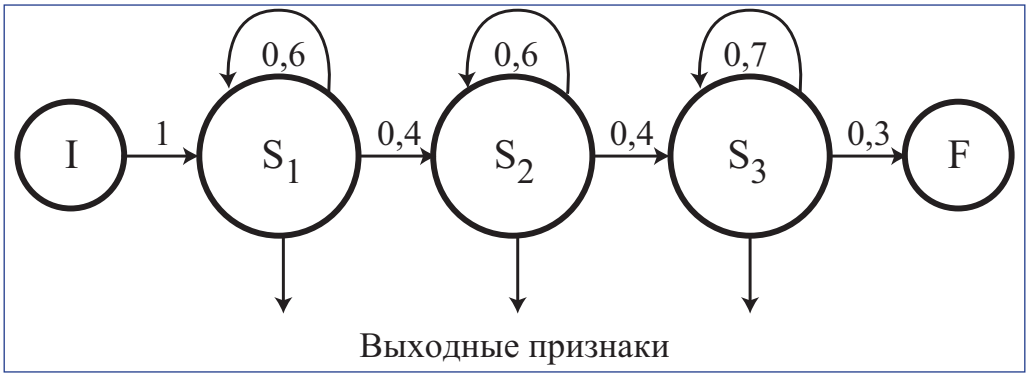

Рис. 1. Скрытая марковская модель с 5 состояниями. Символами I и F обозначены начальное и конечное состояния соответственно, $\{S_i\}_{i=1}^{3}$ – генерирующие состояния, дугами обозначены возможные переходы между состояниями, цифры над дугами обознают вероятности переходов между соответствующими состояниями.

Когда состояние активно, оно может генерировать последовательность векторов признаков, один вектор признаков в каждый момент времени. Эти вектора признаков имеют ту же форму, что и вектора признаков, которые подучаются, когда распознаётся сказанное слово. Однако невозможно узнать

точно последовательность состояний, пройденных системой для генерации данного набора наблюдаемых векторов признаков, так как каждое состояние дополнительно к переходным вероятностям определяется и плотностью распределения вероятности генерации векторов признаков. Она может быть использована для вычисления вероятности того, что вектор признаков был сгенерирован в данном состоянии. В качестве плотностей распределений обычно используются смеси гауссовых плотностей, каждая со своим средним, дисперсией и весом, например, как показано на рис. 2.

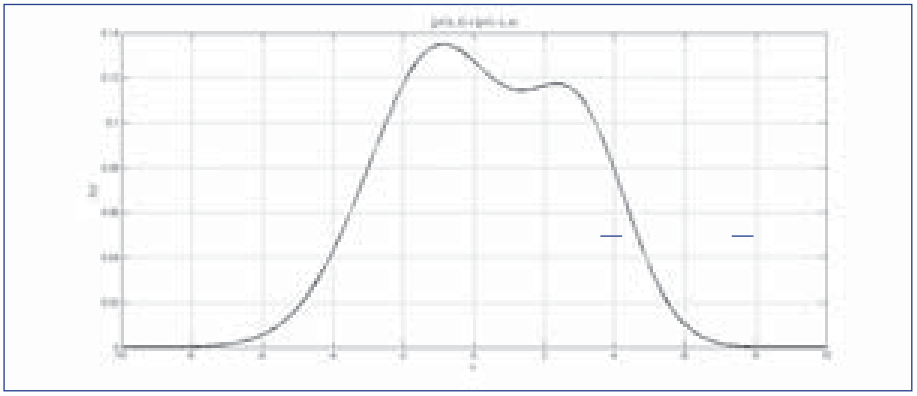

Рис. 2. Плотность смеси гауссовских распределений $\frac{3}{7}N(3,2)+\frac{4}{7}N(-1,4)$

Под обучением HMM понимают определение оценок параметров модели: переходных вероятностей, параметров плотности распределения и их весов. Эти параметры оптимизируются в соответствии с алгоритмом Баума-Уэлша [1]. Стоит отметить, что обычно для обучения требуется большой набор данных, при этом размер обучающего множества зависит от объёма словаря и параметров дикторов.

Распознавание производится посредством нахождения такой последовательности состояний, которая с наибольшей вероятностью сгенерировала последовательность векторов признаков. Такая последовательность находится с помощью алгоритма Витерби [2]. Зная последовательность состояний, можно просто определить соответствующую модельную последовательность – последовательность фонем или слов.

