СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ

Сергей Николенко

СПбГУ — Санкт-Петербург

11 ноября 2020 г.

Random facts:

- 11 ноября в Китае — День холостяков; дата была выбрана потому, что 11.11 символизирует людей, не состоящих в паре; самый популярный способ отметить — на обеде со своим единственным другом, но важно, чтобы каждый показал свою независимость
- 11 ноября 1572 г. Тихо Браге обнаружил вспышку сверхновой SN 1572 в созвездии Кассиопеи; по параллаксу Браге сделал вывод, что «новая звезда» находится намного дальше Луны
- 11 ноября 1842 г. в Пльзене был впервые подан новый сорт пива — Пильзнер
- 11 ноября 1983 г. калифорнийский студент Фред Коэн успешно завершил курсовую работу по созданию прототипа первого компьютерного вируса
- 11 ноября 2002 г. на arXiv появилась статья Григория Перельмана «The entropy formula for the Ricci flow and its geometric applications»

Рейтинг спортивного ЧГК

Рейтинг спортивного ЧГК

- Пример из практики: в какой-то момент я хотел сделать рейтинг спортивного «Что? Где? Когда?»:
    - участвуют команды по ≤ 6 человек, причём часто встречаются неполные команды;
    - игроки постоянно переходят между командами (поэтому TrueSkill);
    - в одном турнире могут участвовать до тысячи команд (синхронные турниры);
    - командам задаётся фиксированное число вопросов (36, 60, 90), т.е. в крупных турнирах очень много команд делят одно и то же место.

Рейтинг спортивного ЧГК

- Первое решение — система TrueSkill
- Расскажу о ней потом, когда будем говорить об Expectation Propagation
- У неё есть проблемы в постановке спортивного ЧГК, мы когда-то их отчасти решили, была сложная и интересная модель, она работала лучше базового TrueSkill
- Но...

- В какой-то момент база турниров ЧГК стала собирать повопросные результаты.
- Мы теперь знаем, на какие именно вопросы ответила та или иная команда.
- Так что когда я вернулся к задаче построения рейтинга ЧГК, задача стала существенно проще.

Рейтинг спортивного ЧГК

- Пример аналогичного приложения:
    - есть набор вопросов для теста из большого числа вопросов (например, IQ-тест или экзамен по какому-то предмету);
    - участники отвечают на случайное подмножество вопросов;
    - надо оценить участников, но уровень сложности вопросов нельзя заранее точно сбалансировать.
- «Что? Где? Когда?» — это оно и есть, только теперь участники объединяются в команды.

Baseline: логистическая регрессия

- Baseline — логистическая регрессия:
  - каждый игрок i моделируется скиллом s_i,
  - каждый вопрос q моделируется сложностью («лёгкостью») c_q,
  - добавим глобальное среднее μ,
  - обучим логистическую модель

$$p(x_{tq} \mid s_i, c_q) \sim \sigma(\mu + s_i + c_q)$$

для каждого игрока $i \in t$ команды-участницы $t \in T^{(d)}$ и каждого вопроса $q \in Q^{(d)}$, где $\sigma(x) = 1/(1 + e^x)$ — логистический сигмоид, x_{tq} — ответила ли команда t на вопрос q.

Модель со скрытыми переменными

- Логистическая модель предполагает фактически, что каждый игрок ответил на каждый вопрос, который взяла команда.
- Это неправда, мы не знаем, кто ответил, только знаем, что кто-то это сделал; а если команда не ответила, то никто.
- Это по идее похоже на presence-only data models (Ward et al., 2009; Royle et al., 2012).

Модель со скрытыми переменными

- Поэтому давайте сделаем модель со скрытыми переменными.
- Для каждой пары игрок-вопрос, добавим переменную z_{iq}, которая означает, что «игрок i ответил на вопрос q».
- На эти переменные есть такие ограничения:
  - если $x_{tq} = 0$, то $z_{iq} = 0$ для каждого игрока $i \in t$;
  - если $x_{tq} = 1$, то $z_{iq} = 1$ для по крайней мере одного игрока $i \in t$.

