Скрытые марковские модели II

Сергей Николенко

Распознавание речи — ИТМО, осень 2008

Outline

Специальные виды моделей

- HMM *эргодична*, если $\forall i, j\ a_{ij} > 0$.
- HMM *ациклична* (left-right model, Bakis model), если $\forall j < i$ $a_{ij} = 0$ (матрица треугольная) и $\pi_1 = 1$ (начинаем всегда в состоянии 1).
- В ацикличных сетях ещё часто бывает ограничение на прыжок (constrained jump) размером Δ: $\forall j > i + \Delta\ a_{ij} = 0$ (матрица мультидиагональная).

Outline

Непрерывные плотности наблюдаемых

- У нас были дискретные наблюдаемые с вероятностями $B = (b_j(k))$.
- Но в реальной жизни всё сложнее: зачастую мы наблюдаем непрерывные сигналы, а не дискретные величины, и дискретизовать их или плохо, или неудобно.
- При этом саму цепь можно оставить дискретной, т.е. перейти к непрерывным $b_j(D)$.

Специальный вид плотности

- Не для всех плотностей найдены алгоритмы пересчёта (обобщения алгоритма Баума–Велха).
- Наиболее общий результат верен, когда $b_j(D)$ можно представить в виде

$$b_j(D) = \sum_{m=1}^{M} c_{jm} \mathcal{P}(D, \mu_{jm}, \sigma_{jm}),$$

где c_{jm} — коэффициенты смеси ($\sum_m c_{jm} = 1$), а $\mathcal{P}$ — выпуклое распределение со средним μ и вариацией σ (гауссиан подойдёт).
- К счастью, такой конструкцией можно приблизить любое непрерывное распределение, поэтому это можно широко применять.

Вспомогательные переменные

- $\gamma_t(j, m)$ — вероятность быть в состоянии j во время t, причём за D отвечает m–й компонент смеси.
- Формально говоря,

$$\gamma_t(j, m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right] \left[\frac{c_{jm}\mathcal{P}(d_t, \mu_{jm}, \sigma_{jm})}{\sum_{m=1}^{M} c_{jm}\mathcal{P}(d_t, \mu_{jm}, \sigma_{jm})}\right].$$

- Если $M = 1$, то это уже известные нам $\gamma_t(j)$.

Алгоритм для этого случая

- Нужно научиться пересчитывать $b_j(D)$, т.е. пересчитывать c_{jm}, μ_{jm} и σ_{jm}.
- Это делается так:

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(j,m)},$$

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \cdot d_t}{\sum_{t=1}^{T} \gamma_t(j,m)},$$

$$\bar{\sigma}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \cdot (d_t - \mu_{jm})(d_t - \mu_{jm})^t}{\sum_{t=1}^{T} \gamma_t(j,m)}.$$

Проблема

- Как моделировать продолжительность нахождения в том или ином состоянии?
- В дискретном случае вероятность пробыть в состоянии i d шагов:
$$p_i(d) = a_{ii}^{d-1}(1 - a_{ii}).$$
- Однако для большинства физических сигналов такое экспоненциальное распределение не соответствует действительности. Мы бы хотели явно задавать плотность пребывания в данном состоянии.
- Т.е. вместо коэффициентов перехода в себя a_{ii} — явное задание распределения $p_i(d)$.

Вспомогательные переменные

- Введём переменные

$$\alpha_t(i) = p(d_1 \dots d_t, x_i \text{ заканчивается во время } t|\lambda).$$

- Всего за первые t шагов посещено r состояний $q_1 \dots q_r$, и мы там оставались $d_1, \dots, d_r$. Т.е. ограничения:

$$q_r = x_i, \qquad \sum_{s=1}^{r} d_s = t.$$

Вычисление $\alpha_t(i)$

- Тогда получается

$$\alpha_t(i) = \sum_q \sum_d \pi_{q_1} p_{q_1}(d_1) p(d_1 d_2 \dots d_{d_1} | q_1)$$
$$a_{q_1 q_2} p_{q_2}(d_2) p(d_{d_1+1} \dots d_{d_1+d_2} | q_2) \dots$$
$$\dots a_{q_{r-1} q_r} p_{q_r}(d_r) p(d_{d_1+\dots+d_{r-1}+1} \dots d_t | q_r).$$

Вычисление $\alpha_t(i)$

- По индукции

$$\alpha_t(j) = \sum_{i=1}^{n} \sum_{d=1}^{D} \alpha_{t-d}(j) a_{ij} p_j(d) \prod_{s=t-d+1}^{t} b_j(d_s),$$

где D — максимальная остановка в любом из состояний.

