Спецсеминар «Байесовские методы машинного обучения», 2011

Boltzmann Machines

Дата: 5 октября 2011

Содержание

Ликбез: экспоненциальное семейство распределений

Распределение вероятностей $p(\boldsymbol{x})$ принадлежит экспоненциальному семейству распределений, если оно может быть представлено как

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(\boldsymbol{x}) \exp(\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{x})),$$

где $\boldsymbol{\theta}$ – набор параметров распределения, количество компонент вектора $\boldsymbol{u}(\boldsymbol{x})$ совпадает с размерностью $\boldsymbol{\theta}$, $h(\cdot)$ – некоторая функция, а $Z(\boldsymbol{\theta}) = \int h(\boldsymbol{x}) \exp(\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{x})) d\boldsymbol{x}$ – нормировочная константа распределения.

Многие стандартные вероятностные распределения принадлежат экспоненциальному семейству, например, нормальное, гамма, бета, Бернулли, Дирихле и многие другие. Соответствие между параметрами этих распределений и компонентами $\boldsymbol{\theta}, \boldsymbol{u}(\boldsymbol{x})$ в экспоненциальном представлении показано в таблице ниже.

Распределение	Плотность	$\boldsymbol{u}(\boldsymbol{x})$	$\boldsymbol{\theta}$
Бернулли	$q^x(1-q)^{1-x}$	x	$\log\frac{q}{1-q}$
Мультиномиальное	$\prod_k \mu_k^{x_k}$	$[x_1,\ldots,x_{K-1}]$	$\theta_i = \log\frac{\mu_i}{1-\sum_j \mu_j}$
Нормальное	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$[x, x^2]$	$\left[-\frac{1}{2\sigma}, \frac{\mu}{\sigma^2}\right]$
Гамма	$\frac{b^a}{\Gamma(a)}x^{a-1}\exp(-bx)$	$[\log x, x]$	$[a-1, -b]$
Бета	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1}$	$[\log(x), \log(1-x)]$	$[a-1, b-1]$
Пуассона	$\exp(-\lambda)\frac{\lambda^x}{x!}$	$[x, \log\Gamma(x+1)]$	$[k, -1]$

Легко показать, что для экспоненциального семейства градиент логарифма правдоподобия по параметрам $\boldsymbol{\theta}$ может быть вычислен как

$$\nabla_{\boldsymbol{\theta}} \log p(\hat{\boldsymbol{x}}|\boldsymbol{\theta}) = \boldsymbol{u}(\hat{\boldsymbol{x}}) - \mathbb{E}\boldsymbol{u}(\boldsymbol{x}),$$

где математическое ожидание берется по распределению $p(\boldsymbol{x}|\boldsymbol{\theta})$. Предположим далее, что у нас есть выборка $\hat{X} = \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N\}$ независимых объектов из распределения $p(\boldsymbol{x}|\boldsymbol{\theta})$. Тогда усредненный градиент логарифма правдоподобия по параметрам может быть вычислен как

$$\frac{1}{N}\nabla_{\boldsymbol{\theta}} \log p(\hat{X}|\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{u}(\hat{\boldsymbol{x}}_n) - \mathbb{E}\boldsymbol{u}(\boldsymbol{x}) = \mathbb{E}_{data}\boldsymbol{u}(\boldsymbol{x}) - \mathbb{E}_{model}\boldsymbol{u}(\boldsymbol{x}),$$

где $p_{data}(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x} - \hat{\boldsymbol{x}}_n)$ – выборочная плотность распределения, а $p_{model}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta})$.

Рассмотрим теперь экспоненциальное семейство для случая наблюдаемых переменных $\boldsymbol{x}$ и ненаблюдаемых переменных $\boldsymbol{t}$:

$$p(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}h(\boldsymbol{x}, \boldsymbol{t})\exp(\boldsymbol{\theta}^T\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{t})).$$

Пусть имеется выборка $\hat{X} = \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N\}$ и нас интересует градиент логарифма неполного правдоподобия по параметрам:

$$\frac{1}{N}\nabla_{\boldsymbol{\theta}} \log p(\hat{X}|\boldsymbol{\theta}) = \mathbb{E}_{data}\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{t}) - \mathbb{E}_{model}\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{t}), \tag{1}$$

где $p_{data}(\boldsymbol{t}, \boldsymbol{x}) = p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{\theta})p_{data}(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N} p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{\theta})\delta(\boldsymbol{x} - \hat{\boldsymbol{x}}_n)$, а $p_{model}(\boldsymbol{x}, \boldsymbol{t}) = p(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta})$.

Ликбез: схема Гиббса генерации выборки из распределения

Рассмотрим вероятностное распределение $p(\boldsymbol{t})$. Схема Гиббса позволяет сгенерировать выборку из этого многомерного распределения:

$$\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N \sim p(\boldsymbol{t}).$$

Данная выборка не является набором независимых величин, но при этом может быть использована для оценки вероятностных интегралов вида

$$\mathbb{E}_{\boldsymbol{t}} f(\boldsymbol{t}) = \int f(\boldsymbol{t})p(\boldsymbol{t})d\boldsymbol{t} \simeq \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{t}_n). \tag{2}$$

Рассмотрим шаг генерации по схеме Гиббса. Пусть на шаге n сгенерирована конфигурация $\boldsymbol{t}^n = \{t_1^n, \ldots, t_d^n\}$. Тогда генерация следующей точки выборки $\boldsymbol{t}^{n+1}$ происходит следующим образом:

$$\begin{aligned} t_1^{n+1} &\sim p(t_1|t_2^n, t_3^n, \ldots, t_d^n), \\ t_2^{n+1} &\sim p(t_2|t_1^{n+1}, t_3^n, t_4^n, \ldots, t_d^n), \\ t_3^{n+1} &\sim p(t_3|t_1^{n+1}, t_2^{n+1}, t_4^n, \ldots, t_d^n), \\ &\ldots \\ t_d^{n+1} &\sim p(t_d|t_1^{n+1}, t_2^{n+1}, \ldots, t_{d-1}^{n+1}). \end{aligned} \tag{3}$$

Здесь через $p(t_i|\boldsymbol{t}_{\setminus i})$ обозначено условное одномерное распределение значений i-ой компоненты при условии всех остальных. Таким образом, согласно схеме Гиббса генерация выборки из многомерного распределения заменяется на итерационную генерацию точек из одномерных распределений. По аналогии с методами одномерной оптимизации генерация выборки из одномерного распределения является существенно более простой задачей, чем генерация выборки из многомерного распределения.

В схеме Гиббса (3) следует обратить внимание на следующие моменты:

- При реализации схемы Гиббса на практике часто допускается следующая ошибка: вместо шага
$$t_p^{n+1} \sim p(t_p|t_1^{n+1}, \ldots, t_{p-1}^{n+1}, t_{p+1}^n, \ldots, t_d^n)$$
делается шаг
$$t_p^{n+1} \sim p(t_p|t_1^n, \ldots, t_{p-1}^n, t_{p+1}^n, \ldots, t_d^n),$$
т.е. в условие подставляются значения компонент только с предыдущей итерации. При таком подходе интересующее нас распределение $p(\boldsymbol{t})$ не будет инвариантным относительно введенной марковской цепи и, следовательно, генерируемая выборка не будет выборкой из распределения $p(\boldsymbol{t})$.

- Порядок генерации компонент $t_1, \ldots, t_d$ может быть произвольным, но не должен меняться на разных итерациях. При изменении порядка генерации компонент на итерациях соответствующая марковская цепь перестает быть однородной, а, следовательно, необходимо отдельно доказывать эргодичность такой цепи (т.е. сходимость к единственному инвариантному распределению из любого начального распределения). В однородном случае для эргодичности достаточно, чтобы все условные вероятности были строго больше нуля.

- В схеме Гиббса необязательно рассматривать только одномерные условные распределения. Здесь достаточно разбить множество компонент вектора $\boldsymbol{t}$ на набор непересекающихся подмножеств и для каждого такого подмножества уметь генерировать выборку из условного распределения $p(\boldsymbol{t}_i|\boldsymbol{t}_{\setminus i})$, где $\boldsymbol{t}_i$ – множество компонент из i-го подмножества.

