Лекция 11. Бейсови мрежи

Както вече сме дискутирали в предишната лекция, наивният Бейсов класификатор интензивно използва предположението, че стойностите на атрибутите $a_1, ..., a_n$ са условно независими при известна стойност на целевия атрибут v. Това предположение драстично намалява сложността на процеса на научаване на целевата функция. Когато то е удовлетворено, наивният Бейсов класификатор извежда оптималната Бейсова класификация. Обаче в много случаи предположението за условна независимост е прекалено ограничаващо.

Бейсовите мрежи на доверие (Bayesian belief networks) или просто *Бейсовите мрежи* описват вероятностното разпределение, управляващо някое множество от променливи чрез указване на определено множество от предположения за условна независимост заедно със самите условни вероятности. В отличие от наивния Бейсов класификатор, който предполага, че всички променливи са условно независими при зададена стойност на целевата променлива, Бейсовите мрежи позволяват да заявяват предположения за условна независимост, отнасящи се само за подмножества от променливи. По този начин Бейсовите мрежи представляват един междинен подход, който е по-малко ограничаващ от този, използващ глобалното предположение за условната независимост, правено от наивния Бейсов класификатор, но и по-гъвкав от този, който изобщо не използва предположенията за условна независимост. Бейсовите мрежи са предмет на интензивни изследвания при които са били разработени множество алгоритми както за тяхното научаване, така и за тяхното използване за правене на извод.

11.1. Условна независимост

Ще започнем разглеждане на Бейсовите мрежи с едно по-точно определение, какво представлява условна независимост. Нека X, Y и Z са три случайни целочислени променливи. Ще говорим, че X е *условно независима* от Y при зададена Z, ако вероятностното разпределение, управляващо X, е независимо от стойността на Y при зададена стойност на Z, т.е., ако:

$$(\forall x_i, y_i, z_i)\ P(X = x_i \mid Y = y_i, Z = z_i) = P(X = x_i \mid Z = z_i),$$

където $x_i \in V(X)$, $y_i \in V(Y)$, $z_i \in V(Z)$ ($V(A)$ означава множество от възможни стойности на променлива A). Посоченият по-горе израз ще записваме за краткост във вида $P(X|Y, Z) = P(X|Z)$. Тази дефиниция на условната независимост може да бъде разширена и на множество от променливи. Ще говорим, че множеството от променливи $X_1, ..., X_l$ е условно независимо от множеството променливи $Y_1,...,Y_m$, при зададено множеството от променливи $Z_1, ..., Z_n$, ако:

$$P(X_1,..., X_l \mid Y_1,...,Y_m, Z_1,...,Z_n) = P(X_1, ...,X_l \mid Z_1, ...,Z_n)$$

Обърнете внимание на съответствие между това определение и използване на условната независимост в определението на наивния Бейсов класификатор. Наивният Бейсов класификатор подразбира, че стойността на атрибута A_1 е условно независима

от стойността на атрибута A_2 при зададена стойност на целевия атрибут V. Това позволява на наивния Бейсов класификатор да изчислява $P(A_1, A_2|V)$ по следния начин:

$$P(A_1, A_2|V) = P(A_1|A_2,V)P(A_2|V) \quad (11.1)$$
$$= P(A_1|V)P(A_2|V) \quad (11.2)$$

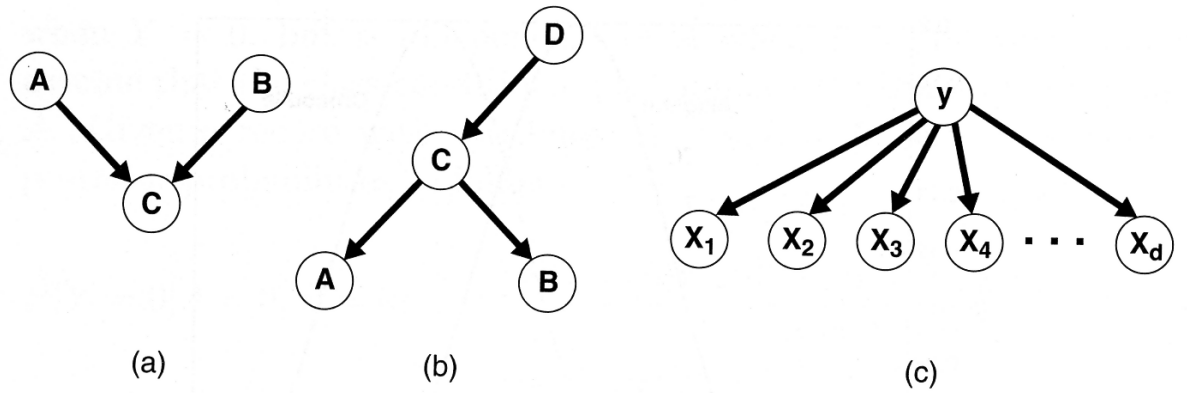
Уравнение (11.1) е просто една по-обща форма на правилото за умножение на вероятности от предишната лекция, а уравнение (11.2) следва от (11.1), тъй като ако A_1 е условно независимо от A_2 при зададено V, то съгласно нашето определение $P(A_1|A_2, V) = P(A_1|V)$.