- Тогда, как и раньше,

$$p(d|\lambda) = \sum_{i=1}^{n} \alpha_T(i).$$

Вспомогательные переменные

- Для пересчёта потребуются ещё три переменные:

$$\alpha_t^*(i) = p(d_1 \dots d_t, x_i \text{ начинается во время } t+1|\lambda),$$

$$\beta_t(i) = p(d_{t+1} \dots d_T|x_i \text{ заканчивается во время } t, \lambda),$$

$$\beta_t^*(i) = p(d_{t+1} \dots d_T|x_i \text{ начинается во время } t+1, \lambda).$$

Вспомогательные переменные

- Соотношения между ними:

$$\alpha_t^*(j) = \sum_{i=1}^{n} \alpha_t(i) a_{ij},$$

$$\alpha_t(i) = \sum_{d=1}^{D} \alpha_{t-d}^*(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(d_s),$$

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij} \beta_t^*(j),$$

$$\beta_t^*(i) = \sum_{d=1}^{D} \beta_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(d_s).$$

Формулы пересчёта

- Приведём формулы пересчёта.
- π_i — просто вероятность того, что x_i был первым состоянием:

$$\hat{\pi}_i = \frac{\pi_i \beta_0^*(i)}{p(d|\lambda)}.$$

- a_{ij} — та же формула, что обычно, только вместе с α есть ещё и β, которая говорит, что новое состояние начинается на следующем шаге:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)}{\sum_{k=1}^{n} \sum_{t=1}^{T} \alpha_t(i) a_{ik} \beta_t^*(k)}.$$

Формулы пересчёта

- $b_i(k)$ — отношение ожидания количества событий $d_t = v_k$ в состоянии x_i к ожиданию количества любого v_j в состоянии x_i:

$$\hat{b}_i(k) = \frac{\sum_{t=1, d_t=v_k}^{T} \left(\sum_{\tau<t} \alpha_\tau^*(i)\beta_\tau^*(i) - \sum_{\tau<t} \alpha_\tau(i)\beta_\tau(i) \right)}{\sum_{k=1}^{m} \sum_{t=1, d_t=v_k}^{T} \left(\sum_{\tau<t} \alpha_\tau^*(i)\beta_\tau^*(i) - \sum_{\tau<t} \alpha_\tau(i)\beta_\tau(i) \right)}.$$

- $p_i(d)$ — отношение ожидания количества раз, которые x_i случилось с продолжительностью d, к количеству раз, которые x_i вообще случалось:

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(d_s)}{\sum_{d=1}^{D} \sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(d_s)}.$$

За и против

- Такой подход очень полезен, когда $p_i(d)$ далеко от экспоненциального.
- Однако он сильно увеличивает вычислительную сложность (в D^2 раз).
- И, кроме того, становится гораздо больше параметров, т.е. нужно, вообще говоря, больше данных, чтобы эти параметры надёжно оценить.

Параметрическая продолжительность состояния

- Чтобы уменьшить количество параметров, можно иногда считать, что $p_i(d)$ — классическое распределение с не слишком большим количеством параметров.
- Например, $p_i(d)$ может быть равномерным, или нормальным $(p_i(d) = \mathcal{N}(d, \mu_i, \sigma_i^2))$, или гамма–распределением:

$$p_i(d) = \frac{\eta_i^{\nu_i} d^{\nu_i - 1} e^{-\eta_i d}}{\Gamma(\nu_i)}.$$

Outline

Общая идея

- Речь идёт о моделях, в которых следующие компоненты вектора наблюдаемых зависят от предыдущих с некоторыми коэффициентами.
- Т.е. $d = (d_1, \ldots, d_K)$ — вектор наблюдаемых, и модель такая:

$$d_k = -\sum_{i=1}^{p} a_i d_{k-i} + e_k,$$

где e_k — гауссианы с нулевым средним и вариацией σ^2, a_i — коэффициенты авторегрессии.