Модель со скрытыми переменными

- Параметры модели те же — скилл и сложность вопросов:

$$p(z_{iq} \mid s_i, c_q) \sim \sigma(\mu + s_i + c_q).$$

- Обучаем EM-алгоритмом:
    - E-шаг: зафиксируем все s_i и c_q, вычислим ожидания скрытых переменных z_{iq} как

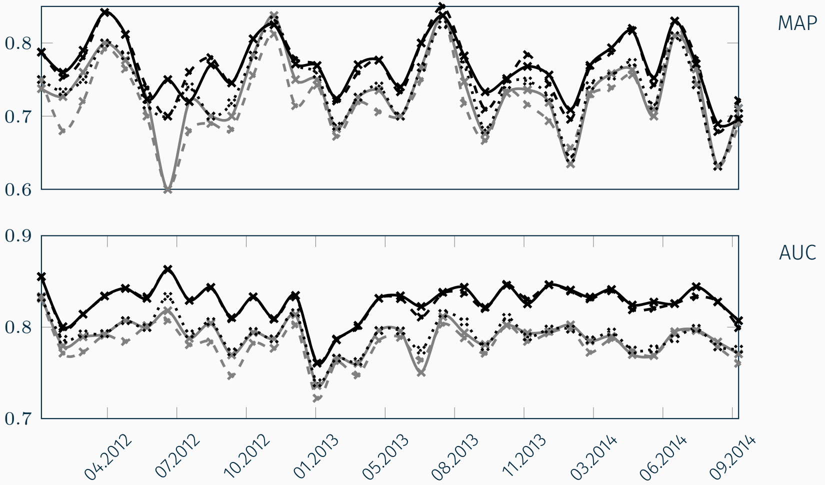
$$\mathbb{E}\left[z_{iq}\right] = \begin{cases} 0, & \text{если } x_{tq} = 0, \\ p(z_{iq} = 1 \mid \exists j \in t \; z_{jq} = 1) = \frac{\sigma(\mu + s_i + c_q)}{1 - \prod_{j \in t}\left(1 - \sigma(\mu + s_j + c_q)\right)}, & \text{если } x_{tq} = 1; \end{cases}$$

    - M-шаг: зафиксируем $\mathbb{E}\left[z_{iq}\right]$, обучим логистическую модель

$$\mathbb{E}\left[z_{iq}\right] \sim \sigma(\mu + s_i + c_q).$$

Результаты

- Ну и, конечно, работает хорошо.



Пример

Рейтинг ЧГК | ИГРОКИ | КОМАНДЫ | ТУРНИРЫ | ВОПРОСЫ | Рейтинг-лист игроков | РЕЛИЗ: | ЯНВАРЬ 2015

РЕЙТИНГ-ЛИСТ ИГРОКОВ НА ЯНВАРЬ 2015

Показывать по 100 записей

Введите id или начало фамилии:

Место	id	Фамилия	Имя	Отчество	Команда	Сыграно	Взято	Рейтинг
1	27177	Ромашова	Вероника	Михайловна	ЛКИ	5278	3784	545.753
2	3083	Белявский	Дмитрий	Михайлович	ЛКИ	4831	3396	543.520
3	27403	Руссо	Максим	Михайлович	ЛКИ	7180	5205	534.540
4	4270	Брутер	Александра	Владимировна	ЛКИ	8244	5974	533.405
5	18332	Либер	Александр	Витальевич	Рабочее название	8658	6210	532.921
6	1585	Архангельская	Юлия	Сергеевна	Ксеп	7542	5286	531.866
7	24384	Пашковский	Евгений	Александрович	ЛКИ	7552	5374	531.327
8	8333	Губанов	Антон	Александрович	Команда Губанова	4475	3153	530.871
9	16332	Крапиль	Николай	Валерьевич	Ксеп	6927	4899	530.559
10	21487	Моносов	Борис	Яковлевич	Команда Губанова	5115	3645	530.175