11.2. Представяне на Бейсови мрежи

Бейсовата мрежа е едно графично представяне на вероятностните отношения между определено множество от случайни променливи и се състои от два основни елемента:

1. Насочен ацикличен граф (Directed Acyclic Graph – DAG), кодиращ зависимости между променливи, и
2. Вероятностни таблици, свързващи всеки възел в графа с възли, които са неговите непосредствени наследници.

Нека да разгледаме три случайни променливи – A, B и C, като A и B са независими променливи, всяка от които имат пряко влияние върху третата променлива – C. Връзките между променливи могат да бъдат представени като насочения ацикличен граф, показан на Фигура 11-1 (a). Всеки възел в графа представя една променлива, а всяка дъга задава зависимост между двойка променливи. Ако има насочена връзка от X към Y, то X се нарича *родител* на Y, а Y – *дете* на X. Освен това, ако съществува някой насочен път, водещ от X към Z, то X се нарича *предшественик* на Z, а Z – *потомък* на X. Например на диаграмата, показана на Фигура 11-1 (b), A е потомък на D и D е предшественик на B. Овен това B и D *не са* потомци на A.



Фиг. 11-1. Представяне на вероятностни връзки чрез използване на насочения ацикличен граф

Едно важно свойство на Бейсовите мрежи може да бъде формулирано по следния начин:

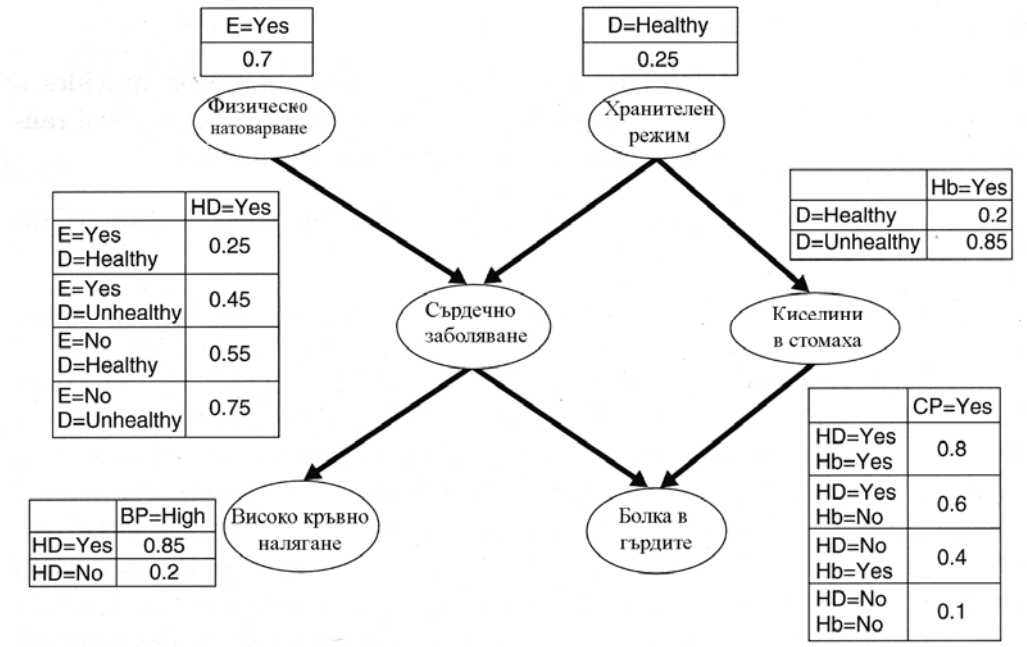
Свойство 1 (Условна независимост). Един възел в Бейсовата мрежа е условно независим от своите възли, които не са му потомци, ако са известни неговите възли – родители.

На диаграма, показана на Фигура 11-1 (b) *A* е условно независима както от *B*, така и от *D* при зададена *C*, тъй като възлите за *B* и *D* не са потомци на възела *A*. Предположението за условната независимост, правено от един наивен Бейсов класификатор, също може да бъде представено чрез Бейсовата мрежа. То е показано на Фигура 11-1 (c), в която *Y* представлява целевата променлива, а $\{X_1, X_2, ..., X_d\}$ – множеството от атрибути.

Освен условията за условна вероятност, налагани от топологията на мрежата, всеки възел в мрежата е свързан с една вероятностна таблица.

1. Ако възел *X* няма родители, то таблицата съдържа само априорната вероятност *P(X)*.
2. Ако възел *X* има само един родител – *Y*, то таблицата съдържа условната вероятност *P(X|Y)*.
3. Ако възел *X* има няколко родителя – $\{Y_1, ..., Y_k\}$, то таблицата съдържа условната вероятност $P(X|Y_1, ..., Y_k)$.

На Фигура 11-2 е показан пример на Бейсовата мрежа, моделираща пациенти със сърдечни заболявания или киселини в стомаха. Всяка променлива е двоична. Родителските възли за *сърдечно заболяване* (HD) съответстват на рискови фактори, които могат да причинят това заболяване – това са *физическо натоварване* (E) и *хранителен режим* (D). Възели-деца за *сърдечното заболяване* съответстват на симптоми на това заболяване – *болка в гърдите* (CP) и *високо кръвно налягане* (BP). Например, Фиг. 11-2 показва, че *киселини в стомаха* (Hb) могат да бъдат резултат от нездравословен режим на хранене и могат да доведат до появата на болка в гърдите.