В связи с её использованием в работе, рассмотрим HMM более подробно.

2.1. Математическое описание скрытых марковских моделей

Пусть имеется марковская цепь в дискретном времени с набором состояний $S = 1, \ldots, M$. Через регулярные промежутки времени в системе происходит переход из одного состояния в другое (возможно, назад в предыдущее состояние). Последовательность состояний обозначим через $S_{1:T} = S_1, \ldots, S_M$, где $S_T \in S$ – состояние в момент времени t. Реализацию $S_{1:T}$ обозначим S_{1T}. Полное вероятностное описание системы требует задания текущего состояния в момент времени t и всех предшествующих состояний.

В частном случае дискретной марковской цепи первого порядка описание выглядит следующим образом:

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \ldots) = P(q_t = S_j \mid q_{t-1} = S_i). \quad (1)$$

В дальнейшем предполагается, что вероятности перехода не зависят от времени.

Обозначим $a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i)$, $1 \le i, j \le M$. При этом, $a_{ij} \ge 0, \Sigma_{j=1}^{M} a_{ij} = 1$.

Указанный случайны процесс может быть назван наблюдаемой марковской моделью, так как выходные значения процесса в каждый момент времени представляют собой состояния процесса. В случае если состояния процесса в каждый момент времени не наблюдаемы, то модель носит название скрытой марковской.

Скрытая марковская модель задаётся следующими элементами:

1. Количеством скрытых состояний N. Множество состояний модели обозначается $S = \{S_1, \dots, S_N\}$. Состояния соединенны таким образом, что любое состояние S_i может быть достигнуто из любого другого состояния S_j за конечное число шагов (эргодическая модель).
2. Размером выходного алфавита M. Набор символов выходного алфавита обозначается через $V = \{v_1, \dots, v_M\}$. Речевыми символами являются вектора из R^n.
3. Матрицей переходных вероятностей $A = (a_{ij})$, где

$$a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i), \quad i, j = 1, \dots M \tag{2}$$

4. Распределением вероятности выходных символов $B = \{b_j(k): j = 1, \dots, N, k = 1, \dots, M\}$ для данного состояния j, где k – порядковый номер символа v_k, а $b_j(k) = P(v \in V \mid q_t = S_j), j = 1, \dots, N, k = 1, \dots, M$, то есть, $b_j(k)$ – вероятность того, что в момент времени t система, находясь в состоянии S_j, выдаст символ v_k.
5. Вероятностью нахождения в состоянии i в начальный момент времени π_i, формирующие начальное распределение π.

Набор компонент A, B, Π, задающих марковскую модель, обозначается $\lambda = \{A, B, \Pi\}$. Последовательность наблюдений, сгенерированных марковской моделью за время T, обозначают $O = O_1, O_2, \dots, O_T$.

Для марковской модели первого порядка переходные вероятности зависят только от предыдущего состояния и не зависят от состояний в более ранние моменты времени.

Справедливо следующее утверждение.

Утверждение. Пусть скрытая марковская модель задаётся набором компонент $\lambda = \{A, D, \Pi\}$. Тогда для любого состояния S_k $P(q_{t+1} = S_k, \dots, q_{t+T-1} = S_k, q_{t+T} \ne S_k \mid q_t = S_k) = a_{kk}^T(1 - a_{kk})$, то есть, время нахождения цепи в состоянии S_k распределено экспоненциально.

2.2. Основные задачи, решаемые с помощью скрытых марковских моделей

Существуют три основные задачи, которые представляют интерес при решении практических задач.

1. Как при заданной последовательности символов наблюдений $O = O_1, O_2, \dots, O_T$ и модели $\lambda = \{A, B, \Pi\}$ вычислить вероятность наблюдения данной последовательности $P(O \mid \lambda)$ при условии, что она была сгенерирована моделью λ? Можно рассматривать эту проблему с точки зрения того, насколько хорошо данная модель соотносится с наблюдаемой последовательностью

наблюдений: при наличии нескольких моделей, решение этой задачи позволяет выбрать модель, которая лучше соответствует данным.