Пример

Рейтинг ЧГК ИГРОКИ КОМАНДЫ ТУРНИРЫ ВОПРОСЫ Александр Друзь vs. Максим Поташев

Ссылка на сайт рейтинга МАК

Друзь
Александр Абрамович

Поташев
Максим Оскарович

Ссылка на сайт рейтинга МАК

ИСТОРИЯ РЕЙТИНГА ИСТОРИЯ ТУРНИРОВ

ИСТОРИЯ РЕЙТИНГОВ

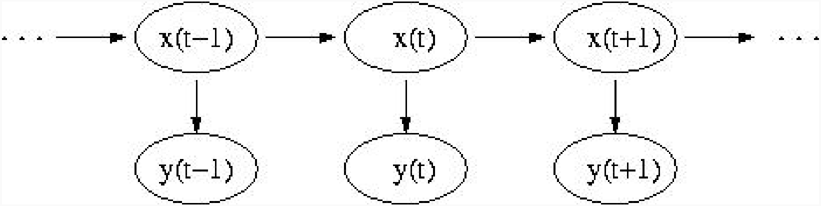
Рейтинг	Команда	Место	Дата	Место	Команда	Рейтинг
450.138	Трансфера	303	Январь 2015	31	Афина	514.643
446.669	Трансфера	336	Декабрь 2014	29	Афина	514.697
442.559	Трансфера	348	Ноябрь 2014	25	Афина	513.155
437.999	Трансфера	400	Октябрь 2014	25	Афина	512.083
445.791	Трансфера	306	Сентябрь 2014	23	Афина	515.737
441.473	Трансфера	372	Август 2014	23	Афина	516.513
446.730	Трансфера	284	Июль 2014	22	Афина	516.755
448.598	Трансфера	322	Июнь 2014	24	Афина	516.603

Скрытые марковские модели: основное

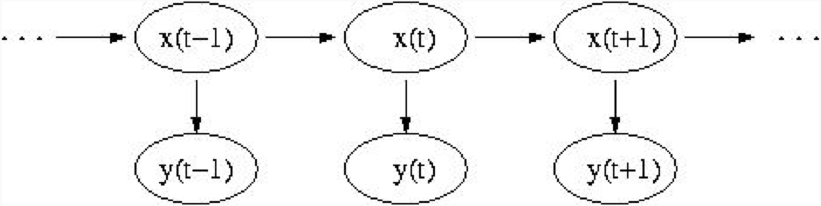
Марковские цепи

- Марковская цепь задаётся начальным распределением вероятностей $p^0(x)$ и вероятностями перехода $T(x'; x)$.
- $T(x'; x)$ — это распределение следующего элемента цепи в зависимости от следующего; распределение на $(t+1)$–м шаге равно
$$p^{t+1}(x') = \int T(x'; x) p^t(x) dx.$$
- В дискретном случае $T(x'; x)$ — это матрица вероятностей $p(x' = i | x = j)$.

- Мы будем находиться в дискретном случае.
- Марковская модель — это когда мы можем наблюдать какие-то функции от марковского процесса.



- Здесь $x(t)$ — сам процесс (модель), а $y(t)$ — то, что мы наблюдаем.
- Задача — определить скрытые параметры процесса.



- Главное свойство — следующее состояние не зависит от истории, только от предыдущего состояния.

$$p(x(t)=x_j|x(t-1)=x_{j_{t-1}},\dots,x(1)=x_{j_1})=$$
$$=p(x(t)=x_j|x(t-1)=x_{j_{t-1}}).$$

- Более того, эти вероятности $a_{ij}=p(x(t)=x_j|x(t-1)=x_i)$ ещё и от времени t не зависят.
- Эти вероятности и составляют матрицу перехода $A=(a_{ij})$.

Вероятности перехода

- Естественные свойства:
- $a_{ij} \geq 0.$
- $\sum_j a_{ij} = 1.$

Прямая задача

- Естественная задача: с какой вероятностью выпадет та или иная последовательность событий?
- Т.е. найти нужно для последовательности $Q = q_{i_1} \dots q_{i_k}$

$$p(Q|\text{модель}) = p(q_{i_1})p(q_{i_2}|q_{i_1}) \dots p(q_{i_k}|q_{i_{k-1}}).$$

- Казалось бы, это тривиально.
- Что же сложного в реальных задачах?

- А сложно то, что никто нам не скажет, что модель должна быть именно такой.
- И, кроме того, мы обычно наблюдаем не $x(t)$, т.е. реальные состояния модели, а $y(t)$, т.е. некоторую функцию от них (данные).
- Пример: распознавание речи.

Задачи скрытых марковских моделей

- Первая: найти вероятность последовательности наблюдений в данной модели.
- Вторая: найти «оптимальную» последовательность состояний при условии данной модели и данной последовательности наблюдений.
- Третья: найти наиболее правдоподобную модель (параметры модели).