Распределение

- Можно показать, что плотность d для больших k стабилизируется на

$$p(d) = (2\pi\sigma^2)^{-K/2} e^{-\frac{1}{2\sigma^2}\delta(d,a)},$$

где (положив $a_0 = 1$)

$$\delta(d,a) = r_a(0)r(0) + 2\sum_{i=1}^{p} r_a(i)r(i),$$

$$r_a(i) = \sum_{j=0}^{p-i} a_j a_{j+i}, \quad r(i) = \sum_{j=0}^{K-i-1} d_j d_{j+i}.$$

Комментарий к распределению

- На самом деле $r(i)$ — автокорреляция выборки, а $r_a(i)$ — автокорреляция коэффициентов авторегрессии.
- Автокорреляция — это корреляция процесса с самим собой в предыдущие периоды времени; используется в статистике и обработке сигналов для поиска паттернов в данных.
- По определению, для процесса X_t $R(t,s) = \frac{E[(X_t-\mu)(X_s-\mu)]}{\sigma^2}$, а если процесс стационарный (не зависит от конкретного времени), то

$$R(k) = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2}.$$

- В обработке сигналов (а мы в ней) часто используют автокорреляцию без нормализации, т.е. для дискретного сигнала $R(j) = \sum_k x_j x_{j-k}$.

Скрытая марковская модель

- Как применить авторегрессивную модель?
- Очень просто: сначала нормализуем вектор наблюдений:

$$\hat{d} = \frac{d}{\sqrt{K\sigma^2}}, \quad p(\hat{d}) = \left(\frac{2\pi}{K}\right)^{-K/2} e^{-\frac{K}{2}\delta(\hat{d},a)}.$$

- А затем рассмотрим смесь

$$b_j(D) = \sum_{m=1}^{M} c_{jm} b_{jm}(D),$$

где b_{jm} задаётся вектором авторегрессии a_{jm} (т.е. его коэффициентами $r_{a_{jm}}$).

Алгоритм пересчёта

- Надо научиться пересчитывать $r_{a_{jm}}$. Для этого схема такая: сначала научимся пересчитывать r_{jm}, потом из них получим a_{jm}, а потом по ним пересчитаем $r_{a_{jm}}$.
- $$\bar{r}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \cdot r_t}{\sum_{t=1}^{T} \gamma_t(j,m)},$$

  где

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jm} b_{jm}(d_t)}{\sum_{m=1}^{M} c_{jm} b_{jm}(d_t)} \right].$$

Outline

Постановка задачи

- Мы раньше предполагали, что наша марковская модель хорошо описывает процесс.
- На самом деле это не всегда так.
- Давайте попробуем решить более практическую задачу: есть несколько сигналов, и мы должны их описать марковскими моделями так, чтобы как можно точнее отделять их друг от друга.

ML

- Стандартная идея в том, что мы хотим научиться сравнивать модели, т.е. для разных моделей λ_ν, $\nu = 1..V$, научиться оценивать $p(d^\nu|\lambda_\nu)$, а потом выбрать из них максимум.
- Это так называемый ML criterion (от maximum likelihood).

MMI

- Альтернатива — метод максимальной взаимной информации (maximum mutual information, MMI).
- В простейшей ситуации этот метод применяется, чтобы выяснить, какие параметры больше всего влияют на различение заданных классов.
- Если есть данные $x = (x_1, \ldots, x_n)$ и набор классов G, по которым их классифицируют, то нужно подсчитать информацию между x_i и классом C:

$$I(x_i, C) = H(C) - H(C|x_i),$$

т.е. максимизировать

$$H(C|x_i) = -\sum_{\in G} \sum_{v_j} p(c, x_i = v_j) \log p(c|x_i = v_j).$$

MMI

- Для марковских моделей это выражается как максимизация среднего расстояния между данными d и полным набором моделей $\lambda = (\lambda_1, \ldots, \lambda_V)$:

$$I_\nu = \max_\lambda \left[\log p(d|\lambda_\nu) - \log \sum_w p(d^\nu|\lambda_w) \right],$$

  т.е. мы отделяем правильную модель μ от других моделей на последовательности d.
- А теперь можно просуммировать по всем моделям и таким образом надеяться, что получится самый «разделённый» набор:

$$I = \max_\lambda \sum_{\nu=1}^{V} \left[\log p(d^\nu|\lambda_\nu) - \log \sum_w p(d^\nu|\lambda_w) \right],$$

MDI

- Третий подход: предположим, что сигнал не обязательно марковский, но у него есть некоторые ограничения (на корреляцию, например).
- Надо найти такие параметры, которые минимизируют разделяющую информацию (discrimination information, DI) между набором распределений Q, удовлетворяющих ограничениям, и набором марковских распределений p_λ:

$$D(Q\|p_\lambda) = \int q(y) \ln \frac{q(y)}{p(y)} dy.$$

- Т.е. мы предполагаем, что сигнал из Q, и ищем такую λ, чтобы расстояние до соответствующего элемента q было минимальным.
- Тоже по сути модифицированный алгоритм Баума–Велха.