Ликбез: вариационный подход

Пусть имеется вероятностная модель $p(\boldsymbol{x}, \boldsymbol{t})$ с наблюдаемыми и ненаблюдаемыми переменными. Пусть также имеется некоторое произвольное распределение $q(\boldsymbol{t})$. Тогда:

$$\log p(\boldsymbol{t}) = \underbrace{\int \log \frac{p(\boldsymbol{x}, \boldsymbol{t})}{q(\boldsymbol{t})} q(\boldsymbol{t}) d\boldsymbol{t}}_{\mathcal{L}(q)} + \mathrm{KL}(q||p(\boldsymbol{t}|\boldsymbol{x})). \tag{4}$$

Здесь через KL обозначена дивергенция Кульбака-Лейблера между двумя вероятностными распределениями. Эта дивергенция является мерой расстояния между двумя распределениями, т.к. она всегда неотрицательна и равна нулю тогда и только тогда, когда распределения тождественно совпадают.

На равенство (4) можно смотреть с нескольких точек зрения. Допустим, что апостериорное распределение $p(\boldsymbol{t}|\boldsymbol{x})$ не поддается вычислению (не вычисляется нормировочная константа), и мы хотим найти приближение $q(\boldsymbol{t})$ для распределения $p(\boldsymbol{t}|\boldsymbol{x})$. Будем искать это приближение путем минимизации KL-дивергенции между распределением $q(\boldsymbol{t})$ и $p(\boldsymbol{t}|\boldsymbol{x})$ в некотором семействе распределений $\mathcal{Q}$. Тогда из равенства выше следует, что

$$\mathrm{KL}(q||p(\boldsymbol{t}|\boldsymbol{x})) \to \min_{q\in\mathcal{Q}} \Leftrightarrow \mathcal{L}(q) \to \max_{q\in\mathcal{Q}}.$$

Теперь задача минимизации, которая зависит от недоступного для вычисления распределения $p(\boldsymbol{t}|\boldsymbol{x})$, сведена к задаче максимизации функционала $\mathcal{L}(q)$, который зависит от известного полного совместного распределения модели $p(\boldsymbol{x},\boldsymbol{t})$.

С другой точки зрения, нас может интересовать значение маргинального распределения $p(\boldsymbol{x})$ (нормировочной константы для распределения $p(\boldsymbol{t}|\boldsymbol{x})$). Так как $\mathrm{KL}(q||p(\boldsymbol{t}|\boldsymbol{x})) \geq 0$, то значение функционала $\mathcal{L}(q)$ является нижней границей для $\log p(\boldsymbol{x})$. Таким образом, решая задачу максимизации функционала $\mathcal{L}(q)$ в некотором семестве распределений $\mathcal{Q}$, мы одновременно получаем аналитическое приближение апостериорного распределения $p(\boldsymbol{t}|\boldsymbol{x})$ и нижнюю границу для обоснованности $\log p(\boldsymbol{x})$.

Рассмотрим решение задачи максимизации $\mathcal{L}(q)$ в семействе т.н. факторизованных распределений:

$$q(\boldsymbol{t}) = \prod_{j=1}^{J} q_j(\boldsymbol{t}_j).$$

Здесь множество переменных $\boldsymbol{t}$ разбито на непересекающиеся подмножества $\boldsymbol{t}_j$, причем $\cup_j \boldsymbol{t}_j = \boldsymbol{t}$, а $q_j(\boldsymbol{t}_j)$ – произвольное распределение в пространстве переменных $\boldsymbol{t}_j$. Таким образом, мы приходим к следующей задачи оптимизации:

$$\mathcal{L}(q) = \int \log \frac{p(\boldsymbol{x},\boldsymbol{t})}{\prod_j q_j(\boldsymbol{t}_j)} \prod_j q_j(\boldsymbol{t}_j) d\boldsymbol{t}_j \to \max_{q_1,\ldots,q_J}.$$

Рассмотрим решение этой задачи с помощью покоординатного подъема, т.е. зафиксируем все компоненты распределения q, кроме q_i, и рассмотрим оптимизацию $\mathcal{L}(q)$ по отдельной компоненте $q_i(\boldsymbol{t}_i)$. Оказывается, что решение такой (вариационной) задачи оптимизации можно получить аналитически:

$$q_i^{new}(\boldsymbol{t}_i) = \frac{\exp\left(\int \log p(\boldsymbol{x},\boldsymbol{t}) \prod_{j\neq i} q_j^{old}(\boldsymbol{t}_j) d\boldsymbol{t}_j\right)}{\int \exp\left(\int \log p(\boldsymbol{x},\boldsymbol{t}) \prod_{j\neq i} q_j^{old}(\boldsymbol{t}_j) d\boldsymbol{t}_j\right) d\boldsymbol{t}_i}. \tag{5}$$

Эта формула является основным результатом вариационного подхода. Заметим, что оптимальное распределение $q_i(\boldsymbol{t}_i)$ зависит от всех остальных распределений $q_j(\boldsymbol{t}_j)$ для $j \neq i$. Поэтому при применении вариационного подхода возникает итерационная схема, в которой последовательно пересчитываются отдельные компоненты факторизованного распределения $q(\boldsymbol{t})$. При этом на каждом шаге итерации происходит монотонное увеличение нижней границы $\mathcal{L}(q)$ для обоснованности $\log p(\boldsymbol{x})$, а итерационная оптимизация проводится до сходимости по значению $\mathcal{L}(q)$. Заметим также, что в построениях выше не накладывалось никаких ограничений на семейство распределений $q(\boldsymbol{t})$, кроме факторизации.

Алгоритм 1 Оценка отношения нормировочных констант двух распределений с помощью схемы Гиббса

Вход: Ненормированные распределения $\tilde{p}_A(\boldsymbol{t})$ и $\tilde{p}_B(\boldsymbol{t})$.
Выход: Оценка отношения нормировочных констант Z_B/Z_A.

1: Построить последовательность распределений $\tilde{p}_k(\boldsymbol{t}) = [\tilde{p}_A(\boldsymbol{t})]^{1-\beta_k}[\tilde{p}_B(\boldsymbol{t})]^{\beta_k}$ для набора значений $0 = \beta_1 < \beta_2 < \cdots < \beta_{K-1} < \beta_K = 1$.
2: **для** $m = 1, \ldots, M$
3: Сгенерировать $\boldsymbol{t}^1$ из распределения $p_1(\boldsymbol{t})$;
4: Сделать шаг по схеме Гиббса $\boldsymbol{t}^2 \sim T_2(\boldsymbol{t}, \boldsymbol{t}^1)$ генерации из распределения $p_2(\boldsymbol{t})$;
5: ...
6: Сделать шаг по схеме Гиббса $\boldsymbol{t}^{K-1} \sim T_{K-1}(\boldsymbol{t}, \boldsymbol{t}^{K-2})$ генерации из распределения $p_{K-1}(\boldsymbol{t})$;
7: $w_m = \prod_{k=1}^{K-1}(\tilde{p}_{k+1}(\boldsymbol{t}^k)/\tilde{p}_k(\boldsymbol{t}^k))$;
8: $Z_B/Z_A \simeq \frac{1}{M}\sum_{m=1}^{M} w_m$.

Ликбез: оценка нормировочной константы распределения с помощью схемы Гиббса

Предположим, что у нас имеется вероятностное распределение, известное с точностью до нормировочной константы $p(\boldsymbol{t}) = \tilde{p}(\boldsymbol{t})/Z$. Как было отмечено выше, схема Гиббса позволяет сгенерировать выборку из этого распределения $\boldsymbol{t}^1, \ldots, \boldsymbol{t}^N$, которая затем может быть использована для оценки статистики распределения $f(\boldsymbol{t})$ по формуле (2).

Рассмотрим задачу оценки нормировочной константы распределения Z. Эта нормировочная константа играет роль обоснованности модели и может быть использована для сравнения различных вероятностных моделей между собой, а также для оценки вероятности $p(\boldsymbol{t})$ для тестовых объектов. Нормировочная константа является «нулевой статистикой» распределения и поэтому не может быть оценена с помощью формулы (2).

