Байесовские сети доверия

Сергей Николенко

Машинное обучение — ИТМО, осень 2006

Outline

1. Идея байесовских сетей
   - Мотивация
   - Пример
2. Основные понятия
   - d-разделимость
   - Теорема о декомпозиции
   - Свидетельства
3. Пропагация в сетях без циклов
   - Идея
   - Вывод алгоритма
   - Алгоритм

Байесовские сети

В этой и следующей лекциях мы на время отвлечёмся от обучения и рассмотрим основные понятия и алгоритмы теории байесовских сетей доверия. Затем мы научимся обучать такие сети.

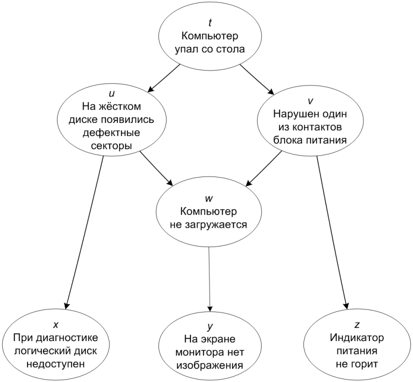
Наивный байесовский классификатор

- Наивный байесовский классификатор часто хорошо работает.
- Но он основывается на очень серьёзном предположении, а именно на условной независимости атрибутов при условии данного целевого значения.
- Зачастую такое предположение делает аппарат неприменимым.
- Что делать?

От байесовского классификатора к байесовским сетям

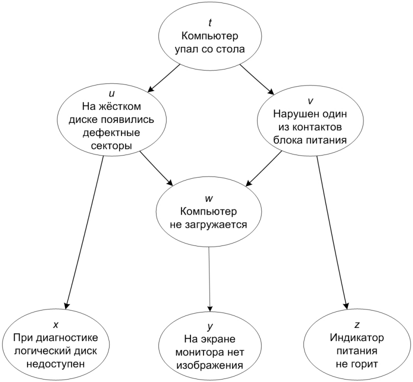
- Нужно научиться представлять множество (не)зависимостей между имеющимися переменными.
- Достаточно естественная идея: направленный граф, в котором стрелки показывают причинно-следственную связь.

Пример



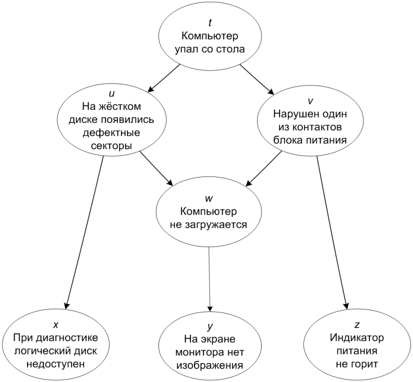
Пример того, как может быть задан граф байесовской сети. Истоки — причины, стоки — следствия. Корни полидерева — первопричины, листья — симптомы (как правило, именно их наблюдают).

Пример



Изначально заданы условные вероятности потомков при условии предков. Это уже позволит определить совместную априорную вероятность любой комбинации событий в сети.

Пример



Суть рассуждений в байесовской сети — *пропагация свидетельств*. На вход поступают данные типа «Логический диск недоступен» и «Индикатор питания горит», а задача — оценить, как изменилась вероятность других узлов (например, того, что «Компьютер упал со стола»).

Применение

Байесовские сети применяются в основном для решения *диагностических* задач. Например, их часто используют в медицине и вообще для оценки рисков. Т.е. пропагация обычно идёт снизу вверх, от следствий к причинам. Но можно и наоборот.

Outline

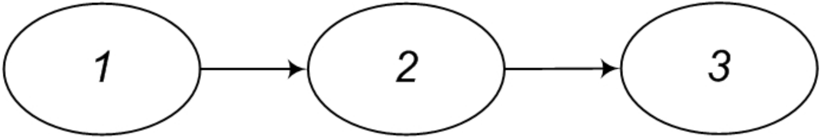
1. Идея байесовских сетей
   - Мотивация
   - Пример
2. Основные понятия
   - d-разделимость
   - Теорема о декомпозиции
   - Свидетельства
3. Пропагация в сетях без циклов
   - Идея
   - Вывод алгоритма
   - Алгоритм

Зависимость в БСД

- Какие узлы (события) в байесовской сети зависимы?
- Понятно, что те, которые соединены ребром.
- Но не только...