2. Как при заданной последовательности символов наблюдений $O = O_1, O_2, \ldots, O_T$ и модели $\lambda = \{A, B, \Pi\}$ вычислить соответствующую последовательность состояний $Q = q_1, q_2, \ldots, q_T$, оптимальную в некотором смысле? Очевидно, что кроме вырожденных случаев не существует единственно «правильной» последовательности состояний, поэтому следует использовать критерий оптимальности для выбора последовательности состояний.

3. Как вычислить оптимальные с точки зрения максимизации $P(O \mid \lambda)$ параметры $\lambda = \{A, B, \Pi\}$?

На практике широко используется следующее определение.

Определение. Последовательность наблюдений, используемая для оптимизации параметров HMM, называется обучающим множеством. Решение первой задачи позволит выбрать лучшую модель для объяснения имеющихся данных.

2.3. Алгоритмы решения основных задач, связанных с HMM

Решением первой задачи является метод, основанный на так называемом алгоритме прямого и обратного хода [3]. Опишем суть этого алгоритма.

Определение. Переменными прямого хода называются вероятность наблюдения частичной последовательности $O = O_1, O_2, \ldots, O_t$ и состояния S_i в момент времени t при заданной модели λ:

$$a_t(i) = P(O = O_1, O_2, \ldots, O_t\, q_t = S_i \mid \lambda) \tag{3}$$

Утверждение. Вероятность $P(O \mid \lambda)$ наблюдения последовательности $O = O_1, O_2, \ldots, O_T$ при условии, что она была сгенерирована моделью λ вычисляются через переменные прямого хода [4] как:

$$P(O \mid \lambda) = \Sigma_{i=1}^{N} a_T(i). \tag{4}$$

Доказательство. Алгоритм нахождения переменных прямого хода состоит из трёх последовательных шагов.

Инициализация:

$$a_1(i) = \pi b_1(O_1),\ 1 \le i \le N. \tag{5}$$

Индукция:

$$a_{t+1}(j) = b_j(O_{t+1})\, \Sigma_{i=1}^{N} a_t(i) a_{ij}. \tag{6}$$

Интерпретация этой формулы достаточно проста. Состояние S_j в момент времени $t + 1$ может быть достигнуто из N возможных состояний s_i, $1 \le i \le N$, в которых система могла находиться в момент t. Из определения $a_t(i)$ следует, что произведение $a_t(i)a_{ij}$ есть совместная вероятность того, что наблюдалась последовательность $O = O_1, O_2, \ldots, O_t$ и состояние S_j было достигнуто в момент времени $t + 1$ из состояния S_j. Суммируя эти вероятности по всем возможным состояниям, получаем вероятность того, что система находится в состоянии S_j и наблюдалась последовательность $O = O_1, O_2, \ldots, O_t$. Осталось принять во внимание, что в момент времени $t + 1$ будет наблюдаться O_{t+1} в состоянии S_j. Для этого необходимо умножить предыдущее на $b_j(O_{t+1})$.

Терминация

$$P(O \mid \lambda) = \Sigma_{i=1}^{N} a_T(i). \tag{7}$$

По определению $a_T(i) = P(O = O_1, O_2, \dots, q_T = S_i \mid \lambda)$, следовательно, для вычисления $P(O \mid \lambda)$ нужно лишь просуммировать все $a_T(i)$.

Аналогично можно определить переменные обратного хода.

Определение. Переменной обратного хода называется совместная вероятность наблюдения последовательности, начиная с момента $t + 1$ до конца, при заданном в момент t состоянии S_i и модели λ:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T \mid q_t = S_i, \lambda).$$

Утверждение. Переменные обратного хода выражаются рекурсивно по формуле:

$$\beta_t(i) = \Sigma_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \le i \le N \tag{8}$$

В отличие от решения первой задачи, при нахождении оптимальной последовательности символов необходимо уточнить критерий оптимальности. В качестве возможного критерия может выступать количество индивидуально наиболее вероятных состояний. Такой критерий обладает следующим недостатком. В случае, если некоторая переходная вероятность $a_{ij} = 0$, то найденная оптимальная последовательность состояний может не быть допустимой. Эта проблема возникает потому, что алгоритм определяет наиболее вероятное состояние в данный момент времени и не учитывает вероятности появления последовательностей символов.