Outline

Как сравнивать HMM?

- Вспомним пример с монеткой и рассмотрим две модели: $\lambda_1 = (A_1, B_1, \pi_1)$, $\lambda_2 = (A_2, B_2, \pi_2)$:

$$A_1 = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}, B_1 = \begin{pmatrix} q & 1-q \\ 1-q & q \end{pmatrix}, \pi_1 = (1/2\ 1/2),$$

$$A_2 = \begin{pmatrix} r & 1-r \\ 1-r & r \end{pmatrix}, B_2 = \begin{pmatrix} s & 1-s \\ 1-s & s \end{pmatrix}, \pi_2 = (1/2\ 1/2),$$

- Чтобы модели производили в среднем одинаковые последовательности наблюдений, нужно, чтобы $E[d_t = v_k|\lambda_1] = E[d_t = v_k|\lambda_2]$, т.е.

$$pq + (1-p)(1-q) = rs + (1-r)(1-s).$$

- Модели с $p = 0.6$, $q = 0.7$ и $r = 0.2$, $s = 13/30$ дадут одинаковые результаты.

Как сравнивать HMM?

- Нужна метрика. Определим расстояние между моделями

$$D(\lambda_1, \lambda_2) = \frac{1}{T}(\log p(d^{(2)}|\lambda_1) - \log p(d^{(2)}|\lambda_2)),$$

где $d^{(2)}$ порождено моделью λ_2, т.е. мы проверяем, насколько хорошо λ_1 соответствует наблюдениям по модели λ_2 по сравнению с самой λ_2.

- Это, конечно, несимметричное расстояние, поэтому обычно берут

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)}{2}.$$

Начальные значения параметров

- Алгоритм Баума–Велха, по сути своей градиентный, конечно, даёт только локальный максимум.
- Значит, важно, как выбирать начальные значения параметров.
- Для π и A на самом деле особой проблемы нет: обычно хорошо работают или случайные, или равномерные начальные значения.
- Для B хорошие начальные оценки могут быть весьма полезны в дискретном случае и практически необходимы в непрерывном.
- Как их получить?

Начальные значения параметров

- В непрерывном случае у нас распределение было смесью других распределений:

$$b_j(D) = \sum_{m=1}^{M} c_{jm} \mathcal{P}(D, \mu_{jm}, \sigma_{jm}).$$

- Как бы нам оценить начальные параметры μ_{jm}, σ_{jm}, имея данные всех наблюдений? Такой алгоритм у нас уже был...

Начальные значения параметров

- В непрерывном случае у нас распределение было смесью других распределений:

$$b_j(D) = \sum_{m=1}^{M} c_{jm} \mathcal{P}(D, \mu_{jm}, \sigma_{jm}).$$

- Как бы нам оценить начальные параметры μ_{jm}, σ_{jm}, имея данные всех наблюдений? Такой алгоритм у нас уже был...
- Кластеризация. Мы просто кластеризуем данные D (по каждому времени j отдельно) и получим неплохие начальные значения.

Интерполяция

- Данных часто не хватает. Параметров бывает слишком много.
- Один из путей — интерполяция. Выбрать вместе с λ модель поменьше λ', для которой данных достаточно, и считать интерполированную модель

$$\hat{\lambda} = \epsilon\lambda + (1 - \epsilon)\lambda'.$$

Интерполяция

- Главное — правильно оценить ϵ (это функция количества имеющихся тестовых данных).
- Есть специальные подходы, при которых данные делятся на $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, потом $\mathcal{D}_1$ используется для тренировки, а $\mathcal{D}_2$ — для оценки ϵ. Разбиение можно со временем двигать.
- Другой путь — добавлять специальные ограничения (например, $b_j(k) \geq \epsilon$).

Спасибо за внимание!

- Lecture notes, слайды и коды программ появятся на моей homepage:

  `http://logic.pdmi.ras.ru/~sergey/index.php?page=teaching`

- Присылайте любые замечания, коды программ на других языках, решения упражнений, новые численные примеры и прочее по адресам:
  `sergey@logic.pdmi.ras.ru`, `smartnik@inbox.ru`