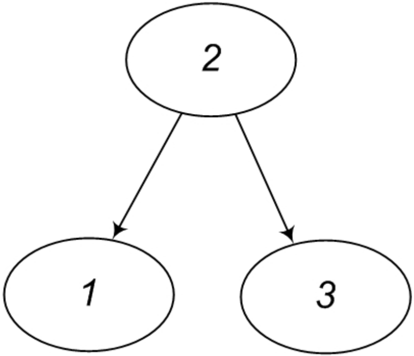
Последовательная связь

Прямая связь между узлами сети. В этом случае 1 влияет на 2, а 2, в свою очередь, влияет на 2, и узлы 1 и 3 получаются связанными. Однако, если в 2 поступило свидетельство, связь между 1 и 3 нарушается/



Последовательная связь

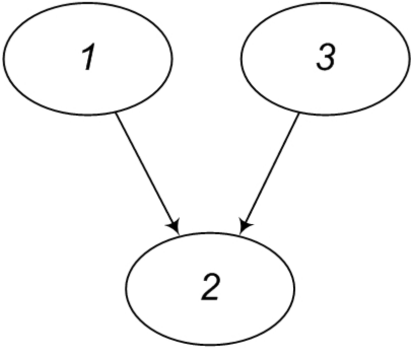
Расходящаяся связь



Расходящаяся связь

Информация об одном из потомков может повлиять на вероятность другого потомка одного и того же узла. Это связано с тем, что не только информация о произошедшей причине повышает вероятность следствия, но и случившееся следствие повышает вероятность причины. Если общий предок уже получил означивание, то связь нарушается.

Сходящаяся связь



Сходящаяся связь

Связи между узлами 1 и 3 нет: если произошло 1, то это повлияет на вероятность события 2, но вероятность 3 измениться не должна.

Однако ситуация меняется, если свидетельство 2 уже получено: если мы знаем, что одна из причин произошла, это должно понизить вероятность другой причины, ведь следствие уже объяснено.

d-разделимость

Определение

Два узла направленного графа x и y называются d-разделенными, если для всякого пути из x в y (здесь не учитывается направление ребер) существует такой промежуточный узел z (не совпадающий ни с x, ни с y), что либо связь в пути в этом узле последовательная или расходящаяся, и узел z получил означивание, либо связь сходящаяся, и ни узел z, ни какой-либо из его потомков означивания не получил.
В противном случае узлы называются d-связанными.

Упражнения

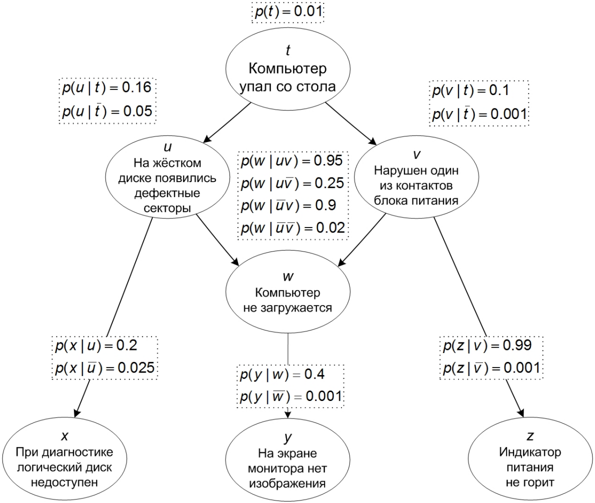
Упражнение

Написать процедуру, которая по заданному графу (формат задания графа произвольный) и двум вершинам определяет, d-разделены они или нет.