Предположим, что у нас имеется два распределения $p_A(\boldsymbol{t}) = \tilde{p}_A(\boldsymbol{t})/Z_A$ и $p_B(\boldsymbol{t}) = \tilde{p}_B(\boldsymbol{t})/Z_B$. Тогда отношение двух нормировочных констант можно оценить по следующей схеме:

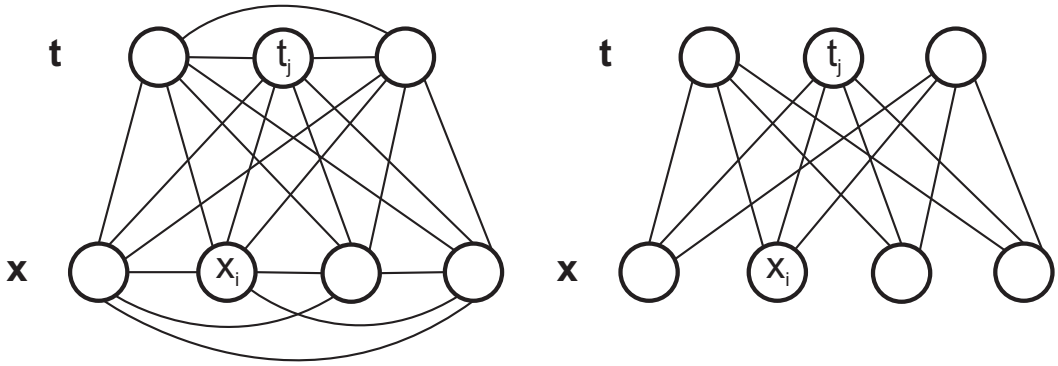
$$\frac{Z_B}{Z_A} = \frac{\int \tilde{p}_B(\boldsymbol{t})d\boldsymbol{t}}{Z_A} = \int \frac{\tilde{p}_B(\boldsymbol{t})}{Z_A}d\boldsymbol{t} = \int \frac{\tilde{p}_B(\boldsymbol{t})}{\tilde{p}_A(\boldsymbol{t})}p_A(\boldsymbol{t})d\boldsymbol{t} \simeq \frac{1}{M}\sum_{m=1}^{M}\frac{\tilde{p}_B(\boldsymbol{t}^m)}{\tilde{p}_A(\boldsymbol{t}^m)}. \tag{6}$$

Здесь $\boldsymbol{t}^1, \ldots, \boldsymbol{t}^M$ – выборка из распределения $p_A(\boldsymbol{t})$, которую можно сгенерировать, например, по схеме Гиббса. Если у распределения $p_A(\boldsymbol{t})$ нормировочная константа известна, то тогда мы можем оценить абсолютное значение Z_B.

К сожалению, схема (6) применима только в случае, когда распределение $p_A(\boldsymbol{t})$ является хорошим приближением для $p_B(\boldsymbol{t})$. На практике поиск хорошего аналитического приближения для интересующего нас распределения $p_B(\boldsymbol{t})$ может оказаться очень трудной задачей. В этом случае можно построить серию промежуточных распределений $p_A = p_1, p_2, \ldots, p_{K-1}, p_K = p_B$ и оценить нормировочную константу Z_B из соотношения:

$$\frac{Z_B}{Z_A} = \frac{Z_K}{Z_1} = \frac{Z_2}{Z_1}\frac{Z_3}{Z_2}\cdots\frac{Z_{K-1}}{Z_{K-2}}\frac{Z_K}{Z_{K-1}}.$$

Здесь каждое отношение Z_{k+1}/Z_k оценивается по схеме (6) путем генерации выборки из распределения $p_k(\boldsymbol{t})$, а серия промежуточных распределений строится как $\tilde{p}_k(\boldsymbol{t}) = [\tilde{p}_A(\boldsymbol{t})]^{1-\beta_k}[\tilde{p}_B(\boldsymbol{t})]^{\beta_k}$ для некоторого набора значений $0 = \beta_1 < \beta_2 < \cdots < \beta_{K-1} < \beta_K = 1$.



(a) (b)

Рис. 1: Граф марковской сети для Boltzmann Machine (a) и для Restricted Boltzmann Machine (b).

Предположим, что для каждого промежуточного распределения $p_k(\boldsymbol{t}) = \tilde{p}_k(\boldsymbol{t})/Z_k$, известного с точностью до нормировочной константы, мы можем применить схему Гиббса генерации выборки из этого распределения. Обозначим один шаг такой схемы Гиббса через $\boldsymbol{t}_{next} \sim T_k(\boldsymbol{t}, \boldsymbol{t}_{pred})$. Тогда итоговую схему оценки нормировочной константы Z_B можно представить как Алгоритм 1.

Boltzmann Machine

Модель Boltzmann Machine (BM) [1] представляет собой марковскую сеть с двумя слоями (см. рис. 1,a). В нижнем слое стоят наблюдаемые переменные $\boldsymbol{x}$, в верхнем слое – ненаблюдаемые переменные $\boldsymbol{t}$. Граф сети является полносвязным. Все переменные являются бинарными $x_i \in \{0, 1\}, t_j \in \{0, 1\}$. Энергию введенной марковской сети можно записать как

$$E(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta}) = -\boldsymbol{t}^T W \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T L \boldsymbol{x} - \frac{1}{2}\boldsymbol{t}^T J \boldsymbol{t}.$$

Здесь $\boldsymbol{\theta} = \{W, L, J\}$, $L_{ii} = J_{jj} = 0$. Как правило, в энергию $E(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta})$ добавляются также унарные потенциалы $-\boldsymbol{x}^T\boldsymbol{a} - \boldsymbol{t}^T\boldsymbol{b}$. Однако, для упрощения дальнейших выкладок унарные потенциалы будут опускаться. Энергия $E(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta})$ задает вероятностное распределение

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{t}} \exp(-E(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta})),$$

где $Z(\boldsymbol{\theta}) = \int \exp(-E(\boldsymbol{x}, \boldsymbol{t}|\boldsymbol{\theta}))d\boldsymbol{x}d\boldsymbol{t}$ – нормировочная константа распределения.

Рассмотрим задачу настройки параметров BM по выборке $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ с помощью метода максимального правдоподобия:

$$p(X|\boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}}.$$

Распределение $p(\boldsymbol{x}|\boldsymbol{\theta})$ принадлежит экспоненциальному семейству распределений. Поэтому мы

Алгоритм 2 Алгоритм обучения BM

Вход: Выборка $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
Выход: Параметры BM $\boldsymbol{\theta} = \{W, L, J\}$

Инициализация параметров $\boldsymbol{\theta}_0$ случайным образом.
для $iter = 1, \ldots, \#iter$
  для $n = 1, \ldots, N$
    Сгенерировать выборку $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_M$ из распределения $p(\boldsymbol{t}|\boldsymbol{x}_n)$ по схеме Гиббса;
    Оценить достаточные статистики $\mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{x}_n^T) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{t}_m \boldsymbol{x}_n^T$, $\mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{t}^T)$;
  Сгенерировать выборку $(\boldsymbol{t}_1, \boldsymbol{x}_1), \ldots, (\boldsymbol{t}_M, \boldsymbol{x}_M)$ из распределения $p(\boldsymbol{t}, \boldsymbol{x})$ по схеме Гиббса;
  Оценить достаточные статистики $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{x}^T) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{t}_m \boldsymbol{x}_m^T$, $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{t}^T)$, $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)$;
  Сделать шаг по градиенту:

$$W_{iter} = W_{iter-1} + \alpha_{iter}^W \left(\sum\nolimits_{n=1}^{N} \mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{x}_n^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{x}^T) \right),$$
$$J_{iter} = J_{iter-1} + \alpha_{iter}^J \left(\sum\nolimits_{n=1}^{N} \mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{t}^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{t}^T) \right),$$
$$L_{iter} = L_{iter-1} + \alpha_{iter}^L \left(\sum\nolimits_{n=1}^{N} (\boldsymbol{x}_n \boldsymbol{x}_n^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T) \right).$$

можем воспользоваться общим результатом (1) и записать:

$$\frac{1}{N}\nabla_W \log p(X|\boldsymbol{\theta}) = \mathbb{E}_{data}(\boldsymbol{t}\boldsymbol{x}^T) - \mathbb{E}_{model}(\boldsymbol{t}\boldsymbol{x}^T),$$
$$\frac{1}{N}\nabla_J \log p(X|\boldsymbol{\theta}) = \mathbb{E}_{data}(\boldsymbol{t}\boldsymbol{t}^T) - \mathbb{E}_{model}(\boldsymbol{t}\boldsymbol{t}^T),$$
$$\frac{1}{N}\nabla_L \log p(X|\boldsymbol{\theta}) = \mathbb{E}_{data}(\boldsymbol{x}\boldsymbol{x}^T) - \mathbb{E}_{model}(\boldsymbol{x}\boldsymbol{x}^T).$$

Следовательно, задачу максимизации правдоподобия в модели BM можно решать с помощью градиентного подъема, в котором на каждом шаге необходимо оценивать значение градиента путем оценки статистик по данным и по модели. Эти статистики можно оценить с помощью схемы Гиббса. Для применения схемы Гиббса необходимо уметь генерировать выборку из одномерных условных распределений вида $p(x_i|\boldsymbol{t}, \boldsymbol{x}_{\setminus i})$ и $p(t_j|\boldsymbol{x}, \boldsymbol{t}_{\setminus j})$. Эти распределения для BM могут быть найдены аналитически:

$$p(x_i = 1|\boldsymbol{t}, \boldsymbol{x}_{\setminus i}) = \frac{1}{1 + \exp(-\sum_j W_{ij} t_j - \sum_k L_{ki} x_i)},$$
$$p(t_j = 1|\boldsymbol{x}, \boldsymbol{t}_{\setminus j}) = \frac{1}{1 + \exp(-\sum_i W_{ij} x_i - \sum_k J_{kj} t_k)}.$$

Рассмотренная схема обучения BM представлена в алгоритме 2. Данный алгоритм был предложен в начале 80-х годов в работе [2]. К сожалению, на практике данный алгоритм не используется, т.к. он требует очень большого времени работы.

