Теоретический минимум. БММО

Авторы: Максим Минец, Мария Павлеева

1. Базовые понятия. Мат.ожидание, мода, медиана, дисперсия и матрица ковариаций случайной величины. Функция правдоподобия, метод максимального правдоподобия, его недостатки.

Математическим ожиданием (или ожидаемым значением) случайной величины X называется число $E(X) = \sum\limits_{i=1}^{N} X(\omega_i)p_i$.

Для непрерывных случ. величин: $E(X) = \int\limits_{-\infty}^{+\infty} x \cdot f(x)dx$.

Дисперсией случайной величины X называется ожидаемое значение случаной величины $(X - E(X))^2$, которое обозначается $D(X)$. Другими словами, $D(X) = \sum\limits_{i=1}^{N}(X(\omega_i) - E(X))^2 p_i$.

Для непрерывных случ. величин: $D(X) = \int\limits_{-\infty}^{+\infty} (x - E(x))^2 \cdot f(x)dx$.

Квантиль порядка 0.5 называется **медианой** случайной величины X.
Точка x_0 называется **модой** случайной величины X, если в точке x_0 функция $f(x)$ имеет локальный максимум (*другими словами* точной максимума плотности).

Ковариацией случайных величин X и Y называется число $Cov(X, Y) = E((X - E(X))(Y - E(Y)))$. Ковариация двух независимых случайных величин равна 0.
Ковариационная матрица случайного вектора – квадратная симметрическая неотрицательно определенная матрица, на диагонали которой располагаются дисперсии компонент вектора, а внедиагональные элементы — ковариации между компонентами.

Функция правдоподобия $L(\theta|x)$ отражает, насколько вероятно наблюдать данные x, если параметры распределения равны θ. Для набора данных $x = \{x_1, \ldots, x_n\}$ и параметров θ функция правдоподобия (в случае независимых наблюдений) может быть записана как:

$$L(\theta|x) = P(x|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

Метод максимального правдоподобия – метод оценки параметров статистической модели. Идея метода заключается в том, чтобы найти такие параметры $\hat{\theta}$, которые максимизируют функцию правдоподобия:

$$\theta_{ML} = \arg\max_{\theta} p(X|\theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta) = \arg\max_{\theta} \sum_{i=1}^{n} \log p(x_i|\theta)$$

Во многих частных случаях сумма логарифмов правдоподобий будет выпуклой вверх функцией, то есть у неё один максимум, который достаточно легко найти даже в про- странствах высокой размерности. Заметим, что θ_{ML} – случайная величина, поскольку она является функцией от выборки.

Максимум апостериорного распределения: $\theta_{ML} = \arg\max_\theta p(\theta|X) = \arg\max_\theta p(X|\theta)p(\theta) = \arg\max_\theta \prod_{i=1}^n p(x_i|\theta)p(\theta) = \arg\max_\theta \sum_{i=1}^n \ln p(x_i|\theta) + \ln p(\theta)$

Попробуем сформулировать нашу задачу **на вероятностном языке**. Для этого нам на- до задать вероятностную модель, то есть $p(T|X,w)$. Пусть $p(T|X,w) = N(T|Xw, \beta^{-1}I)$, тогда используя метод максимального правдоподобия мы получим:

$$p(T|X,w) \to \max_w, \qquad \log p(T|X,w) \to \max_w$$

$$Const - \frac{\beta}{2}||T - Xw||^2 \to \max_w$$

Полученная задача эквивалентна задаче минимизации $||T - Xw||^2 \to \max_w$.

Если добавим регуляризацию, то получим: $\ldots = -\frac{\beta}{2}||T - Xw||^2 - \frac{1}{2}\alpha||w||^2 \to \max_w$

Оценка максимума правдоподобия (ОМП) обладает очень хорошими свойствами:
– Состоятельность: ОМП сходится к истинному значению параметров по вероятности;
– Асимптотическая несмещенность;
– Асимптотическая нормальность: θ_{ML} распределена нормально при $n \to +\infty$;
– Асимптотическая эффективность: ОМП обладает наим. дисперсией среди всех состоятельных асимптотически нормальных оценок.

Недостатки:
– оценка является точечной;
– переобучение (в сложных моделях при большом количестве параметров модель может «подогнаться» под данные, что приводит к плохой обобщающей способности).