Вероятности

- До сих пор были только графы и рассуждения «на пальцах».
- Теперь пора перейти к заданию вероятностей и прочим численным примерам.
- В вершинах заданы условные вероятности при условии всего множества предков. Если предков нет, вероятности не условные (а маргинальные).

Пример


$p(t) = 0.01$
t
Компьютер упал со стола
$p(u \mid t) = 0.16$
$p(u \mid \bar{t}) = 0.05$
$p(v \mid t) = 0.1$
$p(v \mid \bar{t}) = 0.001$
u
На жёстком диске появились дефектные секторы
$p(w \mid uv) = 0.95$
$p(w \mid u\bar{v}) = 0.25$
$p(w \mid \bar{u}v) = 0.9$
$p(w \mid \bar{u}\bar{v}) = 0.02$
v
Нарушен один из контактов блока питания
w
Компьютер не загружается
$p(x \mid u) = 0.2$
$p(x \mid \bar{u}) = 0.025$
$p(y \mid w) = 0.4$
$p(y \mid \bar{w}) = 0.001$
$p(z \mid v) = 0.99$
$p(z \mid \bar{v}) = 0.001$
x
При диагностике логический диск недоступен
y
На экране монитора нет изображения
z
Индикатор питания не горит

Численный пример

Вычислим совместные вероятности цепочки $\tilde{u}\tilde{v}\tilde{w}$:

$$
\begin{aligned}
p(uvw) &= p(w|uv)\sum_{\tilde{t}} p(u|\tilde{t})p(v|\tilde{t})p(\tilde{t}) = 0.000199025,\\
p(uv\bar{w}) &= p(\bar{w}|uv)\sum_{\tilde{t}} p(u|\tilde{t})p(v|\tilde{t})p(\tilde{t}) = 0.000010475,\\
p(u\bar{v}w) &= p(w|u\bar{v})\sum_{\tilde{t}} p(u|\tilde{t})p(\bar{v}|\tilde{t})p(\tilde{t}) = 0.012722625,\\
p(u\bar{v}\bar{w}) &= p(\bar{w}|u\bar{v})\sum_{\tilde{t}} p(u|\tilde{t})p(\bar{v}|\tilde{t})p(\tilde{t}) = 0.038167875,\\
p(\bar{u}vw) &= p(w|\bar{u}v)\sum_{\tilde{t}} p(\bar{u}|\tilde{t})p(v|\tilde{t})p(\tilde{t}) = 0.00160245,\\
p(\bar{u}v\bar{w}) &= p(\bar{w}|\bar{u}v)\sum_{\tilde{t}} p(\bar{u}|\tilde{t})p(v|\tilde{t})p(\tilde{t}) = 0.00017805,\\
p(\bar{u}\bar{v}w) &= p(w|\bar{u}\bar{v})\sum_{\tilde{t}} p(\bar{u}|\tilde{t})p(\bar{v}|\tilde{t})p(\tilde{t}) = 0.01894239,\\
p(\bar{u}\bar{v}\bar{w}) &= p(\bar{w}|\bar{u}\bar{v})\sum_{\tilde{t}} p(\bar{u}|\tilde{t})p(\bar{v}|\tilde{t})p(\tilde{t}) = 0.92817711.
\end{aligned}
$$

Важные замечания

- Если у узла n предков, нужно задавать 2^n условных вероятностей.
- Если предков у узла x нет, нужно задавать маргинальные вероятности $p(x)$.
- В графе запрещены направленные циклы.
- Вся эта информация в сумме даст возможность вычислять любую вероятность в сети, т.е. единственным образом задаст распределение.