Значительное улучшение рассмотренного алгоритма было предложено в начале 90-х годов в работе [3] (см. Алгоритм 3). В этом варианте для оценки статистик по каждому из распределений предлагается запускать несколько т.н. продолжающихся (persistent) цепей Гиббса. При этом на каждой итерации делается всего один или несколько шагов по каждой из схем Гиббса, а шаг движения по градиенту выбирается маленьким. Здесь предполагается, что при небольшом изменении параметров $\boldsymbol{\theta}$ условные распределения $p(\boldsymbol{t}|\boldsymbol{x}_n, \boldsymbol{\theta})$ и совместное распределение

Алгоритм 3 Улучшенный алгоритм обучения BM с помощью продолжающихся цепей Гиббса

Вход: Выборка $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
Выход: Параметры BM $\boldsymbol{\theta} = \{W, L, J\}$

1: Инициализация параметров $\boldsymbol{\theta}_0$ случайным образом;
2: Инициализация M общих цепей Гиббса и M цепей Гиббса для каждого объекта обучения $(\boldsymbol{t}_0^{(0),1}, \boldsymbol{x}_0^1), \ldots, (\boldsymbol{t}_0^{(0),M}, \boldsymbol{x}_0^M), \boldsymbol{t}_0^{(1),1}, \ldots, \boldsymbol{t}_0^{(N),M}$;
3: **для** $iter = 1, \ldots, \#iter$
4: **для** $n = 1, \ldots, N$
5: Сделать шаг по схеме Гиббса для каждой из M цепей для объекта n и получить $\boldsymbol{t}_{iter}^{(n),1}, \ldots, \boldsymbol{t}_{iter}^{(n),M}$;
6: Оценить достаточные статистики $\mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{x}_n^T) = \frac{1}{M}\sum_{m=1}^M \boldsymbol{t}_{iter}^{(n),m}\boldsymbol{x}_n^T$, $\mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{t}^T)$;
7: Сделать шаг по схеме Гиббса для каждой из M общих цепей и получить $(\boldsymbol{t}_{iter}^{(0),1}, \boldsymbol{x}_{iter}^1), \ldots, (\boldsymbol{t}_{iter}^{(0),M}, \boldsymbol{x}_{iter}^M)$;
8: Оценить достаточные статистики $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{x}^T) = \frac{1}{M}\sum_{m=1}^M \boldsymbol{t}_{iter}^{(0),m}(\boldsymbol{x}_{iter}^m)^T$, $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{t}^T)$, $\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)$;
9: Сделать шаг по градиенту:

$$W_{iter} = W_{iter-1} + \alpha_{iter}^W \left(\sum\nolimits_{n=1}^N \mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{x}_n^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{x}^T)\right),$$
$$J_{iter} = J_{iter-1} + \alpha_{iter}^J \left(\sum\nolimits_{n=1}^N \mathbb{E}_{\boldsymbol{t}|\boldsymbol{x}_n}(\boldsymbol{t}\boldsymbol{t}^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{t}\boldsymbol{t}^T)\right),$$
$$L_{iter} = L_{iter-1} + \alpha_{iter}^L \left(\sum\nolimits_{n=1}^N (\boldsymbol{x}_n\boldsymbol{x}_n^T)/N - \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)\right).$$

$p(\boldsymbol{t}, \boldsymbol{x}|\boldsymbol{\theta})$ незначительно отличаются от инвариантных распределений цепей Гиббса. Поэтому достаточно сделать лишь небольшое число шагов по каждой из цепей Гиббса, чтобы достигнуть новых инвариантных распределений.

Алгоритм 3 используется на практике для обучения небольших BM. При переходе к BM, которые представляют практический интерес во многих задачах, описанный алгоритм работает слишком долго. Дальнейшие попытки улучшения алгоритма обучения BM связаны с рассмотрением упрощенной модели – Restricted Boltzmann Machine.

Restricted Boltzmann Machine

Модель Restricted Boltzmann Machine [4] полностью повторяет модель Boltzmann Machine за тем исключением, что в графе марковской сети отсутствуют связи на одном уровне (см. рис. 1,b). Энергия и вероятностное распределение такой сети могут быть записаны как

$$E(\boldsymbol{x}, \boldsymbol{t}|W) = -\boldsymbol{t}^T W \boldsymbol{x}, \quad p(\boldsymbol{x}|W) = \frac{1}{Z(W)}\sum_{\boldsymbol{t}} \exp(-E(\boldsymbol{x}, \boldsymbol{t}|W)).$$

Это вероятностное распределение принадлежит экспоненциальному семейству. Поэтому задачу обучения RBM по данным можно решать с помощью градиентного подъема, где градиент попрежнему вычисляется как

$$\frac{1}{N}\nabla_W \log p(X|W) = \mathbb{E}_{data}(\boldsymbol{t}\boldsymbol{x}^T) - \mathbb{E}_{model}(\boldsymbol{t}\boldsymbol{x}^T).$$

Ключевое отличие RBM от BM состоит в том, что скрытые переменные $\boldsymbol{t}$ являются условно независимыми при известных $\boldsymbol{x}$ и наоборот: переменные $\boldsymbol{x}$ являются условно независимыми

Алгоритм 4 Алгоритм обучения RBM по схеме 1-Contrastive Divergence

Вход: Выборка $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
Выход: Параметры RBM W

1: Инициализация параметров W случайным образом;
2: **для** $iter = 1, \ldots, \#iter$
3: **для** $n = 1, \ldots, N$
4: $\boldsymbol{t}_n \sim p(\boldsymbol{t}|\boldsymbol{x}_n)$;
5: $\boldsymbol{x}_n^{pred} \sim p(\boldsymbol{x}|\boldsymbol{t}_n)$;
6: $\boldsymbol{t}_n^{pred} \sim p(\boldsymbol{t}|\boldsymbol{x}_n^{pred})$;
7: Оценить необходимые достаточные статистики $\mathbb{E}_{data}(\boldsymbol{t}\boldsymbol{x}^T) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{t}_n \boldsymbol{x}_n^T$, $\mathbb{E}_{model}(\boldsymbol{t}\boldsymbol{x}^T) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{t}_n^{pred}(\boldsymbol{x}_n^{pred})^T$;
8: Сделать шаг по градиенту: $W_{iter} = W_{iter-1} + \alpha_{iter}(\mathbb{E}_{data}(\boldsymbol{t}\boldsymbol{x}^T) - \mathbb{E}_{model}(\boldsymbol{t}\boldsymbol{x}^T))$.

при известных $\boldsymbol{t}$. Соответствующие вероятностные распределения могут быть вычислены аналитически:

$$p(\boldsymbol{t}|\boldsymbol{x}, W) = \prod_j p(t_j|\boldsymbol{x}, W), \;\; p(t_j = 1|\boldsymbol{x}, W) = \frac{1}{1 + \exp(-\sum_i W_{ij} x_i)},$$
$$p(\boldsymbol{x}|\boldsymbol{t}, W) = \prod_i p(x_i|\boldsymbol{t}, W), \;\; p(x_i = 1|\boldsymbol{t}, W) = \frac{1}{1 + \exp(-\sum_j W_{ij} t_j)}.$$

Таким образом, генерация выборки из распределений $p(\boldsymbol{t}|\boldsymbol{x})$ и $p(\boldsymbol{x}|\boldsymbol{t})$ в модели RBM не требует привлечения затратной по времени схемы Гиббса и может быть осуществлена непосредственно. Это обстоятельство позволяет предложить эффективный алгоритм обучения RBM, представленный как Алгоритм 4. В этом алгоритме для оценки статистик по данным и по модели на каждой итерации осуществляется всего один шаг генерации. При таком подходе оценка обеих статистик будет обладать большой дисперсией. Однако, оцениваемый градиент представляет собой разность двух статистик. По утверждению авторов алгоритма 4, шум, который появляется при оценке обеих статистик, имеет в значительной степени одну и ту же природу и взаимно уничтожается при вычитании статистик. Тем не менее, для более точной сходимости к оптимуму, авторы метода рекомендуют увеличивать число шагов генерации на каждом шаге итерационного процесса по мере приближения к оптимуму.