Наиболее часто встречаемый критерий заключается в том, чтобы найти единственную лучшую последовательность наблюдений, то есть максимизации $P(q_1, \dots, q_t \mid O_1, \dots, O_T, \lambda)$, что в силу теоремы Байеса эквивалентно максимизации $P(q_1, \dots, q_T O_1, \dots, O_T \mid \lambda)$. Алгоритм, решающий указанную задачу, называется алгоритмом Витерби [2].

Для нахождения лучшей последовательности состояний $Q = q_1, \dots, q_T$ определим величину

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} P(q_1, \dots, q_t = i, O_1, \dots, O_t \mid q_t = S_i, \lambda). \tag{9}$$

Тогда для нахождения $\delta_{t+1}(j)$ нужно взять максимальную (то есть, наиболее вероятную) $\delta_t(i)$ с предыдущего шага и умножить на вероятность наблюдения символа O_{t+1} в состоянии S_j:

$$\delta_{t+1}(j) = b_j(O_{t+1}) \max_i \delta_t(i) a_{ij}. \tag{10}$$

Чтобы определить искомую последовательность символов, необходимо сохранять $\psi_t(i) = arg\ max \delta_t(i)$ для каждого i.

Теперь полная процедура нахождения оптимальной последовательности состояний (алгоритм Витерби) может быть записан следующим образом.

Инициализация:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N, \psi_1(i) = 0. \tag{11}$$

Рекурсия:

$$\delta_t(j) = b_j(O_{t+1}) \max_{1 \le i \le N} \delta_t(i)\, a_{ij}, \quad 1 \le i \le N, \tag{12}$$

$$\psi_t(i) = arg \max_{1 \le i \le N} \delta_t(i)\, a_{ij} \quad 2 \le t \le T. \tag{13}$$

Терминация:

$$\hat{P} = \max_{1 \le i \le N} \delta_T(j), \tag{14}$$

$$\hat{q}_T = arg\ max_{1 \le i \le N} \delta_t(j). \quad (15)$$

Определение последовательности состояний:

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad t = T-1, \ldots, 1 \quad (16)$$

Стоит отметить, что алгоритм Витерби очень похож на вычисление переменных прямого хода за исключением того, что вместо суммирования по всем предыдущим состояниям, происходит максимизация.

Решение третьей задачи представляется самым сложным. Сложность заключается в том, что нет аналитического решения максимизационной задачи по нахождению оптимальных параметров модели. Представляется возможным найти такие параметры модели $\lambda = \{A, B, \Pi\}$, которые дают локальный максимум $P(O \mid \lambda)$. Поиск локального максимума может быть осуществлен с помощью итеративного алгоритма Баума-Велша.

Обозначим вероятность нахождения в состоянии S_i в момент времени t и в состоянии S_j в момент $T + 1$ при данной модели и последовательности наблюдений через $\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$.

Из определения переменных прямого и обратного хода следует, что

$$\xi_t(i,j) = \frac{a_t(i) a_{ij} b_j (O_{t+1} B_{t+1}(j)}{P(O \mid \lambda)} \quad (17)$$

Обозначим через $\gamma_t(i)$ вероятность нахождения в состоянии S_i в момент времени t при заданной последовательности наблюдений и модели. Тогда $\gamma_t(i) = \Sigma_{j=1}^{N} \xi_t(i,j)$. Более того, используя $\gamma_t(i)$ можно подсчитать количество переходов из состояния S_i как $\Sigma_{t=1}^{T-1} \gamma_t(i)$. Кроме того, $\Sigma_{t=1}^{T-1} \xi_t(i,j)$ – ожидаемое количество переходов из состояния S_i в состояние S_j.

Запишем формулы, которые необходимо будет использовать при переоценке параметров модели A, B, Π:

$$\hat{\pi} = \gamma_1(j) \quad (18)$$

$$\hat{a}_{ij} = \frac{\Sigma_{j=1}^{N} \xi_t(i,j)}{\Sigma_{t=1}^{T-1} \gamma_t(i)} \quad (19)$$

$$\hat{b}_k = \frac{\Sigma_{t=1,\, O_t = v_k}^{T-1} \gamma_t(i)}{\Sigma_{t=1}^{T} \gamma_t(i)} \quad (20)$$

В работе Баума [1] было показано, что либо параметры начальной модели λ являются критическими для функции правдоподобия, либо существует другая модель $\hat{\lambda}$ с параметрами $\hat{A}, \hat{B}, \hat{\Pi}$, которая является более вероятной в том смысле, что $P(O \mid \hat{\lambda}) > P(O \mid \lambda)$. Таким образом, если итеративно использовать модель $\hat{\lambda}$ вместо λ и повторять переоценку параметров, то на каждом шаге увеличивается вероятность того, что данная последовательность наблюдений была получена из текущей модели. Будем повторять эту процедуру, пока не достигнем какой-либо точки остановки. Финальный результат процедуры переоценивания называется оценкой максимального правдоподобия.