Теорема о декомпозиции

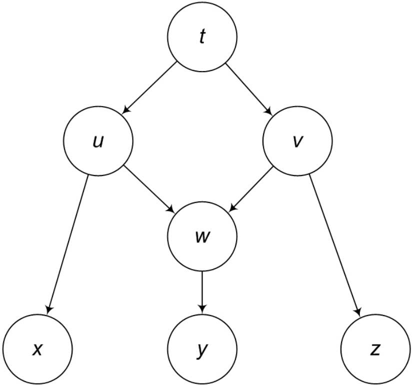
Теорема

Для БСД, построенной на множестве переменных $\mathcal{S} = \{x_1, x_2, \ldots, x_n\}$, общее распределение вероятностей $p(\tilde{x}_1 \tilde{x}_2 \ldots \tilde{x}_n)$, индуцирующее заданные условные и маргинальные вероятности и согласованное с условной независимостью, вытекающей из d-разделимости узлов, существует, единственно и представляет собой произведение всех тензоров, заданных в байесовской сети доверия:

$$p(\tilde{\mathcal{S}}) = \prod_{x \in \mathcal{S}} p(\tilde{x} | \widetilde{\mathrm{pa}(x)}),$$

где $\mathrm{pa}(x)$ *— множество родителей узла* x *в базовом графе сети.*

Пример

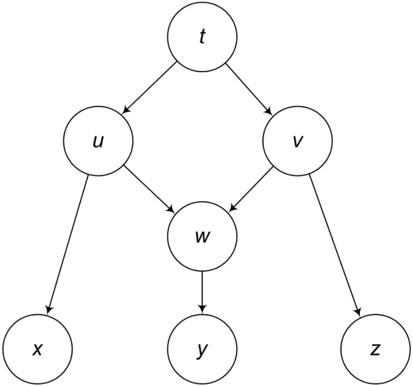


Попробуем в нашем примере вычислить $p(z)$. Для этого надо просуммировать по всем остальным вероятностям:

$$p(z) = \sum_{\tilde{t}\tilde{u}\tilde{v}\tilde{w}\tilde{x}\tilde{y}} p(\tilde{t}\,\tilde{u}\,\tilde{v}\,\tilde{w}\,\tilde{x}\,\tilde{y}\,z).$$

Тут безумное количество вычислений.

Пример

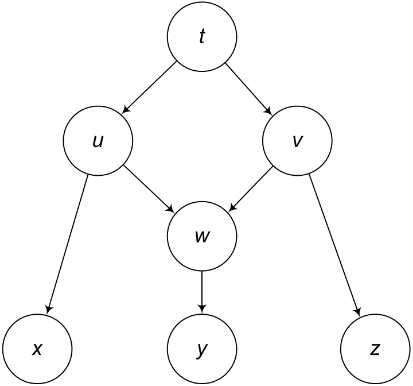


После применения правила декомпозиции получается:

$$p(\tilde{z}) = \sum_{\tilde{t}} p(\tilde{t}) \sum_{\tilde{v}} p(\tilde{v}|\tilde{t}) p(\tilde{z}|\tilde{v}) \sum_{\tilde{u}} p(\tilde{u}|\tilde{t}) \sum_{\tilde{x}} p(\tilde{x}|\tilde{u}) \sum_{\tilde{w}} p(\tilde{w}|\tilde{u}\tilde{v}) \sum_{\tilde{y}} p(\tilde{y}|\tilde{w}),$$

и вычислений уже гораздо меньше.

Пример



В этом и заключается смысл байесовских сетей доверия — разложить большое распределение на произведение маленьких.

Свидетельства

- Свидетельства — утверждения вида «событие в узле x произошло». Например: «Компьютер не загружается», т.е. w.
- Главная наша задача: научиться пересчитывать вероятности при поступлении свидетельств.