Несмотря на наличие эффективного алгоритма обучения, модель RBM редко используется сама по себе в практических задачах, т.к. является слишком простой и не позволяет описывать данные со сложными взаимосвязями. Вместо этого на практике используется модель Deep Boltzmann Machine, которая, с одной стороны, является значительным усложнением RBM для моделирования сложных экспериментальных данных, а, с другой стороны, имеет относительно эффективный алгоритм обучения.

Deep Boltzmann Machine

Модель Deep Boltzmann Machine [1] представляет собой марковскую сеть, граф которой показан на рис. 2, справа. В этом графе есть несколько слоев, нижний слой соответствует наблюдаемым переменным $\boldsymbol{x}$, остальные – ненаблюдаемым. Связи в графе есть только между переменными соседних слоев. Энергия модели DBM с тремя слоями скрытых переменных выглядит следую-

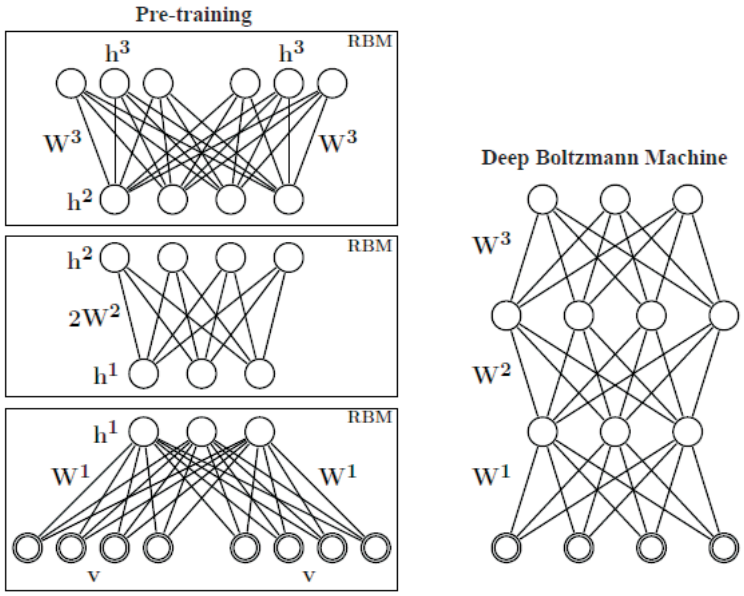


Рис. 2: Слева: схема алгоритма предобучения для модели DBM, справа: граф марковской сети для DBM.

щим образом:

$$E(\boldsymbol{x}, \boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3 | W^1, W^2, W^3) = -\boldsymbol{x}^T W^1 \boldsymbol{t}^1 - (\boldsymbol{t}^1)^T W^2 \boldsymbol{t}^2 - (\boldsymbol{t}^2)^T W^3 \boldsymbol{t}^3. \tag{7}$$

Как и раньше, распределение вероятностей для введенной марковской сети принадлежит экспоненциальному семейству:

$$p(\boldsymbol{x}|W^1, W^2, W^3) = \frac{1}{Z(W^1, W^2, W^3)} \sum_{\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3} \exp(-E(\boldsymbol{x}, \boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3 | W^1, W^2, W^3)).$$

Эффективный алгоритм обучения DBM [1] предлагает два основных новшества по сравнению с алгоритмом 3: 1) использование специальной процедуры предобучения параметров сети W с помощью многократного вызова процедуры обучения RBM, описанной выше, и 2) использование вариационного подхода вместо схем Гиббса для оценки достаточных статистик по данным.

Рассмотрим процедуру предобучения параметров DBM W (см. рис. 2). Веса первого слоя W^1 находятся путем обучения RBM, в которой нижний слой соответствует наблюдаемым переменным $\boldsymbol{x}$ в DBM, а верхний слой соответствует ненаблюдаемым переменным первого уровня $\boldsymbol{t}^1$. Затем значения $\boldsymbol{t}^1$ генерируются из обученной модели RBM $p(\boldsymbol{t}|\boldsymbol{x}, W^1)$ и используются в качестве наблюдаемых переменных в RBM для обучения весов следующего уровня W^2. Процедура повторяется вплоть до верхнего уровня. Одним из недостатков описанной процедуры является тот факт, что при обучении очередной RBM никак не учитывается влияние верхних уровней скрытых переменных на текущий слой скрытых переменных. Для частичной компенсации данного обстоятельства авторы метода предобучения предлагают следующую модификацию. В RBM для первого слоя наблюдаемые переменные дублируются, а обучение весов W^1 происходит при ограничении равенства весов у соответствующих связей между скрытыми и наблюдаемыми переменными. Аналогично, в RBM для последнего слоя дублируются скрытые

Алгоритм 5 Предобучение модели DBM

Вход: Выборка $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
Выход: Начальное приближение для весов DBM $W^1, \ldots, W^L$.

Продублировать узлы $\boldsymbol{x}$ на нижнем слое и обучить RBM (настроить W^1) для первого скрытого слоя при условии равенства весов W^1 для дублированных узлов;
Зафиксировать W^1 и сгенерировать выборку $\boldsymbol{t}^1$ из распределения $p(\boldsymbol{t}|\boldsymbol{x}, 2W^1)$;
Обучить RBM для второго слоя с наблюдаемыми данными $\boldsymbol{t}^1$ с весами $2W^2$;
...
При обучении RBM на последнем слое продублировать скрытые переменные и приравнять веса продублированных узлов;
Использовать веса $\{W^1, \ldots, W^L\}$ в качестве начального приближения для весов DBM.

переменные и обучение происходит при ограничении равенства весов у соответствующих связей между переменными. Результирующие веса для промежуточных слоев устанавливаются как половина от весов, полученных в результате обучения набора RBM. Итоговая схема предобучения DBM представлена в алгоритме 5.

Рассмотрим применение вариационного подхода для оценки достаточных статистик по данным при обучении DBM. Для этого рассмотрим приближение для апостериорного распределения $p(\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|\boldsymbol{x}, W^1, W^2, W^3)$ в полностью факторизованном семействе распределений:

$$q(\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \boldsymbol{\mu}^3) = \prod_j q(t_j^1|\mu_j^1) \prod_k q(t_k^2|\mu_k^2) \prod_m q(t_m^3|\mu_m^3).$$

Здесь $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \boldsymbol{\mu}^3$ — параметры распределения q, μ_l^i имеет смысл вероятности того, что $t_l^i = 1$. Наилучшее приближение q в выбранном семействе может быть найдено путем максимизации нижней границы для неполного правдоподобия $\log p(\boldsymbol{x}|W)$:

$$\log p(\boldsymbol{x}|W) \geq \mathcal{L}(q) = \boldsymbol{x}^T W^1 \boldsymbol{\mu}^1 + (\boldsymbol{\mu}^1)^T W^2 \boldsymbol{\mu}^2 + (\boldsymbol{\mu}^2)^T W^3 \boldsymbol{\mu}^3 - \log Z(W) + \mathcal{H}(q) \to \max_{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \boldsymbol{\mu}^3}. \quad (8)$$

Здесь через $\mathcal{H}(q)$ обозначена энтропия распределения q. Задача оптимизации (8) может быть решена с помощью итерационного процесса (5), который в данном случае выглядит как

$$\mu_j^1 = \frac{1}{1 + \exp(-\sum_i W_{ij}^1 x_i - \sum_k W_{jk}^2 \mu_k^2)},$$
$$\mu_k^2 = \frac{1}{1 + \exp(-\sum_j W_{jk}^2 \mu_j^1 - \sum_m W_{km}^3 \mu_m^3)},$$
$$\mu_m^3 = \frac{1}{1 + \exp(-\sum_k W_{km}^3 \mu_k^2)}.$$

После того, как параметры $\boldsymbol{\mu}$ определены, новые значения весов W можно найти путем движения вдоль градиента нижней границы (8)

$$\nabla_{W^1} \mathcal{L}(q) = \boldsymbol{x}(\boldsymbol{\mu}^1)^T - \nabla_{W^1} \log Z(W),$$
$$\nabla_{W^2} \mathcal{L}(q) = \boldsymbol{\mu}^1(\boldsymbol{\mu}^2)^T - \nabla_{W^2} \log Z(W),$$
$$\nabla_{W^2} \mathcal{L}(q) = \boldsymbol{\mu}^2(\boldsymbol{\mu}^3)^T - \nabla_{W^3} \log Z(W).$$

Здесь, как и раньше, градиенты нормировочной константы $Z(W)$ могут быть оценены с помощью продолжающихся цепей Гиббса по схеме алгоритма 3.