Фиг. 11-2. Бейсова мрежа за диагностициране на сърдечно заболяване и киселини в стомаха при пациенти

Възлите, асоциирани с рисковите фактори, съдържат само априорните вероятности, докато възлите за сърдечното заболяване, киселинните в стомаха и техните симптоми съдържат условните вероятности. За да спестим място, някои от вероятностите не са указани на фигурата. Пропуснатите вероятности могат лесно да бъдат изчислени, имайки предвид, че $P(X = \neg x) = 1 - P(X = x)$ и $P(X = \neg x|Y) = 1 - P(X = x|Y)$. Например условната вероятност:

$P(HD = No|E = No, D = Healthy) = 1 - P(HD = Yes|E = No, D = Healthy) = 1 - 0.55 = 0.45$

11.3. Вероятностен извод

Бейсовите мрежи могат да се използват за извеждане на стойността на някоя целева променлива (например, *сърдечно заболяване* - HD) при наблюдавани стойности на други променливи. Тъй като мрежите работят със случайни променливи, то няма да е коректно да се назначава на целевата променлива само една определена стойност. На практика се извежда вероятностното разпределение на целевата променлива, което определя вероятността, че променливата ще приема всяка от нейните възможни стойности при наблюдаваните стойности на други променливи. Този вероятностен извод може да бъде доста лесен, ако стойностите на други променливи в мрежата са известни абсолютно точно. В по-общ случай вероятностните разпределения на някои променливи трябва да бъдат изведени при наблюдаване само на *някое подмножество* от други променливи. В общия случай една Бейсова мрежа може да се използва за изчисляване на вероятностното разпределение на всяко едно подмножество от променливите в мрежата при наличие на стойности или разпределения на всяко подмножество от останалите променливи.

Съвместната вероятност на всяко назначаване на конкретни стойности $(y_1, \ldots, y_n)$ на променливите в мрежата $(Y_1, \ldots, Y_n)$ се изчислява по формулата:

$$P(y_1,\ldots,y_n) = \prod_{i=1}^{n} P(y_i \mid \pi(Y_i)),$$

където $\pi(Y_i)$ означава множеството от променливи – родители на Y_i в мрежата.

11.3.1. Примери на вероятностен извод

Да видим, как можем да използваме Бейсовата мрежа, изобразена на Фигура 11-2, за извеждане на диагноза, дали някой пациент има сърдечно заболяване. Ще разгледаме няколко случая, илюстриращи как може диагнозата да бъде изведена при различни налични данни.

Случай 1. Без никакви наблюдавани данни

Дори ако нямаме никаква предварителна информация (т.е. наблюдавани данни), ние може да използваме Бейсовата мрежа, за да преценим, доколко е вероятно пациентът да има сърдечно заболяване. Това става чрез изчисляване на вероятности $P(HD = Yes)$ и $P(HD = No)$. За да опростим нотацията, да означим с $\alpha \in \{Yes, No\}$ двоичните стойности

на променливата *физическо натоварване* (E), а чрез $\beta \in \{Healthy, Unhealthy\}$ - стойностите на променлива *хранителен режим* (D).

$$P(HD = Yes) = \sum_{\alpha}\sum_{\beta} P(HD = Yes \mid E = \alpha, D = \beta)P(E = \alpha, D = \beta) =$$
$$= \sum_{\alpha}\sum_{\beta} P(HD = Yes \mid E = \alpha, D = \beta)P(E = \alpha)P(D = \beta) =$$
$$= 0.25 \times 0.7 \times 0.25 + 0.45 \times 0.7 \times 0.75 + 0.55 \times 0.3 \times 0.25 + 0.75 \times 0.3 \times 0.75 =$$
$$= 0.49.$$

Тъй като *P(HD = No) = 1 – P(HD = Yes) = 0.51*, то пациентът има малко по-голям шанс да няма сърдечното заболяване.

Случай 2. При високо кръвно налягане

Ако пациентът има високо кръвно налягане, ние можем да изведем диагноза за вероятното сърдечно заболяване чрез сравняване на апостериорните вероятности *P(HD = Yes|BP = High)* и *P(HD = No|BP = High).* За да направим това трябва да изчисли вероятността *P(BP =High):*

$$P(BP = High) = \sum_{\gamma} P(BP = High \mid HD = \gamma)P(HD = \gamma) =$$
$$= 0.85 \times 0.49 + 0.2 \times 0.51 = 0.5185.$$

където $\gamma \in \{Yes, No\}$. Следователно апостериорната вероятност, че пациентът има сърдечно заболяване е:

$$P(HD = Yes \mid BP = High) = \frac{P(BP = High \mid HD = Yes)P(HD = Yes)}{P(BP = High)} = \frac{0.85 \times 0.49}{0.5185} = 0.8033.$$

Следователно *P(HD = No|BP = High) = 1 – P(HD = Yes|BP = High) = 1 – 0.8033 = 0.1967.* И така, наличието на високо кръвно налягане повишава риск от наличие на сърдечно заболяване.