2. Условная вероятность. Правило суммы и произведения для вероятностей. Формула Байеса. Условная независимость случайных величин.

Определение 1. Пусть x и y – две случайные величины. Тогда условным распределением $p(x|y)$ (conditional distribution) x относительно y называется отношение совместного распределения $p(x|y)$ (joint distribution) и маргинального распределения $p(x)$ (marginal distribution, оно же безусловное):

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

Совместное распределение можно выразить через условное и маргинальное: $p(x,y) = p(x|y)p(y)$

Теорема 1 (Правило произведения). Пусть $x_1, \ldots, x_n$ — случайные величины. Тогда их совместное распределение можно представить в виде произведения n одномерных условных распределений с постепенно уменьшающейся посылкой:

$$p(x_1, \ldots, x_n) = p(x_n|x_1, \ldots, x_{n-1}) \ldots p(x_2|x_1)p(x_1) = p(x_1)\prod_{k=2}^n p(x_k|x_1, \ldots, x_{k-1}).$$

Теорема 2 (Правило суммирования). Пусть $x_1, \ldots, x_n$ — случайные величины. Если известно их совместное распределение $p(x_1, \ldots, x_n)$, то совместное распределение подмножества случайных величин $x_1, \ldots, x_k$ будет равно

$$p(x_1, \ldots, x_k) = \int p(x_1, \ldots, x_n)\mathrm{d}_{k+1} \ldots \mathrm{d}x_n.$$

Теорема Байеса. Пусть x и y – случайные величины. Тогда

$$p(x|y) = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$$

Вспомним какую задачу мы решаем. У нас есть выборка X и параметры θ для заданной вероятностной модели $p(X|\theta)$ (правдоподобие). Для использования теоремы Байеса нам необходимо задать априорное распределение на параметр — $p(\theta)$.
Наша цель найти апостериорное распределение $p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta}$. Важно, что знаменатель не зависит от θ, то есть является нормировочной константой. Мы будем писать для краткости $p(\theta|X) = \frac{1}{z}p(X|\theta)p(\theta)$, где z — нормировочная константа.
Сделать байесовский вывод (найти апостериорное распределение) возможно только при нахождении нормировочной константы, что в общем виде является трудной задачей. Одним из основных инструментов является использование сопряженных распределений.

A и B **условно независимы** относительно $C \iff P(AB \mid C) = P(A \mid C)P(B \mid C)$.
или равносильно:
A и B **условно независимы** относительно $C \iff P(A|BC) = P(A|B) + P(A|C) = 1$.

3. Сопряженные распределения: определение и примеры. Связь сопряженных распределений и полного байесовского вывода.

Коротко говоря, если и априорное распределение, и апостериорное распределение принадлежат к одному семейству, то они сопряжены.

Пусть функция правдоподобия и априорное распределение принадлежат некоторым параметрическим семействам распределений: $p(X|\theta) \sim \mathcal{A}(\theta)$ и $p(\theta|\beta) \sim \mathcal{B}(\beta)$. Семейства являются сопряжёнными (conjugate) тогда и только тогда, когда $p(\theta|X) \sim \mathcal{B}(\beta')$.
Благодаря сопряженности, мы можем найти новые параметры β', через которые полу- чится восстановить всё распределение (например, найти нормировочную константу).

Пример 1 (Биномиальное распределение и бета-распределение)
Если данные x следуют биномиальному распределению $x \sim Bin(n, \theta)$, где θ — вероятность успеха, то бета- распределение $Beta(\alpha, \beta)$ является сопряженным априорным распределением для параметра θ. Апостериорное распределение $p(\theta|x)$ также будет бета- распределением.

$$p(x|\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x}$$

$$p(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}$$

$$p(\theta|x) = p(x|\theta)p(\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} = \frac{1}{z}\cdot\frac{\theta^{\alpha+x-1}(1-\theta)^{\beta+n-x-1}}{B(\alpha,\beta)}$$

$$p(\theta|x) = Beta(\alpha+x, \beta+n-x)$$

Пример 2 (Пуассоновское распределение и гамма-распределение)
Пуассоновское распределение описывает число событий в единицу времени и параметризуется интенсивностью λ. Если данные $x \sim \Pi(\lambda)$, то гамма-распределение $Gamma(\alpha, \beta)$ является сопряженным для параметра α.