3. МЕТОД ОПОРНЫХ ВЕКТОР

Задачу распознавания речи можно рассматривать не только с позиций транскрибирования (то есть представления речевого сигнала в виде текста), но и с позиций идентификации и верификаций говорящего.

Здесь задача идентификации диктора по звуковому сообщению является частным случаем задачи распознавания образов, для решения которой требуется построить статистический критерий принадлежности нового звукового сообщения к одному из классов, задаваемых «обучающими» сообщениями.

Задача идентификации решалась в следующей постановке. Пусть X – пространство объектов, Y – множество ответов, $f: X \rightarrow Y$ – целевая зависимость. Пусть $X^l \in X \times Y$ – обучающее множество, то есть множество пар (X_i, y_i), где $y_i = f(X_i)$. По известному обучающему множеству требуется построить $f: X \rightarrow Y$ аппроксимирующую f на всем X.

В настоящее время существуют два подхода к идентификации диктора: закрытый и открытый [4]. В первом случае классификации предполагается, что новое сообщение принадлежит одному из рассматриваемых дикторов, во втором – сообщение может относиться и к неизвестному диктору. Задача идентификации рассматривается как построение статистического критерия разделения полученных точек для конечного числа простых гипотез в случае закрытой задачи или для конечного числа простых и одной сложной гипотезы (сообщение неизвестного диктора) в случае открытой задачи.

3.1. Метод распознавания диктора, основанный на SVM

Метод опорных векторов (Support Vector Machines, SVM) [5] является базовым инструментом для распознавания образов на основе статистической теории обучения. SVM широко используется в естественно-языковых приложениях при обработке речевых сигналов. Например, задачи распознавания языка и идентификация диктора.

Идея SVM основана на следующих предпосылках. Предположим, что существуют два класса объектов в некотором n-мерном пространстве, которые можно разделить гиперплоскостью так, что с одной стороны от гиперплоскости должны находиться вектора одного класса, с другой – второго. Очевидно, что такая гиперплоскость может быть не единственной. Построим две такие параллельные гиперплоскости. Для лучшего разделения классов требуется, чтобы расстояние между плоскостями было как можно больше. Обычно для нахождения параллельных разделяющих гиперплоскостей с максимальным расстоянием в методе опорных векторов минимизируется квадратичная функция с линейными ограничениями. Решение такой задачи выражается через координаты обучающих векторов, лежащих на краю разделяющей полосы – так называемых опорных векторов. В случае, когда классы линейно неразделимы в исходном пространстве, строится отображение (необязательно линейное) в пространство большей размерности, образы классов в котором линейно разделимы. Это пространство называется пространством вторичных признаков.

3.2. Базовая модель SVM

Алгоритм SVM обладает следующими преимуществами:

1. В силу решения задачи минимизации выпуклой функции алгоритм гарантирует получение единственного решения. Это является серьёзным преимущество перед нейронными сетями, в которых решением может быть локальный минимум или ответ может быть неопределенным.

2. В связи с тем, что алгоритм робастен к зашумленности исходного сигнала, он хорошо приспособлен для распознания речи.

3. Алгоритм позволяет работать с данными очень больших размерностей, что важно при распознавании речи, где размерность вектора признаков может достигать многих сотен или тысяч.

Для формализации задачи обучения SVM, обозначим вектора признаков как $\{X_n\}_{n=1}^{N}$, а линейную функцию $(W, X) + b = 0$, где $(\cdot,\cdot)$ – скалярное произведение в R^k. Обозначим разделяемые классы через A и B и введем метки классов:

$$y_i = \mathrm{f(x)} = \begin{cases} 1, X_i \in A \\ -1, X_i \in \mathrm{B} \end{cases}$$

Будем искать $\hat{f}$ в виде $\hat{f}(X) = sign(W^T X + b))$, используя метод опорных векторов, разработанный В. Н. Вапником [5].