Случай 3. Високо кръвно налягане, здравословен хранителен режим и регулярно физическо натоварване

Да предположим, че пациентът има високо кръвно налягане, но има регулярно физическо натоварване и има здравословен хранителен режим. Как тази информация ще повлияе на нашата диагноза? При наличието на новата информация вероятността, че пациентът има сърдечно заболяване е:

$$P(HD = Yes \mid BP = High, D = Healty, E = Yes) =$$
$$= \frac{P(BP = High \mid HD = Yes, D = Healty, E = Yes)}{P(BP = High \mid D = Healthy, E = Yes)} P(HD = Yes \mid D = Healthy, E = Yes) =$$
$$= \frac{P(BP = High \mid HD = Yes)P(HD = Yes \mid D = Healthy, E = Yes)}{\sum_{\gamma} P(BP = High \mid HD = \gamma)P(HD = \gamma \mid D = Healthy, E = Yes)} =$$
$$= \frac{0.85 \times 0.25}{0.85 \times 0.25 + 0.2 \times 0.75} = 0.5862.$$

Вероятността, че пациентът няма сърдечно заболяване е:

$$P(HD = No \mid BP = High, D = Healthy, D = Yes) = 1 - 0.5862 = 0.4138.$$

И така, моделът утвърждава, че здравословното хранене и редовно физическо натоварване намаляват риска от сърдечни заболявания.

В общия случай точният извод на вероятности за произволна Бейсова мрежа е NP-пълна задача (Cooper 1990). Множество различни методи са били предложени за вероятностния извод в Бейсовите мрежи, включващи както точните методи, така и приближени методи, които жертват точността в името на ефективността. Например, методите Монте Каро предлагат приближени решения чрез случайна извадка на разпределенията на ненаблюдавани променливи (Pradham and Dagum 1996). На теория дори приближеният извод на вероятности в Бейсовите мрежи може да бъде NP-пълна задача (Dagum and Luby 1993). За щастие на практика приближените методи са доказали, че могат да бъдат много полезни в повечето от случаи.

11.4. Научаване на Бейсови мрежи

Могат ли да бъдат създадени ефективни алгоритми за научаване на Бейсови мрежи от обучаващи данни? Този въпрос е предмет на текущи изследвания. Отговорът на този въпрос зависи от различни предположения, използвани при решаване на тази задача за обучение. Първо, дали структурата на мрежа е известна предварително или трябва да бъде изведена от обучаващите данни. Второ, дали всички променливи в мрежа могат да бъдат наблюдавани директно във всеки обучаващ пример или само част от тях да е достъпна за наблюдаване. Ще започнем от най-лесния случай – топологията на мрежа се извежда от експертните знания и всички променливи от мрежата присъстват в обучаващите примери.

11.4.1. Построяване на Бейсова мрежа от експертни знания

По-долу е приведен алгоритъм, който систематично създава топология на Бейсова мрежа, използвайки знания на експерти.

Алгоритъм за създаване на топологията на Бейсова мрежа.

1. Нека $T = \{X_1, \ldots, X_d\}$ означава общата наредба на променливи.
2. **for** $j = 1$ to d **do**
3. Нека $X_{T(j)}$ означава j-а най-висока по ред променлива в T.
4. Нека $\pi(X_{T(j)}) = \{X_{T(1)}, X_{T(2)}, \ldots, X_{T(j-1)}\}$ означава множеството от променливи, предшестващи $X_{T(j)}$.
5. Изтрий от $\pi(X_{T(j)})$ тези променливи, които не въздействат върху X_j (използвайки предварителните знания)
6. Създай дъга между $X_{T(j)}$ и останалите променливи в $\pi(X_{T(j)})$.
7. **end for**

Пример 1. Да разгледаме променливи, показани на Фигура 11-2. Нека да смятаме, че след извършване на Стъпка 1 променливите са подредени по следния начин: *{E, D, HD,*

Hb, CP, BP}. Със стъпки от 2 до 7, започвайки с променлива *D*, получаваме следните условни вероятности:

- *P(D|E)* се опростява до *P(D)*.
- *P(HD|E, D)* не може да се опрости.
- *P(Hb|HD, E, D)* се опростява до *P(Hb|D)*.
- *P(CP|Hb, HD, E, D)* се опростява до *P(CP|Hb, HD)*.
- *P(BP|CP, Hb, HD, E, D)* се опростява до *P(BP|HD)*

Въз основа на тези условни вероятности вече можем да създадем дъги между възлите *(E, HD)*, *(D, HD)*, *(D, Hb)*, *(HD, CP)*, *(Hb, CP)* и *(HD, BP)*. В резултат получаваме структура на мрежата, показана на Фигура 11-2.

Приведеният алгоритъм гарантира създаване на топология, която не съдържа цикли. Доказателството за този факт е доста праволинейно. Ако съществува някакъв цикъл, то в топологията трябва да присъства най-малко една дъга, свързваща възлите от по-нисък ред с тези с по-високия и най-малко една дъга, свързваща възлите от по-висок ред с тези от по-нисък. Тъй като алгоритъм не позволява създаване на дъги, свързващи възли от по-нисък ред с тези от по-висок, в създаваната топология не може да има цикли.