$$p(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$p(\lambda) = \frac{\beta^\alpha\lambda^{\alpha-1}e^{-\beta\lambda}}{\Gamma(\alpha)} = \frac{1}{z}\cdot\lambda^{\alpha-1}e^{-\beta\lambda}$$

$$p(\lambda|x) = p(x|\lambda)p(\lambda) = \frac{1}{z}\cdot\frac{\lambda^x e^{-\lambda}}{x!}\cdot\lambda^{\alpha-1}e^{-\beta\lambda} = \frac{1}{z}\cdot\frac{\lambda^{\alpha+x-1}e^{-\lambda(\beta+1)}}{x!}$$

$p(\lambda|x) = Gamma(\alpha + x, \beta + 1)$

Связь сопряженных распределений и полного байесовского вывода
Полный байесовский вывод заключается в нахождении апостериорного распределения на основе априорного распределения и наблюдаемых данных. Сопряжённые распределения существенно упрощают этот процесс, так как они обеспечивают аналитическую форму апостериорного распределения без необходимости вычисления сложных интегралов.

4. Метод релевантных векторов для задачи регрессии: постановка задачи, зачем используется, с помощью какого метода обучается.

Постановка задачи
Нам бы хотелось, чтобы веса «важных» признаков подстраивались под данные, а веса «неважных» признаков этого не делали, потому что последние могут подстроиться только под шум в данных, что непременно приведет к переобучению. Однако, варьируя α мы не можем этого добиться, потому что она одинаково влияет на все веса.

Зачем используется
– Обеспечить разреженность и улучшить интерпретируемость модели;
– Автоматически исключить нерелевантные признаки и уменьшить переобучение;
– Сделать вероятностные предсказания с оценкой их неопределенности.

С помощью какого метода обучается
Обучается с помощью метода наибольшей обоснованности, что позволяет выбрать наиболее обоснованную модель, учитывая что распределения сопряжены и подсчёт обоснованности должен быть несложным. Заметим, что множество, из которого мы выбираем модели, не конечно, то есть необходимо посчитать обоснованность так, чтобы можно было бы вести оптимизацию.

5. Дивергенция Кульбака-Лейблера, её основные свойства и использование для поиска аппроксимирующих распределений.

Дивергенцией Кульбака-Лейблера называется функционал: $\mathrm{KL}(p\|q) = \int p(x) \log \dfrac{p(x)}{q(x)} dx$.
Данный функционал имеет смысл «расстояния» между распределениями и обладает следующими свойствами:

- $\mathrm{KL}(p\|q) \geq 0, \ \forall p, \ q;$
- $\mathrm{KL}(p\|q) = 0 \iff p = q.$

Использование для поиска аппроксимирующих распределений
Предположим, у нас есть сложное апостериорное распределение $P(\theta|x)$, которое невозможно вычислить напрямую. Мы можем аппроксимировать его с помощью более простого распределения $Q(\theta)$, минимизируя KL-дивергенцию:

$$KL(Q(\theta)||P(\theta|x)) = \int Q(\theta) \log \frac{Q(\theta)}{P(\theta|x)} d\theta$$

Минимизация этой дивергенции приводит к выбору такого $Q(\theta)$, которое наиболее близко соответствует апостериорному распределению. Для этого мы можем использовать методы оптимизации, такие как градиентный спуск или метод ожидания-максимизации (EM).

6. Общая схема EM-алгоритма: какая задача решается и как именно? Формулы для оптимизируемого функционала, для E и M шагов.

EM-алгоритм — итерационный метод максимизации правдоподобия выборки, которая зависит от скрытых переменных. Пусть есть следующая задача: $\log p(X|\theta) \to \max_\theta$.

Идея EM-алгоритма заключается в том, чтобы вместо оптимизации логарифма неполного правдоподобия оптимизировать полученную нижнюю оценку, но теперь уже как по θ так и по распределению q.