Утверждение. Максимизация ширины, разделяющей полосы, эквивалентна минимизации нормы W:

$$\begin{aligned} &½(W, W) \to \min_{W,\, b} \\ &y_i((W, X_i) + b) \geq 1, \quad i = 1, \ldots, N \end{aligned} \tag{21}$$

Утверждение. Решение задачи (21) выражается через вектора, для которых $y_i(((W, X_i) + b) - 1) = 0$, то есть, лежащих на разделяющей полосе.

Решение задачи (21) называется обучением классификатора.

3.3. Метод SVM с ядрами

В общем случае линейное разделение векторов может быть невозможно. Для решения задачи в этом случае можно преобразовать имеющиеся пространство таким образом, чтобы вектора классов после преобразования стали линейно разделимыми. Рассмотрим теперь, как будет ставиться и решаться задача в нелинейном случае.

Пусть ϕ произвольное отображение пространства признаков в гильбертово пространство H. От отображения требуется, чтобы образы обучающих векторов были линейно разделимы в H (оно называется пространством вторичных признаков). Тогда с подстановкой $\phi(X_i)$ вместо X_i получается классифицирующий алгоритм SVM [5]. Для его настройки и применения нужно знать не само отображение ϕ, а только функцию $K : X \times X \to R$, вычисляющую скалярное произведение в H образов пары векторов признаков $K(X_i, X_j) = (\phi(X_i), \phi(X_j))$.

Такая функция K называется ядром, поскольку при наличии меры, в частности при $H = R$, она является ядром интегрального оператора

$$f \to K(F) = \int_F K(\cdot, Y) f(Y)\, dY \tag{22}$$

Наиболее часто используются следующие ядра:

1. Линейное:

$$K(X, Y) = a(X, Y) + c \tag{23}$$

2. Полиномиальное:

$$K(X, Y) = ((X, Y) + 1)^d + \mathrm{c} \tag{24}$$

3. Гауссово:

$$K(X, Y) = e^{-\mathrm{y}\|\mathrm{x}-y\|} \tag{25}$$

Этих ядер обычно бывает достаточно для разделения любого набора векторов. Действительно, полиномиальное ядро переводит вектор X в набор всех мономов (то есть одночленов) степени не большей N от координат X, то

есть сводит разделимость к полиномиальной и гарантирует разделение не менее чем $N+1$ вектора.

Если же ϕ задано гауссовым ядром, то для любого конечного набора векторов X_1, ... , X_N функции $\phi(X_1)$, ... , $\phi(X_N)$ линейно независимы, что и обеспечивает линейную разделимость.

После преобразования построение оптимальной разделяющей полосы производится таким же способом, что и в случае (21), за исключением того, что все X_i заменяются на $\phi(X_i)$, а скалярные произведения (X_i, X_j) – на $K(X_i, X_j)$.

3.4. Метод SVM со штрафами

После обучения может оказаться так, что полученный классификатор не способен к обобщению. То есть он очень хорошо классифицирует обучающие вектора, но на произвольном тестовом наборе он показывает плохие результаты. Такой классификатор называется неспособным к обобщению (результатов обучения) [5]. Другое название – переобучение (overfitting). Переобучение происходит потому, что классификатор настраивается на шумы и помехи в данных.

Решение этой проблемы может быть следующим.

Вместо системы запретов вводится система штрафов за нарушение. В этом случае обучения классификатора сводится к решению следующей задачи:

$$\begin{gathered} \tfrac{1}{2}(W, W) + C\sum_{i=1}^{N} \rho(e_i) \to \min_{W,\, b} \\ y_i((W, X_i) + b) \geq 1 - e_i, \\ e_i \geq 0 \\ i = 1, \ldots, N \end{gathered} \tag{26}$$

где $\rho(e)$ – неотрицательная, монотонно неубывающая функция, такая, что $\rho(0) = 0$, а $C > 0$ – эмпирически подобранный коэффициент.

Идеальный штраф $\rho(e) = \theta(e - 1)$, где $\theta(t)$ – функция Хевисайда, при котором $\sum_{i=1}^{N} \rho(e_i)$ представляет собой количество неправильно классифицированных векторов, оказывается неудобен из-за своей разрывности. На практике применяется непрерывный штраф, иногда квадратичный, а чаще – линейный.