Объединяя вместе все вышесказанное, можно составить итоговую схему обучения DBM (см. Алгоритм 6).

Алгоритм 6 Обучение модели DBM

Вход: Выборка $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
Выход: Параметры DBM $W^1, \ldots, W^L$.

1: Предобучить параметры DBM $W^1, \ldots, W^L$ по алгоритму 5;
2: Инициализировать M цепей Гиббса случайным образом $(\boldsymbol{x}_0^1, \boldsymbol{t}_0^{1,1}, \ldots, \boldsymbol{t}_0^{L,1}), \ldots, (\boldsymbol{x}_0^M, \boldsymbol{t}_0^{1,M}, \ldots, \boldsymbol{t}_0^{L,M})$;
3: **для** $iter = 1, \ldots, \#iter$
4: **для** $n = 1, \ldots, N$
5: Сделать вариационное приближение для распределения $p(\boldsymbol{t}^1, \ldots, \boldsymbol{t}^L | \boldsymbol{x}_n)$ и найти его параметры $\{\boldsymbol{\mu}_n^1, \ldots, \boldsymbol{\mu}_n^L\}$;
6: **для** $m = 1, \ldots, M$
7: Сделать шаг по схеме Гиббса и получить $(\boldsymbol{x}_{iter}^m, \boldsymbol{t}_{iter}^{1,m}, \ldots, \boldsymbol{t}_{iter}^{L,m})$;
8: Сделать шаг по градиенту:

$$W_{iter}^1 = W_{iter-1}^1 + \alpha_{iter} \left[\left(\sum\nolimits_{n=1}^{N} \boldsymbol{x}_n (\boldsymbol{\mu}_n^1)^T \right) / N - \left(\sum\nolimits_{m=1}^{M} \boldsymbol{x}_{iter}^m (\boldsymbol{t}_{iter}^{1,m})^T \right) / M \right],$$

$$W_{iter}^2 = W_{iter-1}^2 + \alpha_{iter} \left[\left(\sum\nolimits_{n=1}^{N} \boldsymbol{\mu}_n^1 (\boldsymbol{\mu}_n^2)^T \right) / N - \left(\sum\nolimits_{m=1}^{M} \boldsymbol{t}_{iter}^{1,m} (\boldsymbol{t}_{iter}^{2,m})^T \right) / M \right],$$

$$\ldots$$

$$W_{iter}^L = W_{iter-1}^L + \alpha_{iter} \left[\left(\sum\nolimits_{n=1}^{N} \boldsymbol{\mu}_n^{L-1} (\boldsymbol{\mu}_n^L)^T \right) / N - \left(\sum\nolimits_{m=1}^{M} \boldsymbol{t}_{iter}^{L-1,m} (\boldsymbol{t}_{iter}^{L,m})^T \right) / M \right].$$

Использование Deep Boltzmann Machine

Deep Boltzmann Machine как порождающая модель

Рассмотрим вероятностную модель DBM как способ восстановления плотности в пространстве $\boldsymbol{x}$ по данным. В этом случае нам необходимо уметь оценивать значение плотности $p(\boldsymbol{x}|W)$ для обучающих и/или тестовых объектов, что в свою очередь требует оценки нормировочной константы распределения $Z(W)$. Для ее оценивания воспользуемся методом построения промежуточных распределений, описанный выше.

Пусть модель DBM имеет два слоя скрытых переменных. Следовательно, энергия системы определяется формулой (7). Суммируя по $\boldsymbol{t}^1$ и $\boldsymbol{t}^3$, получим ненормированную плотность в пространстве $(\boldsymbol{x}, \boldsymbol{t}^2)$

$$\tilde{p}(\boldsymbol{x}, \boldsymbol{t}^2|W) = \sum_{\boldsymbol{t}^1, \boldsymbol{t}^3} \tilde{p}(\boldsymbol{x}, \boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|W) = \prod_j \left(1 + \exp(\sum_i W_{ij}^1 x_i + \sum_k W_{jk}^2 t_k^2) \right) \prod_m \left(1 + \exp(\sum_k W_{km}^3 t_k^2) \right).$$

Для DBM с произвольным числом слоев мы всегда можем аналитически просуммировать по всем четным или нечетным слоям. Для оценки нормировочной константы применим метод промежуточных распределений между равномерным распределением и интересующим нас распределением $p(\boldsymbol{x}, \boldsymbol{t}^2|W)$:

$$p_k(\boldsymbol{x}, \boldsymbol{t}^2|W) \propto [p_{const}(\boldsymbol{x}, \boldsymbol{t}^2)]^{1-\beta_k} [p(\boldsymbol{x}, \boldsymbol{t}^2|W)]^{\beta_k},$$

где $p_{const}(\boldsymbol{x}, \boldsymbol{t}^2) = 1/2^{N_1+N_2}$, N_1, N_2 – число узлов на первом и втором слое DBM соответственно. Легко показать, что схема Гиббса для генерации выборки из распределения p_k реализуется с

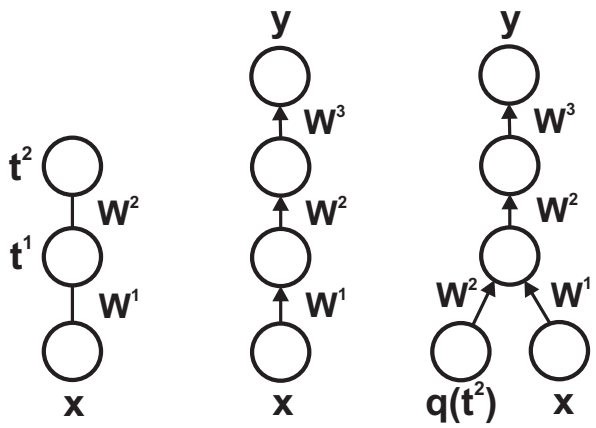


Рис. 3: Слева: модель DBM с двумя слоями скрытых переменных, Центр: преобразование DBM к многослойному персептрону, Справа: преобразование DBM к многослойному персептрону, в котором дополнительно на вход подается вектор признаков, полученный с помощью DBM.

помощью следующих формул:

$$p_k(x_i = 1|\boldsymbol{t}^2, \boldsymbol{x}_{\setminus i}) = \frac{1}{1 + \prod_j \left(\frac{1+\exp(\sum_{i_1 \neq i} W^2_{i_1 j} x_{i_1} + \sum_k W^2_{jk} t^2_k)}{1+\exp(W^2_{ij} + \sum_{i_1 \neq i} W^2_{i_1 j} x_{i_1} + \sum_k W^2_{jk} t^2_k)}\right)^{\beta_k}},$$

$$p_k(t^2_k = 1|\boldsymbol{x}, \boldsymbol{t}^2_{\setminus k}) = \frac{1}{1 + \left[\prod_j \frac{1+\exp(\sum_j W^1_{ij} x_i + \sum_{k_1 \neq k} W^2_{jk_1} t^2_{k_1})}{1+\exp(\sum_j W^1_{ij} x_i + \sum_{k_1 \neq k} W^2_{jk_1} t^2_{k_1} + W^2_{jk})} \prod_m \frac{1+\exp(\sum_{k_1 \neq k} W^3_{k_1 m} t^2_{k_1})}{1+\exp(\sum_{k_1 \neq k} W^3_{k_1 m} t^2_{k_1} + W_{km})}\right]^{\beta_k}}.$$

После получения оценки для нормировочной константы $Z(W)$ значение плотности для тестового объекта $\boldsymbol{x}_*$ можно оценить с помощью вариационного подхода:

$$\log p(\boldsymbol{x}_*|W) \geq -\sum_{\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3} q(\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|\boldsymbol{\mu}) E(\boldsymbol{x}_*, \boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|W) + \mathcal{H}(q) - \log Z(W).$$

Здесь q – факторизованное приближение для апостериорного распределения $p(\boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3|\boldsymbol{x}, W)$ из вариационного подхода.