Обаче топологията може да се промени, ако се използва друга схема на подреждане на променливи. Някои от топологиите могат да бъдат по-лоши от други, тъй като съдържат повече дъги, свързващи различни двойки възли в мрежата. По принцип е възможна ситуацията, в която трябва проверим всички *d!* възможни наредби на променливи, за да определим най-подходяща топология – тази задача е доста скъпа от изчислителната гледна точка. Един алтернативен подход се състои в разбиването на променливите на причини и следствия – след това всеки възел-причина се съединява с дъги със свои възли-следствия. Този подход значително опростява процеса на създаване на структура на Бейсови мрежи.

След като една подходяща топология е създадена, се определят вероятностните таблици, асоциирани с всеки възел на мрежата. Предполагайки, че всички променливи присъстват в обучаващите данни, изчисляването на оценки за съответните условни вероятности става чрез определяне на съответните честоти на срещания – по същия начин, както това се прави в наивния Бейсов класификатор.

11.4.2. Научаване на структура на Бейсова мрежа от примери

Задачата за научаване на структура на Бейсова мрежа е доста сложна. В работата (Cooper and Herskovits 1992) е предложена една оценъчна метрика за избор между алтернативни структури. Авторите също така предлагат един евристичен алгоритъм за търсене, наречен K_2, който научава структурата на една Бейсова мрежа, когато всички променливите в нея са директно наблюдаеми. Както и други подобни алгоритми K_2 използва евристичното претърсване, което опитва да намери баланс между сложността на научаваната мрежа и нейната точност върху обучаващите данни. При един от експериментите на K_2 са били дадени 3000 обучаващи примера, генерирани по случаен начин от една ръчно конструирана Бейсова мрежа, съдържаща 37 възела и 46 дъги. Тази конкретна мрежа описвала потенциални проблеми при анестезия, прилагана при хирургически операции в една болница. Като допълнение към тези данни на

програмата също така е било дадено началното подреждане на 37 променливи, което е било съвместимо с частичната подредба на зависимости между променливи в действителната мрежа. Програмата е успяла да реконструира структурата на Бейсовата мрежа почти точно – само с едно изключение от една некоректно изтрита дъга и една некоректно добавена дъга.

Били са разработени и подходи за научаване на структури на Бейсови мрежи, основани на ограничения (constraint-based). При тях от данните се извеждат връзки за зависимост или независимост между променливи, а след това тези връзки се използват за конструиране на Бейсови мрежи. За по-детайлен обзор на методи за научаване на структури на Бейсови мрежи виж, например (Heckerman 1995).

11.4.3. Научаване на стойности на вероятности в мрежата

Както беше казано по-рано, създаването на една Бейсова мрежа включва две стъпки – на първата се научава структурата на мрежата, а на втората се оценяват стойностите на вероятности, асоциирани с всеки възел от мрежата. Когато структурата е известна и всички променливи са наблюдаеми (т.е. техните стойности присъстват в обучаващите данни), научаването на вероятностите (условни и априорни) е лесно и се извършва по същия начин, както при наивния Бейсов класификатор. Когато структурата на мрежата е известна, но само някои от стойностите на променливи са наблюдаеми в обучаващите данни, задачата е по-тежка. Проблемът е донякъде сходен със задачата за научаване на теглата на скритите възли в изкуствени невронни мрежи, където стойностите на входните и изходните възли са зададени, а стойностите на скритите възли не присъстват в обучаващите данни.

11.4.3.1. Обучение на Бейсови мрежи по метода на градиентното изкачване

Ръсел (Russel 1993) е предложил процедура за научаване на стойностите на условни вероятности в мрежата, която напомня метода за научаване на тегла, използван при обучение на невронни мрежи. Тази процедура претърсва пространство от хипотези, съответстващо на множество от всички възможни стойности на условни вероятности от таблиците. Функцията на качество, която се максимизира при градиентното изкачване, е вероятност $P(D|h)$ на наблюдавани обучаващи данни D при зададена хипотеза h. По определение, това отговаря на търсенето на максимално възможна (ML) хипотеза за стойностите на условните вероятности в мрежата.

Използваното правило за градиентното изкачване максимизира $P(D|h)$ по отношение на параметрите, които определят таблици с условни вероятности в Бейсовата мрежа. Нека w_{ijk} означава една стойност в една от клетките в таблиците с условни вероятности. В частност w_{ijk} означава условната вероятност, че променливата Y_i в мрежата приема стойност y_{ij} при условие, че са зададени стойности u_{ik} на променливи U_i, които са непосредствени родители на Y_i. Например, ако w_{ijk} е стойността на горната дясна клетка в таблицата с условни вероятности, свързана с променлива *Болка в гърдите* (CP) от Фигура 11-2, то Y_i е променливата CP, U_i са променливи *Сърдечно заболяване* (HD) и *Киселини в стомаха* (Hb), $y_{ij} = Yes$ и $u_{ik} = <Yes, Yes>$.

Градиентът на $ln\ P(D|h)$ се задава от частни производни $\frac{\partial \ln P(D \mid h)}{\partial w_{ijk}}$ за всяко w_{ijk}. Всяка от тези производни може да бъде изчислена по формула:

$$\frac{\partial \ln P(D \mid h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{P(Y_i = y_{ij}, U_i = u_{ik} \mid d)}{w_{ijk}} \quad (11.3)$$

Например, за да изчислим производната от *ln P(D|h)* по отношение на най-горната дясна клетка в таблицата, свързана с променлива *Болка в гърдите* (CP) от Фигура 11-2, трябва да изчислим стойността на *P(CP = Yes, HD = Yes, Hb = Yes|d)* за всеки обучаващ пример *d* от *D*. Когато тези променливи не са наблюдаеми в обучаващия пример *d*, съответната вероятност може да бъде изчислена от наблюдаемите в *d* променливи с помощта на стандартния извод в Бейсови мрежи. На практика необходимите стойности лесно се извеждат от изчисления, извършвани при различни изводи в Бейсова мрежа, така че обучението може да бъде извършвано с малка допълнителна цена, когато Бейсовата мрежа се използва за извод и се постъпват нови данни.