Пересчёт вероятностей в одном узле

Пусть поступило свидетельство «Компьютер не загружается». Давайте рассчитаем апостериорную вероятность $p(u|w)$. Сначала нужно приравнять нулю несовместимые со свидетельством случаи в таблице совместных вероятностей:

$$
\begin{aligned}
p(uvw \wedge w) &= 0.000199025, & p(uv\bar{w} \wedge w) &= 0,\\
p(u\bar{v}w \wedge w) &= 0.012722625, & p(u\bar{v}\bar{w} \wedge w) &= 0,\\
p(\bar{u}vw \wedge w) &= 0.00160245, & p(\bar{u}v\bar{w} \wedge w) &= 0,\\
p(\bar{u}\bar{v}w \wedge w) &= 0.01894239, & p(\bar{u}\bar{v}\bar{w} \wedge w) &= 0.
\end{aligned}
$$

Пересчёт вероятностей в одном узле

И нормировать то, что получилось:

$$p(u|w) = \frac{\sum_{\tilde{v}\tilde{w}} p(u\tilde{v}\tilde{w}\wedge w)}{p(w)} = 0.386107\ldots ,$$
$$p(\bar{u}|w) = \frac{\sum_{\tilde{v}\tilde{w}} p(\bar{u}\tilde{v}\tilde{w}\wedge w)}{p(w)} = 0.61389288\ldots .$$

Пересчёт вероятностей в одном узле

Вопрос: каким предположением мы неявно пользовались? Что делать, если оно не выполнено?

Outline

1. Идея байесовских сетей
   - Мотивация
   - Пример
2. Основные понятия
   - d-разделимость
   - Теорема о декомпозиции
   - Свидетельства
3. Пропагация в сетях без циклов
   - Идея
   - Вывод алгоритма
   - Алгоритм

Пропагация в сетях без циклов

- Мы сейчас запрещаем не только направленные, но и ненаправленные циклы. Т.е. граф байесовской сети — полидерево.
- Чтобы не запрещать ненаправленные циклы, нужно рассматривать алгоритмы построения морального графа, триангуляции и построения дерева смежности. Этим мы займёмся на следующей лекции.
- Мы уже поняли, как пропагировать свидетельство через один узел. Осталось формализовать это и собрать алгоритм целиком.

Идея алгоритма

- Мы делим поступившие свидетельства на две части — те, что выше данного узла x (E_x^+) и те, что ниже (E_x^-). Наша задача — найти

$$p(x|E) = p(x|E_x^-, E_x^+).$$

- Затем вычисляем их действие на x по отдельности и складываем это всё вместе.

Обозначения

- Evid(x) возвращает 1, если x инстанциирован нашим свидетельством.
- Normalize(Pr) нормализует распределение Pr.
- $E_{x\setminus Y}$ означает свидетельства, связанные с x, кроме тех, путь к которым пролегает через элементы множества Y.

Вывод

$$p(x|E_x^-, E_x^+) = \frac{p(E_x^-|x, E_x^+)p(x|E_x^+)}{p(E_x^-|E_x^+)}.$$

Но в полидереве x d-отделяет E_x^+ от E_x^-, поэтому $p(E_x^-|x, E_x^+) = p(E_x^-|x)$. Кроме того, вероятности должны в сумме давать 1, поэтому можно забыть про $\frac{1}{p(E_x^-|E_x^+)}$, а потом просто нормализовать результат. Итого нужно вычислить

$$p(E_x^-|x)p(x|E_x^+).$$

Начнём с $p(x|E_x^+)$.

Вывод

Рассмотрим все конфигурации родителей x $\tilde{U} = \widetilde{\mathrm{pa}(x)}$. Тогда

$$p(x|E_x^+) = \sum_{\tilde{U}} p(x|\tilde{U}, E_x^+)p(\tilde{U}|E_x^+).$$

- U d-отделяет x от E_x^+, поэтому первый сомножитель — это просто $p(x|\tilde{U})$, и это дано нам в таблицах условных вероятностей.
- E_x^+ d-отделяет каждый $u \in U$ от других, и

$$p(\tilde{U}|E_x^+) = \prod_{u \in U} p(\tilde{u}|E_X^+).$$

Вывод

Рассмотрим все конфигурации родителей x $\tilde{U} = \widetilde{\mathrm{pa}(x)}$. Тогда

$$p(x|E_x^+) = \sum_{\tilde{U}} p(x|\tilde{U}, E_x^+)p(\tilde{U}|E_x^+).$$

- Осталось заметить, что $E_x^+ = \bigcup_{u \in U} E_{u\setminus x}$, все они не пересекаются, и $E_{u\setminus x}$ d-отделяет u от всех остальных свидетельств в E_x^+. Итого получается:

$$p(x|E_x^+) = \mathrm{Norm}\left(\sum_{\tilde{U}} p(x|\tilde{U}) \prod_{u \in U} p(\tilde{u}|E_{u\setminus x})\right).$$