Deep Boltzmann Machine как дискриминативная модель

Модель DBM можно использовать различными способами для решения задач классификации и регрессии.

Первый способ предполагает использовать скрытые признаки, полученные с помощью DBM, в качестве новых признаков объектов, для которых запускается отдельный независимый метод классификации или регрессии. Пусть у нас имеется обученная модель DBM с двумя слоями скрытых переменных (см. рис. 3, слева). Тогда новые признаки для объекта $\boldsymbol{x}$ представляют собой вероятности $q_k(t^2_k = 1)$ из вариационной оценки апостериорного распределения $p(\boldsymbol{t}^2|\boldsymbol{x}, W)$. Модификацией этой идеи является преобразование модели DBM в модель многослойного персептрона вместе с добавлением на последнем слое линейного (для задачи регрессии) или логистического/мультиномиального (для задачи классификации) узла (см. рис. 3, центр). Построенный таким образом многослойный персептрон обучается с помощью метода обратного

распространения ошибки, в котором в качестве начального приближения для весов сети используются обученные значения W для DBM. Можно также добавить признаки $q_k(t_k^2 = 1)$, обученные с помощью DBM, в набор входов многослойного персептрона (см. рис. 3, справа).

Описанный способ применения DBM как дискриминативной модели позволяет решать задачи классификации и регрессии с частично размеченной обучающей выборкой (semi-supervised learning). Действительно, на этапе обучения DBM информация о метках объектов не используется. Фактически с помощью DBM восстанавливается плотность $p(\boldsymbol{x})$, которая соответствует осмысленным входным объектам. Затем производится обучение многослойного персептрона по полностью размеченной выборке, которое приводит лишь к небольшой коррекции весов сети.

Второй способ решения задач классификации с помощью DBM предполагает использование байесовского классификатора, где для моделирования плотности каждого из классов используется своя модель DBM. В отличие от предыдущего способа, здесь необходимо обучать отдельную DBM для каждого класса. При этом нельзя решать задачу классификации с частично размеченными выборками.

При третьем способе метка класса добавляется в качестве входа в модель DBM. На этапе распознавания тестового объекта к признакам тестового объекта добавляется по очереди каждая из возможных меток классов и вычисляется энергия $E(\boldsymbol{x}|W)$ такой конфигурации. Метка класса, соответствующая минимальной энергии, выдается в качестве результата распознавания. В отличие от второго способа, здесь производится обучение одной общей модели DBM, но также нельзя решать задачи классификации с частично размеченными выборками.

Виды переменных в Deep Boltzmann Machine

К настоящему моменту рассматривались модели DBM только с бинарными наблюдаемыми и скрытыми переменными. Однако, на практике существует необходимость работы с переменными других типов, в частности, с непрерывными и K-значными переменными.

K-значные переменные являются удобными для моделирования метки класса, например, при третьем способе использования DBM как дискриминативной модели. K-значные переменные добавляются в DBM как K бинарных переменных $x_1, \ldots, x_K$ без внутренних связей с дополнительным ограничением, что одна и только одна из этих переменных может принимать значение единицы. Для применения схемы Гиббса необходимо уметь вычислять условное распределение на каждую из переменных x_k при условии всех ненаблюдаемых переменных $\boldsymbol{t}$. Данное условное распределение определяется мультиномиальной функцией:

$$p(x_k|\boldsymbol{t}, W) = \frac{\exp(\sum_j W_{kj} t_j)}{\sum_{m=1}^{K} \exp(\sum_j W_{mj} t_j)}.$$

Рассмотрим случай непрерывных наблюдаемых переменных $\boldsymbol{x}$ и бинарных скрытых переменных $\boldsymbol{t}$ в модели RBM. В этом случае энергия системы записывается как

$$E(\boldsymbol{x}, \boldsymbol{t}|W, \boldsymbol{a}, \boldsymbol{b}) = \sum_i \frac{(x_i - a_i)^2}{2} - \sum_j b_j t_j - \sum_{i,j} W_{ij} x_i t_j.$$

Таким образом, для непрерывных переменных добавляются квадратичные унарные потенциалы вместо линейных для случая бинарных переменных. Условные одномерные распределения в

такой модели вычисляются как

$$p(x_i|\boldsymbol{t}, W, \boldsymbol{a}, \boldsymbol{b}) = \mathcal{N}(x_i|a_i + \sum_j W_{ij} t_j, 1),$$

$$p(t_j = 1|\boldsymbol{x}, W, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{1 + \exp(-b_j - \sum_i W_{ij} x_i)}.$$

Наконец, в случае непрерывных наблюдаемых и скрытых переменных энергия RBM записывается как

$$E(\boldsymbol{x}, \boldsymbol{t}|W, \boldsymbol{a}, \boldsymbol{b}) = \sum_i \frac{(x_i - a_i)^2}{2} + \sum_j \frac{(t_j - b_j)^2}{2} - \sum_{i,j} W_{ij} x_i t_j.$$

Одномерные условные распределения в такой модели вычисляются как

$$p(x_i|\boldsymbol{t}, W, \boldsymbol{a}, \boldsymbol{b}) = \mathcal{N}(x_i|a_i + \sum_j W_{ij} t_j, 1),$$

$$p(t_j|\boldsymbol{x}, W, \boldsymbol{a}, \boldsymbol{b}) = \mathcal{N}(t_j|b_j + \sum_i W_{ij} x_i, 1).$$

Примеры применения Deep Boltzmann Machine

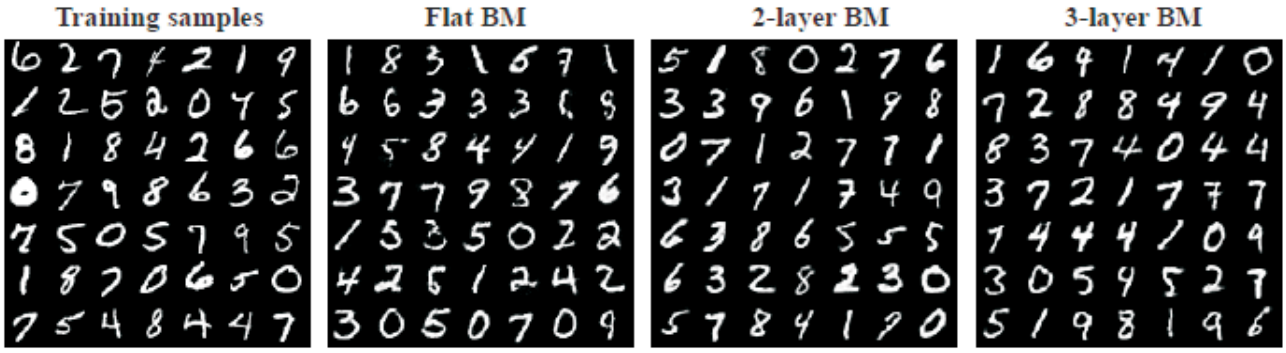


Рис. 4: Примеры сгенерированных объектов в задаче MNIST для различных моделей. Слева: примеры изображений из обучающей выборки, Центр слева: примеры сгенерированных изображений из модели BM, Центр справа: примеры сгенерированных изображений из модели DBM с двумя слоями скрытых переменных, Справа: примеры сгенерированных изображений из модели DBM с тремя слоями скрытых переменных.

Рассмотрим известную задачу распознавания рукописных цифр MNIST[1]. В этой задаче объектами являются изображения размера 28×28, каждое из которых содержит написание одной из цифр 0–9 (см. рис. 4,слева). Всего имеется 60000 изображений в обучающей выборке и 10000 изображений в тестовой выборке. При применении BM в данной задаче каждое изображение интерпретировалось как бинарный вектор длины 784, где значение каждого элемента вектора генерировалось с вероятностью, пропорциональной интенсивности пиксела.