Состояния и наблюдаемые

- $X = \{x_1, \dots, x_n\}$ — множество состояний.
- $V = \{v_1, \dots, v_m\}$ — алфавит, из которого мы выбираем наблюдаемые y (множество значений y).
- q_t — состояние во время t, y_t — наблюдаемая во время t.

Распределения

- $a_{ij} = p(q_{t+1} = x_j | q_t = x_i)$ — вероятность перехода из i в j.
- $b_j(k) = p(v_k | x_j)$ — вероятность получить данные v_k в состоянии j.
- Начальное распределение $\pi = \{\pi_j\}$, $\pi_j = p(q_1 = x_j)$.
- Данные будем обозначать через $D = d_1 \dots d_T$ (последовательность наблюдаемых, d_i принимают значения из V).

Комментарий

- Проще говоря, вот как работает HMM (hidden Markov model).
- Выберем начальное состояние x_1 по распределению π.
- По t от 1 до T:
  - Выберем наблюдаемую d_t по распределению $p(v_k|x_j)$.
  - Выберем следующее состояние по распределению $p(q_{t+1} = x_j | q_t = x_i)$.
- Таким алгоритмом можно выбрать случайную последовательность наблюдаемых.

Задачи

- Теперь можно формализовать постановку задач.
- Первая задача: по данной модели $\lambda = (A, B, \pi)$ и последовательности D найти $p(D|\lambda)$. Фактически, это нужно для того, чтобы оценить, насколько хорошо модель подходит к данным.
- Вторая задача: по данной модели λ и последовательности D найти «оптимальную» последовательность состояний $Q = q_1 \ldots q_T$. Как и раньше, будет два решения: «побитовое» и общее.
- Третья задача: оптимизировать параметры модели $\lambda = (A, B, \pi)$ так, чтобы максимизировать $p(D|\lambda)$ при данном D (найти модель максимального правдоподобия). Эта задача — главная, в ней и заключается обучение скрытых марковских моделей.

Постановка первой задачи

- Формально, первая задача выглядит так. Нужно найти

$$p(D|\lambda) = \sum_Q p(D|Q,\lambda)p(D|\lambda) =$$

$$= \sum_{q_1,\dots,q_T} b_{q_1}(d_1)\dots b_{q_T}(d_T)\pi_{q_1}a_{q_1q_2}\dots a_{q_{T-1}q_T}.$$

- Ничего не напоминает?

Суть решения первой задачи

- Правильно, это такая же задача маргинализации, как мы решаем всё время.
- Мы воспользуемся так называемой forward–backward procedure, по сути — динамическим программированием на решётке.
- Будем последовательно вычислять промежуточные величины вида

  $$\alpha_t(i) = p(d_1 \ldots d_t, q_t = x_i | \lambda),$$

  т.е. искомые вероятности, но ещё с учётом текущего состояния.

Решение первой задачи

- Инициализируем $\alpha_1(i) = \pi_i b_i(d_1)$.
- Шаг индукции:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i) a_{ij}\right] b_j(d_{t+1}).$$

- После того как дойдём до шага T, подсчитаем то, что нам нужно:

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

- Фактически, это только прямой проход, обратный нам здесь не понадобился.
- Что вычислял бы обратный проход?

Обратный проход

- Он вычислял бы условные вероятности $\beta_t(i) = p(d_{t+1} \dots d_T | q_t = x_i, \lambda)$.
- Их можно вычислить, проинициализировав $\beta_T(i) = 1$, а затем по индукции:

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij} b_j(d_{t+1}) \beta_{t+1}(j).$$

- Это нам пригодится чуть позже, при решении второй и третьей задачи.

Два варианта второй задачи

- Как мы уже упоминали, возможны два варианта.
- Первый: решать «побитово», отвечая на вопрос «какое наиболее вероятное состояние во время j?».
- Второй: решать задачу «какая наиболее вероятная последовательность состояний?».

Побитовое решение

- Рассмотрим вспомогательные переменные

$$\gamma_t(i) = p(q_t = x_i | D, \lambda).$$

- Наша задача – найти

$$q_t = \arg\max_{1 \le i \le n} \gamma_t(i), \quad 1 \le t \le T.$$

- Как это сделать?

Побитовое решение

- Выражаем через α и β:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(D|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{n} \alpha_t(i)\beta_t(i)}.$$

- На знаменатель можно не обращать внимания — нам нужен $\arg\max$.

Спасибо за внимание!