E-step.

$$q(z_i) = p(z_i|l_i, \alpha, \beta_i) = \frac{p(z_i, l_i|\alpha, \beta_i)}{p(l_i|\alpha, \beta_i)} = \frac{p(z_i)\prod_j \left(\sigma(\alpha_j\beta_i)^{[l_{ij}=z_i]} + \sigma(-\alpha_j\beta_i)^{[l_{ij}\neq z_i]}\right)}{\sum_{t\in[0,1]} p(t)\prod_j \left(\sigma(\alpha_j\beta_i)^{[l_{ij}=t]} + \sigma(-\alpha_j\beta_i)^{[l_{ij}\neq t]}\right)}$$

$$q_i^*(z_i) = \frac{exp\left(\sum\limits_j \left([l_{ij}=z_i]\log\left(\sigma(\alpha_j\beta_i)\right) + [l_{ij}\neq z_i]\log\left(\sigma(-\alpha_j\beta_i)\right)\right)\right)}{\sum\limits_{t\in\{0,1\}} exp\left(\sum\limits_j \left([l_{ij}=t]\log\left(\sigma(\alpha_j\beta_i)\right) + [l_{ij}\neq t]\log\left(\sigma(-\alpha_j\beta_i)\right)\right)\right)}$$

M-step.

$$\mathbb{E}_{z\sim q(z)} \log p(z, l|\alpha, \beta) \to \min_{\alpha,\beta}$$

$$\mathbb{E}_{z\sim q(z)} \log p(z, l|\alpha, \beta) = \sum_i \sum_j \sum_{t\in\{0,1\}} q_i^*(t)\left([l_{ij}=t]\log\sigma(\alpha_j, \beta_i) + [l_{ij}\neq t]\log\sigma(-\alpha_j, \beta_i)\right) + const$$

Формула для оптимизируемого функционала E шага:
Максимизируем по q при фиксированном θ (минимизируем KL-дивергенцию (интеграл)).
Из полученного ранее следует, что максимум $\mathcal{L}(q, \theta)$ по q достигается в том случае, когда достигается минимум $\mathrm{KL}(q\|p)$, то есть когда $q = p$:

$$\mathcal{L}(\theta, q_0) \to \max_q \Rightarrow q(z) = p(z|X, \theta)$$

$$q^*(Z) = argmax_q \mathcal{L}(q, \theta^{\text{old}}) = argmin_q \int q(Z)\log\frac{q(Z)}{p(Z|X, \theta^{\text{old}})}dZ = p(Z|X, \theta^{\text{old}})$$

или

$$KL(q(z)||p(z|x, Q)) \to \min_q$$

Формула для оптимизируемого функционала M шага:
Максимизируем по θ при фиксированном q

$$\mathcal{L}(q_0, \theta) \to \max_\theta \Rightarrow \int q(Z)\log p(X, Z|\theta)dZ \to \max_\theta$$

$$\theta^{\text{new}} = argmax_\theta \int q(Z)\log\frac{p(X, Z|\theta)}{q(Z)}dZ = argmax_\theta \mathbb{E}_{Z\sim q(Z)}\log p(X, Z|\theta)$$

7. Методы Монте-Карло, какая связь с методами семплирования

Метод Монте-Карло применяется для оценки математического ожидания функции $f(x)$ по распределению $p(x)$, особенно в задачах машинного обучения, где такие интегралы сложно посчитать аналитически из-за высокой размерности пространства. То есть, когда мы переходим от интеграла к какой-то оценке – мы используем метод Монте-Карло.

Какая связь с методами семплирования
Семплирование — это ключевая составляющая методов Монте-Карло. Важным этапом многих задач Монте-Карло является генерация случайных выборок (семплов) из распределения, представляющего интерес. Методы семплирования используются для того, чтобы извлекать случайные значения из вероятностных распределений, которые могут быть сложными или невозможными для прямого семплирования.

Чтобы посчитать какой-то интеграл, нужно посчитать его в какой-то точке из распределения $q(z)$. Как получить z из распределения – это буквально то, как засэплировать из распределения (например, из нормального или равномерного).

Методы генерации выборок из одномерных распределений:
1) Метод Rejection Sampling (если не удаётся явно сгенерировать выборку из плотности $p(x)$, выбирается другая плотность $q(x)$, которую можно промасштабировать, чтобы она покрывала $p(x)$ сверху);
2) Метод Importance sampling (уменьшить отклонения сэмплов);
3) Метод Метрополиса-Хастингса (см. *вопрос 8*).

8. Алгоритм Метрополиса-Хастингса: какая задача решается и как именно?

Основной идеей метода Метрополиса-Хастингса является отказ от сэмплирования из равномерного распределения. Вместо этого предлагается сэмплировать последовательно из марковской цепи.