Рассмотрим применение трех моделей Boltzmann Machine для задачи MNIST: 1) двухслойная полносвязная BM с 784 наблюдаемыми переменными и 500 скрытыми переменными, 2)

[1]http://yann.lecun.com/exdb/mnist/

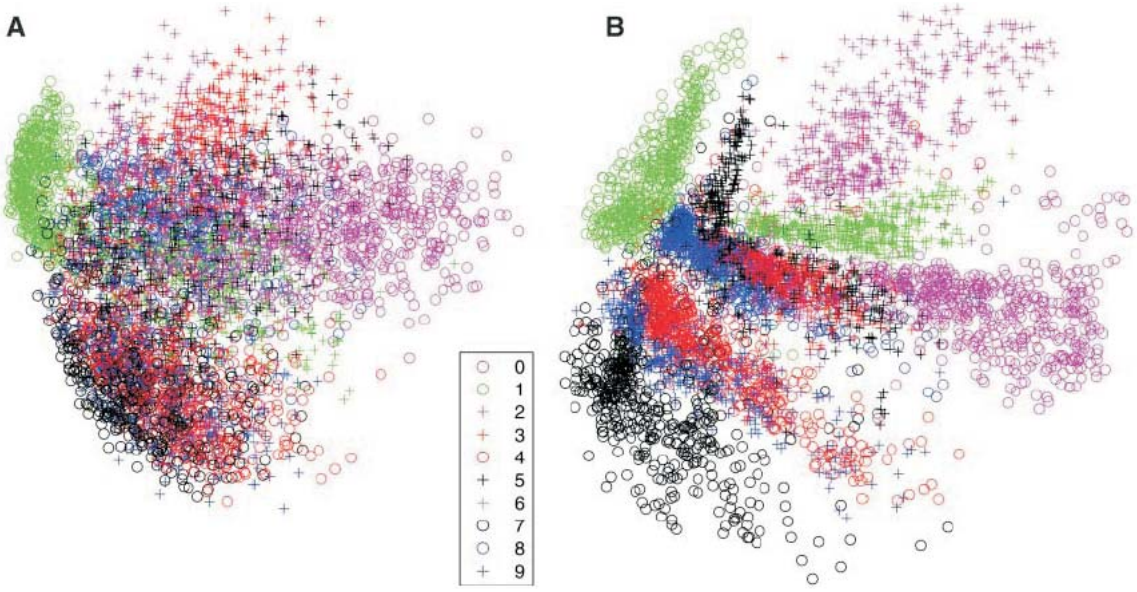


Рис. 5: Визуализация выборки MNIST с помощью метода главных компонент (A) и с помощью модели DBM с конфигурацией 784-1000-500-250-2.

DBM с двумя слоями скрытых переменных 784-500-1000 и 3) DBM с тремя слоями скрытых переменных 784-500-500-1000. На рис. 4 показаны примеры сгенерированных изображений из обученных моделей. Как видно из этого рисунка, генерируемые изображения очень похожи на изображения рукописных цифр. При этом модели DBM обучаются существенно быстрее, чем полносвязная модель BM. Дополнительное дискриминативное обучение моделей DBM в этой задаче дает процент ошибок для тестовой выборки 0.95% для DBM с двумя слоями и 1.01% для DBM с тремя слоями скрытых переменных. По утверждению авторов статьи [1] это наилучший процент, которого удалось добиться человечеству для задачи MNIST на сегодняшний день.

Рассмотрим также применение модели DBM с конфигурацией 784-1000-500-250-2 в качестве метода уменьшения размерности в задаче MNIST. На рис. 5 показана проекция обучающей выборки на первые две главные компоненты (слева) и найденное в помощью DBM двухпризнаковое описание для обучающей выборки (справа). Как видно из этого рисунка, модель DBM обеспечивает значительно лучшую разделимость по классам по сравнению с методом главных компонент.

В заключение данного раздела рассмотрим применение DBM для задачи NORB. Исходные данные в этой задаче представляют собой картинки размера 96×96, содержащие изображения трехмерных игрушек, снятых в различных условиях освещенности и с различных углов зрения (см. рис. 6, слева). При этом данные объекты разбиты на 5 общих классов: машины, грузовики, самолеты, животные и люди. Задача состоит в предсказании общих классов для тестовых изображений.

Для решения данной задачи применялась модель DBM с конфигурацией 9216-4000-4000-4000, где наблюдаемые переменные являлись непрерывными величинами, значения которых соответствовали уровню интенсивности пикселов. На рис. 6,справа показаны примеры сгенерированных изображений из обученной модели DBM. Дополнительное дискриминативное обучение данной модели DBM дает уровень ошибок 10.8% на тестовой выборке. По утверждению авторов статьи [1] этот уровень ошибок является наилучшим из опубликованных для дан-

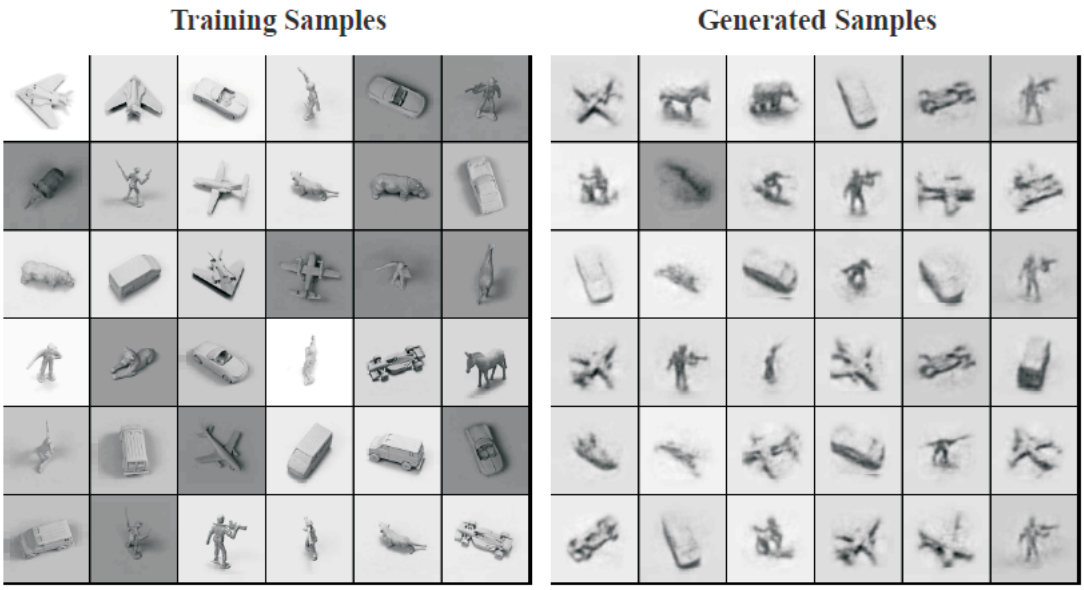


Рис. 6: Слева: примеры изображений в задаче NORB, справа: примеры сгенерированных изображений из обученной модели DBM.

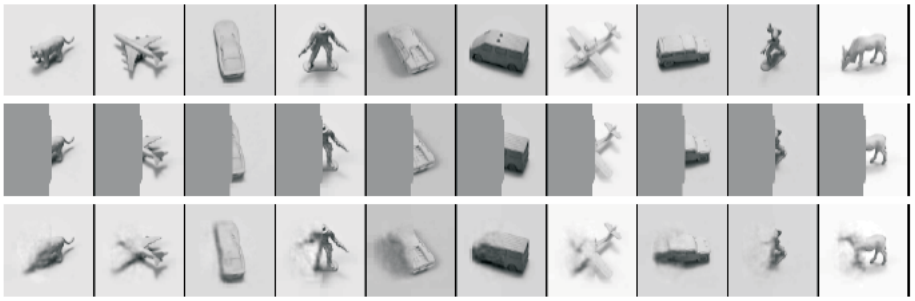


Рис. 7: Иллюстрация к задаче восстановления изображений. Сверху: примеры изображений из тестовой выборки, середина: частично-стертые изображения, подаваемые на вход модели DBM, снизу: восстановленные изображения с помощью обученной модели DBM.

ной задачи на настоящий момент. Кроме того, модель DBM может обучаться на частично-классифицированной обучающей совокупности. При добавлении в обучение большого количества изображений, полученных из исходных простыми преобразованиями пикселов и различными размытиями, позволяет снизить уровень ошибок до 7.2%.

Наконец, рассмотрим применение модели DBM для решения задачи восстановления изображений по данным NORB. На рис. 7 показаны примеры тестовых изображений (сверху), их частично-стертые варианты, подаваемые на вход обученной модели DBM (середина) и результат восстановления (снизу). В частности, здесь модель сумела восстановить стертые части изображений даже для объектов ковбоя, которые ни разу не встречались на этапе обучения.

Список литературы

[1] R. Salakhutdinov, G. Hinton. An Efficient Learning Procedure for Deep Boltzmann Machines // MIT Technical Report MIT-CSAIL-TR-2010-037, 2010.

[2] G. Hinton, T. Sejnowski. Optimal perceptual inference // CVPR, 1983.

[3] R. Neal. Connectionist learning of belief networks // Artificial Intelligence, V. 56, No. 1, 1992, pp. 71–113.

[4] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory // Parallel Distributed Computing, V. 1, 1986.