Преди да дефинираме процедурата за обучение с използване на градиентното изкачване, трябва да обърнем внимание на следното. Ние изискваме, че при обновяване на теглата w_{ijk} те трябва да остават коректни вероятности, т.е да са в интервал *[0, 1]*. Ние също така изискваме, че сумата $\sum_j w_{ijk}$ да остава равна на 1 за всички *i, k*. Тези ограничения могат да бъдат удовлетворени чрез една дву-стъпкова процедура по обновяване на теглата. Първо, ще обновяваме всяко w_{ijk} чрез градиентното изкачване:

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P(y_{ij}, u_{ik} \mid d)}{w_{ijk}}$$

където η е малка константа, наричана скоростта на обучение. Второ, ще повторно нормализираме теглата w_{ijk}, за да гарантираме, че формулираните по-горе ограничения са удовлетворени. Както се дискутира в (Russel et al. 1993) този процес има сходимост към една локална максимално възможна хипотеза за условните вероятности в Бейсовата мрежа.

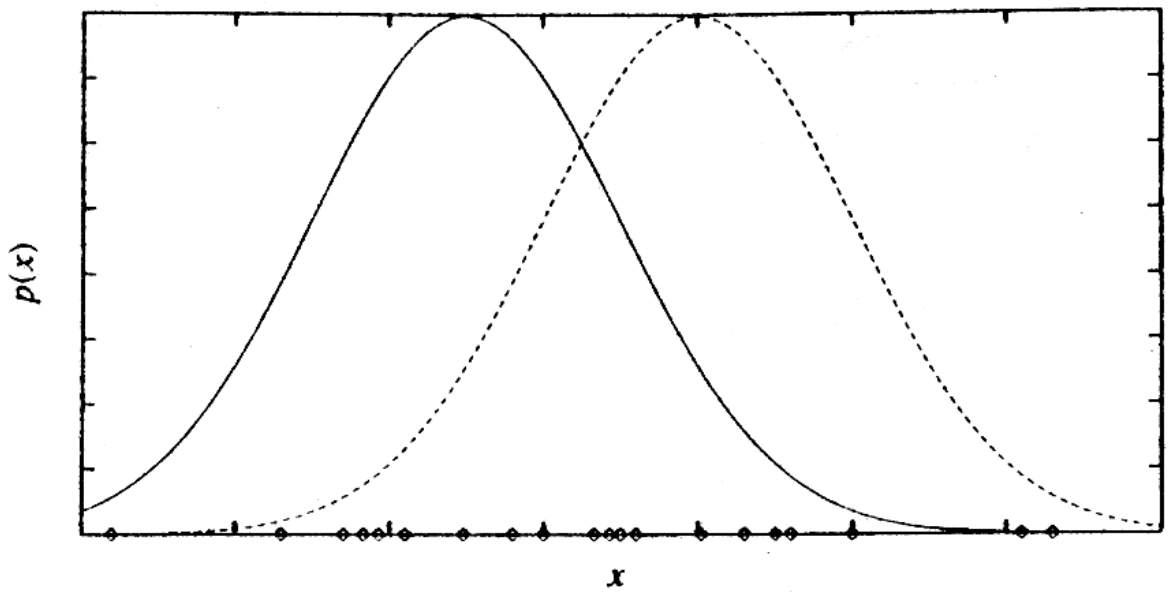
Една от алтернативите на разгледания по-горе метод за обучение на Бейсови мрежи, е алгоритъм EM, който ще разгледаме в следващия раздел.

4.3.2. Алгоритъм EM

Алгоритъм EM (Dempster et al. 1997) е един широко използван подход за самообучение при наличие на ненаблюдаеми променливи. Алгоритмът може да се използва дори за променливи, чиито стойности никога директно не наблюдаеми, ако е известно тяхното вероятностно разпределение. Алгоритмът е използван за обучение на Бейсови мрежи (Heckerman 1995), за невронни мрежи от сферични функции, както и за различни клъстеризационни алгоритми (Cheeseman 1988). Името на алгоритъма произлиза от Expectation (Очакване) – Maximization (Максимизация). Алгоритмът започва работата си с една произволна начална хипотеза. След това той многократно изчислява очакваните стойности на скритите променливи (предполагайки, че текущата хипотеза е коректна), а след това преизчислява максимално възможна хипотеза предполагайки, че скритите променливи имат очакваните стойности (изчислени на първата стъпка). Тази процедура има сходимост към някоя локална максимално възможна хипотеза за стойности на скритите променливи.