Вывод

Рассмотрим все конфигурации родителей x $\tilde{U} = \widetilde{\mathrm{pa}(x)}$. Тогда

$$p(x|E_x^+) = \sum_{\tilde{U}} p(x|\tilde{U}, E_x^+) p(\tilde{U}|E_x^+).$$

$$p(x|E_x^+) = \mathrm{Norm}\left(\sum_{\tilde{U}} p(x|\tilde{U}) \prod_{u \in U} p(\tilde{u}|E_{u\setminus x})\right).$$

- Это уже похоже на рекурсивный алгоритм: $p(\tilde{u}|E_{u\setminus x})$ — рекурсивный вызов исходной процедуры.

Вывод

Упражнение

Провести аналогичный вывод для $p(E_x^-|x)$. Нужно будет учитывать возможных других родителей потомков x. Должно получиться:

$$p(E_x^-|x) = \mathrm{Norm}\left(\prod_{u\in \mathrm{ch}(x)} \sum_{\tilde{u}} p(E_u^-|\tilde{u}) \times \right.$$
$$\left. \times \sum_{\widetilde{\mathrm{pa}(u)}} p(\tilde{u}|x, \widetilde{\mathrm{pa}(u)}) \prod_{z\in \mathrm{pa}(u)} p(\tilde{z}|E_{z\setminus u})\right)$$

(везде $\mathrm{pa}(u)$ не включает в себя x, разумеется).

Алгоритм

$\mathrm{Belief}(x)$:

- Вернуть $\mathrm{BeliefExcept}(x, \emptyset)$.

$\mathrm{BeliefExcept}(x, V)$:

- Если $\mathrm{Evid}(x)$, то вернуть уже имеющееся распределение x.
- Вычислить $p(E^-_{x\setminus V}|x) = \mathrm{EvidenceExcept}(X, V)$.
- $U = \mathrm{pa}(x)$.
- Если $U = \emptyset$ вернуть $\mathrm{Norm}\left(p(E^-_{x\setminus V}|x)p(x)\right)$.
- Иначе для каждого $u \in U$ вычислить и сохранить $p(\tilde{u}|E_{u\setminus x}) = \mathrm{BeliefExcept}(\tilde{u}, x)$.
- Вернуть $\mathrm{Norm}\left(p(E^-_{x\setminus V}|x)\sum_{\tilde{U}} p(x|\tilde{U})\prod_{u\in U} p(\tilde{u}|E_{u\setminus x})\right)$.

Алгоритм

$\mathrm{EvidenceExcept}(x, V)$:

- $U = \mathrm{ch}(x) \setminus V$.
- Если $(U == \emptyset)$ вернуть равномерное распределение.
- Иначе для каждого $u \in U$:
  - Вычислить $p(E_u^-|\tilde{u}) = \mathrm{EvidenceExcept}(\tilde{u}, \emptyset)$.
  - $Z = \mathrm{pa}(u) \setminus \{x\}$.
  - Для каждого $z \in Z$ вычислить

$$p(\tilde{Z}|E_{z\setminus u}) = \mathrm{BeliefExcept}(\tilde{w}, u).$$

- Вернуть

$$p(E_x^-|x) = \mathrm{Norm}\left(\prod_{u\in \mathrm{ch}(x)} \sum_{\tilde{u}} p(E_u^-|\tilde{u}) \sum_{\tilde{Z}} p(\tilde{u}|x, \tilde{Z}) \prod_{z\in Z} p(\tilde{z}|E_{z\setminus u})\right).$$

Упражнения

Упражнение

Реализовать алгоритм пропагации в байесовских сетях без циклов.

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