Метод Метрополиса-Хастингса
Пусть имеется распределение $p(x) = \frac{\hat{p}(x)}{B}$, известное с точностью до нормировочной константы, и функция $q(x' \mid x) > 0 \, \forall x', x \in \mathcal{X}$. Определим вероятность $A(x, x')$ принятия перехода из x' в x как

$$A(x, x') = \min\left(1, \frac{\hat{p}(x')q(x \mid x')}{\hat{p}(x)q(x' \mid x)}\right) = \min\left(1, \frac{p(x')q(x \mid x')}{p(x)q(x' \mid x)}\right)$$

Тогда для $\forall x_0$ и марковской цепи с правилом перехода:

$$x_{n+1} = \begin{cases} x' \sim q(x' \mid x_n), & \text{с вероятностью } A(x_n, x'), \\ x_n, & \text{с вероятностью } 1 - A(x_n, x'), \end{cases}$$

гарантируется, что $\exists N : \forall n \geq N \Rightarrow x_n \sim \hat{p}(x_n)$.

Схема Метрополиса-Хастингса (кратко):
1) Инициализация: задаём начальное состояние цепи $x_0 \sim p(x_0)$.
2) Для каждой итерации $n = \overline{1, k}$
   a) Шаг предложения: Генерируем новое предложение x' из вспомогательного распределения $q(x'|x_n)$, которое зависит от текущего состояния x_t, то есть $q(x_{n+1}|x_n) = N(x_{k+1}|x_k, 1)$
   b) Шаг принятия: Рассчитываем вероятность принятия нового состояния x' с помощью формулы для расчета $A(x, x')$
   c) Сгенерируем случайное число u из равномерного распределения $U(0, 1)$. В зависимости от этого принимаем/отклоняем новое состояние для x_{n+1}.
3) Повторяем шаги до тех пор, пока не будет получено нужное количество выборок.

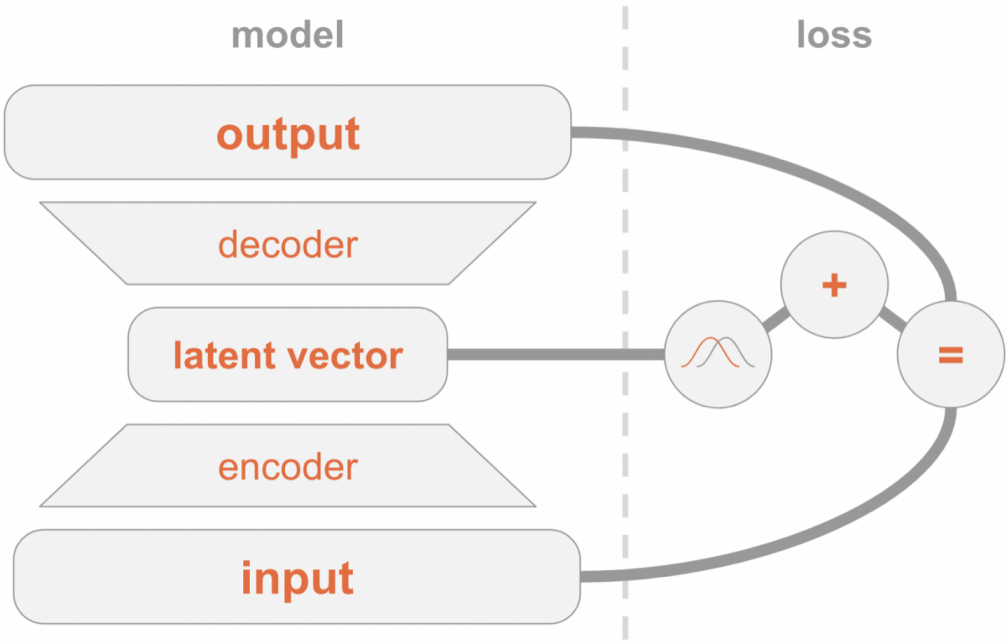
Схема считается эффективной, если по ходу процесса точки-кандидаты x' принимаются с высокой частотой. Если же большая часть сгенерированных точек будет отвергать- ся, то в схеме будет много повторов, что можно считать недостатком метода. Еще одним недостатком является корреляция между последовательными элементами В принципе, это недостаток

MCMC методов – высокая скоррелированность соседних случайных величин, а хотели бы выборку.