Оценяване на средни стойности на k Гаусиани

Най-лесният начин за запознаване с алгоритъма EM е чрез пример. Ще разгледаме задача, в която данните D представляват множество примери, генерирани от едно вероятностно разпределение, представляващо комбинация от k различни нормални разпределения. Тази задача е илюстрирана от Фиг. 11-3, в която $k = 2$ и примерите са точки, разположени по оста x. Всеки пример е генериран чрез прилагане на двустъпковата процедура: първо, се избира по случаен начин едно от k нормални разпределения; второ, се генерира един случаен пример x_i съгласно това избрано разпределение. Описаната процедура се повтаря, за да генерира множеството от точки, показани на Фиг. 11-3. За да улесни обяснението, ще разгледаме един специален случай, когато изборът на едно единствено нормално разпределение на всяка стъпка се базира на избора с еднаква вероятност, като всяко от k нормални разпределения има една и съща дисперсия σ, която е известна. Задачата за самообучение е да бъде изведена някаква хипотеза $h = <\mu_1, ..., \mu_k>$, която описва средните на всяко от k разпределения. Ще искаме да намерим максимално възможна хипотеза за тези средни, т.е. хипотеза h, която максимизира $p(D|h)$.



Фиг. 11-3. Примери, генерирани от една комбинация от две нормални разпределения с еднаква дисперсия. Примерите са показани като точки, разположени по оста x. Ако средните на нормалните разпределения не са известни, за намирането на техните максимално възможни приближения може да се ползва алгоритъм EM.

Искам да напомня, че решаването на задача за намиране на максимално възможна хипотеза за средното на едно единствено нормално разпределение по зададени наблюдавани примери $x_1, ..., x_m$ е лесно. В една от предишните лекции ние показахме, че максимално възможната хипотеза е тази, която минимизира сумата от квадрати на грешки на m примера. Следователно

$$\mu_{ML} = \arg\min_{\mu} \sum_{i=1}^{m} (x_i - \mu)^2 \quad (11.4)$$

Тази сума се минимизира от средното на извадката:

$$\mu_{ML} = \frac{1}{m} \sum_{i-1}^{m} x_i \quad (11.5)$$

Нашата задача, обаче, включва комбинация от k различни нормални разпределения, и ние не можем да наблюдаваме, кои примери са генерирани от кое разпределение. По този начин ние имаме един типичен пример на задачата със скритите променливи. В случая, показан на Фиг. 11-3, можем да смятаме, че пълното описание на всеки пример се състои от тройката $<x_i, z_{i1}, z_{i2}>$, където x_i е наблюдаваната стойност на i-я пример, а z_{i1} и z_{i2} указват, кое от двете нормални разпределения е било използвано за генериране на стойността x_i. В частност, z_{ij} приема стойност 1, ако x_i е било създадено от j-то нормално разпределение, или 0 – в противен случай. В този случай x_i е наблюдаваната променлива в описанието на примера, а z_{i1} и z_{i2} са скрити (ненаблюдаеми) променливи. Ако стойностите на z_{i1} и z_{i2} се наблюдават, за намиране на μ_1 и μ_2 можем да използваме уравнение (11-4). Обаче, тъй като те не са известни, ще използваме алгоритъм EM.

В задачата за k средни алгоритъмът EM търси максимално възможна хипотеза чрез повтарящо се преоценяване на очакваните стойности на скритите променливи z_{ij} при зададената текуща хипотеза $<\mu_1, \ldots, \mu_k>$, а след това преизчислява максимално възможна хипотеза, използвайки тези очаквани стойности на скритите променливи. В този раздел ще опишем частния случай на алгоритъма EM, решаващ тази конкретна задача, а в следващия раздел – неговата обща форма.

В случая на задачата за оценяване на средните на два нормални разпределения, показани на Фиг. 11-3, алгоритъм EM отначало инициализира хипотезата $h = < \mu_1, \mu_2>$, където μ_1 и μ_2 са произволни начални стойности. След това той итеративно преизчислява h чрез повтаряне на следните две стъпки, докато процедурата не постигне една постоянна стойност на h.

Стъпка 1. Изчисли очакваната стойност $E[z_{ij}]$ на всяка от скрити променливи z_{ij}, предполагайки, че е изпълнена текущата хипотеза $h = < \mu_1, \mu_2>$.

Стъпка 2. Изчисли една нова максимално възможна хипотеза $h' = < \mu'_1, \mu'_2>$, предполагайки, че всяка от скритите променливи z_{ij} приема нейната стойност $E[z_{ij}]$, изчислена на Стъпка 1. Замени хипотезата $h = < \mu_1, \mu_2>$ с хипотеза $h' = < \mu'_1, \mu'_2>$ и започни нова итерация.

Нека да видим, как тези две стъпки могат да бъдат реализирани на практика. Стъпка 1 трябва да изчисли очакваната стойност за всяка z_{ij}. Тази стойност $E[z_{ij}]$ е просто вероятността, че пример x_i е бил генериран от j-то нормално разпределение:

$$E[z_{ij}] = \frac{p(x = x_i \mid \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i \mid \mu = \mu_n)} = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

И така, първата стъпка е реализирана чрез подставяне на текущите стойности на $< \mu_1, \mu_2>$ и на наблюдаваната стойност x_i в горния израз.