В постановке задачи (21) векторам запрещено находиться на той стороне от гиперплоскости, которая соответствует другому классу, разделяющая полоса носит называние «жесткой». В общем случае, задача решается в пространстве вторичных признаков, и вместо «жесткой» разделяющей полосы рассматривается штраф за нарушение ограничения. Использование пространства вторичных признаков помогает справиться с линейной неразделимостью, а использование штрафов – с возможным «переобучением» (overfitting). В результате преобразований исходной задачи, получаем задачу квадратичного программирования с линейными ограничениями:

$$\begin{gathered} \tfrac{1}{2}(W, W) + C\sum_{i=1}^{N} \rho(e_i) \to \min_{W,\, b} \\ y_i((W, \phi(X_i)) + b) \geq 1 - e_i, \\ e_i \geq 0 \\ i = 1, \ldots, N \end{gathered} \tag{27}$$

Коэффициент штрафа C подбирают так, чтобы и количество векторов, попавших на неправильную сторону разделяющей полосы, было небольшим, и общее количество опорных векторов тоже было невелико, поскольку, чем их меньше, тем быстрее работает классификатор.

Однако на практике приходиться решать задачу разделения на три и более классов. Для её решения можеть быть применён метод, называемый «каждый против каждого» (One vs. One [4]). Суть метода заключается в следующем. На этапе обучения рассматривается $\frac{q(+1)}{2}$ классификаторов SVM, различающих пары классов, где q – количество классов. Каждый из классификаторов обучается только на векторах, принадлежащих двум, соответствующим данному классификатору классам, поэтому время обучения и количество опорных векторов получаются меньше, чем у SVM типа «каждый против всех». Для каждого распознаваемого вектора рассчитаем все значения классифицирующих функций, отделяющих i-й класс от j-го, затем вычислим q сумм $f(\phi, X) = \Sigma_{i=j}\, \phi(f_{ij}(X))$, где ϕ – некоторая монотонно неубывающая функция, (например, $sign$) и выберем из них наибольшую. Соответствующий класс, для которого получена максимальная оценка, и будет ответом распознавателя.

Следует отметить, что метод имеет следующие особенности:

1. Вычислительная сложность обычных алгоритмов, решающих задачу квадратичного программирования (таких как метод Ньютона), делает задачу обучения SVM крайне трудоемкой для больших наборов данных.
2. В каждом конкретном случае решение задачи подбора ядра требует предварительного изучения.
3. Для определения величины параметра штрафа C также необходимо предварительное исследование.

ЗАКЛЮЧЕНИЕ

В данной статье проанализированы математические модели, применяющиеся для построения систем распознавания речи:

- Приведёно математическое описание скрытых марковских моделей.
- Рассмотрены особенности применения метода опорных векторов для идентификации диктора, а также дано математическое описание задачи обучения машины опорных векторов.

В настоящее время все большей популярностью при построении систем распознавания речи пользуются модификации описанных в статье алгоритмов с помошью глубоких нейронных сетей (Deep Belief Networks). Подобные модификации позволяют существенно увеличить точность распознавания систем, сохраняя при этом возможность относительно быстрого обучения на больших объемах данных. В качестве возможного расширения данной статьи представляется полезным дать описание подобного рода модификаций.

ЛИТЕРАТУРА

1 *Baum L., Petrie T.* Statistical inference for probabilistic functions of finite state Markov chains. The annals of mathematical statistics, 37(6):1554–1563, 1966.

2 *Viterbi A.* Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Information Theory, IEEE Transactions on, 13(2):260–269, 1967

3 *Аграновский А.В., Леднов Д.А.* Теоретические аспекты алгоритмов обработки и классификации речевых сигналов. М.: Радио и связь, 2004.

4 *Hsu C.-W., Lin C.J.* A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13(2):415–425, 2002.

5 *Cortes C.,Vapnik V.* Support-vector networks. Machine learning, 20(3):273–297, 1995.

Сведения об авторе:

Ермилов Алексей Валерьевич,

аспирант кафедры управления разработкой программного обеспечения Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ). Область научных интересов: машинное обучение, распознавание речи. Email: alvalerm@mail.ru