На втората стъпка ние използваме $E[z_{ij}]$, изчислено на първата стъпка, за да изведем нова максимално възможна хипотеза $h' = <\mu'_1, \mu'_2>$. Тази максимално възможна хипотеза се изчислява по следния начин:

$$\mu_j \leftarrow \frac{1}{m} \sum_{i=1}^{m} E[z_{ij}] x_i$$

Обърнете внимание, че тази формула прилича на формулата (11.4) – в нашия случай тя е просто претегленото средно на извадката за μ_j, като всеки пример е претеглен с очакването $E[z_{ij}]$, че той е генериран от j-то нормално разпределение.

Описаният по-горе алгоритъм за оценяване на средните на една комбинация от k нормални разпределения илюстрира основната идея на EM подхода – текущата хипотеза се използва за оценяване на ненаблюдаемите променливи, а очакваните стойности на тези променливи след това се използват за изчисляване на нова по-добра хипотеза. Може да се докаже, че на всяка итерация от този цикъл алгоритъм EM увеличава стойнстта на $P(D|h)$, докато тя не попадне в някой локален максимум. Алгоритъмът има сходимост към някоя локално максимално възможна хипотеза за $<\mu_1, \mu_2>$.

Обща формулировка на алгоритъма EM

Алгоритъм EM може да бъде приложен за решаване на различни задачи, в които трябва да бъдат оценено множество от параметри θ, описващи някое вероятностно разпределение, когато се наблюдава само една част от пълните данни, генерирани от това разпределение. В разгледания по-горе пример с двете средни, интересуващите нас параметрите бяха $\theta = <\mu_1, \mu_2>$, а пълните данни представляваха тройки $<x_i, z_{i1}, z_{i2}>$, в които само x_i бяха наблюдаеми. В общия случай ще означаваме с $X = \{x_1, \ldots, x_m\}$ данни, наблюдавани в едно множество от m независими примера, а с $Z = \{z_1, \ldots, z_m\}$ – ненаблюдаеми данни в същите примери, така че $Y = X \cup Z$ означават пълните данни. Обърнете внимание, че ненаблюдаемите данни Z могат да се разглеждат като една случайна променлива, чието вероятностно разпределение зависи от неизвестни за нас параметри θ и от наблюдаемите данни X. По същия начин и Y е случайна променлива, тъй като е определена в термини на случайната променлива Z. Ще използваме h за означаване на текущите предполагаеми стойности на параметрите θ и h' – за уточнената хипотеза, която се получава на всяка итерация на алгоритъма EM.

Алгоритъм EM търси максимално вероятна хипотеза h' чрез намиране на такава h', която максимизира $E[\ln P(Y|h')]$. Тази очаквана стойност се взима по отношение на вероятностното разпределение на Y, което се определя от неизвестните параметри θ. Да разгледаме, какво точно означава този израз. Първо, $P(Y|h')$ е вероятността на пълните данни Y при зададена хипотеза h'. Разумно е да поискаме да намерим такава h', която максимизира някоя функция от тази величина. Второ, максимизирайки логаритъм от тази величина – $\ln P(Y|h')$ ние също така максимизираме и $P(Y|h')$. Трето, ние въвеждаме очакваната стойност $E[\ln P(Y|h')]$, защото пълните данни Y са случайна променлива. Имайки предвид, че пълните данни Y са комбинация от наблюдаемите данни X и ненаблюдаемите данни Z, ние трябва да осредним върху всички възможни стойности на ненаблюдаемите данни Z, претегляйки всяко от тях съгласно тяхната вероятност. С други думи трябва да вземем очакваната стойност $E[\ln P(Y|h')]$ върху вероятностното разпределение, определящо случайната променлива Y. Това разпределение се определя от напълно известни стойности на X плюс от разпределение на Z.

Какво е вероятностното разпределение на Y? В общия случай ние няма да знаем това разпределение, тъй като то се определя от параметрите θ, които опитваме се да оценим. По тази причина алгоритъм EM използва текущата хипотеза h вместо действителните стойности на тези параметри, за да оцени вероятностното разпределение на Y. Нека да

определим функцията $Q(h'|h)$, която представя $E[\ln P(Y|h')]$ като функция от h' при предположение, че $\theta = h$ и зададените наблюдавани данни X – част от Y:

$$Q(h'|h) = E[\ln P(Y|h')|h, X]$$

Ще записваме тази функция във вида $Q(h'|h)$, за да подчертаем, че тя се определя отчасти от предположението, че текущата хипотеза h е равна на θ. В своята обща форма алгоритъм ЕМ повтаря следните две стъпки докато не се получи сходимостта:

Стъпка 1. *Очакване (E):* Изчисли $Q(h'|h)$, използвайки текущата хипотеза h и наблюдаваните данни X, за да оцени вероятностното разпределение на Y.

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

Стъпка 2. *Максимизация (M):* Замени хипотеза h с хипотеза h', която максимизира функцията Q:

$$h \leftarrow \arg\max_{h'} Q(h'|h)$$

Когато функцията Q е непрекъсната, алгоритъм ЕМ има сходимост към една стационарна точка на вероятностната функция $P(Y|h')$. Когато тази вероятностна функция има само един максимум, ЕМ има сходимост към този глобален максимум в оценката на h'. В противен случай алгоритмът гарантира само сходимостта към някой локален максимум. От тази гледна точка алгоритъм ЕМ има същите ограничения както и други оптимизационни методи, например методът на градиентното спускане.