9. Вариационный автокодировщик: модель и как ее тренировать

Вариационные автоэнкодеры (Variational Autoencoders) — это автоэнкодеры, которые учатся отображать объекты в заданное скрытое пространство и, соответственно, сэмплить из него. Поэтому вариационные автоэнкодеры относят также к семейству генеративных моделей (оценивает плотность вероятности (PDF) обучающих данных. Если такая модель обучена на натуральных изображениях, то присвоит изображению льва высокое значение вероятности, а изображению случайной ерунды — низкое значение).

Как работает VAE:
– **Энкодер (encoder):** Энкодер принимает входные данные и преобразует их в набор параметров распределения, а не в фиксированное представление. Обычно это параметры многомерного нормального распределения — среднее (μ) и стандартное отклонение (σ) для каждого измерения скрытого пространства. Эти параметры затем используются для генерации вероятностного представления (латентного вектора).
– **Выборка в скрытом пространстве (latent space):** Чтобы моделировать вероятностное распределение, используется трюк репараметризации (reparameterization trick). Вместо того, чтобы напрямую выбирать значения для скрытого вектора, выборка делается из нормального распределения, а затем результат масштабируется и смещается с помощью μ и σ от энкодера. Это позволяет осуществлять backpropagation, несмотря на случайный характер выборки.
– **Декодер (decoder):** Декодер принимает случайно выбранное значение из скрытого пространства и преобразует его обратно в исходные данные. Он пытается реконструировать входные данные с учетом вероятностной природы представления, что позволяет эффективно моделировать сложные структуры данных. **Слева у нас определение модели:**



1. Входное изображение передаётся через сеть кодировщика.
2. Кодировщик выдаёт параметры распределения $Q(z|x)$.
3. Скрытый вектор z берётся из $Q(z|x)$. Если кодировщик хорошо обучен, то в большинстве случае z содержат описание x.

4. Декодер декодирует z в изображение.

С правой стороны у нас функция потери:
1. Ошибка восстановления: выходные данные должны быть аналогичны входным.
2. $Q(z|x)$ должно быть аналогично предыдущему, то есть многомерному стандартному нормальному распределению.

Вариационный автокодировщик: как ее тренировать
Задача тренировки заключается в максимизации ELBO, что эквивалентно минимизации расстояния между настоящим апостериорным распределением и вариационной аппроксимацией.

Шаги тренировки VAE
1. Прямой проход: Кодировщик принимает на вход данные x и предсказывает параметры распределения $q(z|x)$ — среднее $\mu(x)$ и дисперсию $\sigma(x)$. Затем сэмплируется латентная переменная z с использованием трюка репараметризации. Декодер принимает z и предсказывает параметры распределения данных $P(x|z)$, по которым оценивается вероятность восстановления данных.
2. Вычисление ELBO: Вычисляется функция реконструкции $E_{q(z|x)}[logP(x|z)]$, которая сравнивает восстановленное x с реальными данными. Вычисляется KL-дивергенция между $q(z|x)$ и $P(z)$.
3. Обратный проход и обновление параметров: Вычисляются градиенты ELBO по параметрам кодировщика и декодера. Параметры обновляются с помощью стохастического градиентного спуска (или других методов оптимизации).

Трюк репараметризации
Предположим, что кодировщик аппроксимирует распределение $q(z|x)$ как гауссовское $N(z; \mu(x), \sigma^2(x)$ Тогда вместо того, чтобы напрямую сэмплировать z, мы можем представить его как: $z = \mu(x) + \sigma(x)\epsilon$ где $\epsilon N(0, I)$ — это случайная переменная, сэмплированная из стандартного нормального распределения. Теперь z выражается как дифференцируемая функция от $\mu(x)$, $\sigma(x)$ и ϵ, и можно дифференцировать ELBO с помощью стандартных методов автоматического дифференцирования.

10. Модель диффузии: как ее обучить

Алгоритм обучения
Важно отметить, что мы можем обучаться не на μ, а на ϵ или даже на x_0, которым зашумили чистые объекты.
1) Обучение модели происходит через итеративный процесс, который повторяется много раз до тех пор, пока модель не сойдется. В каждом повторе выполняются шаги 2-6:
2) Выбирается случайный объект x_0 из обучающего набора данных D;
3) Случайным образом выбирается шаг времени $t \sim \text{Uniform}(\{1, 2, \ldots, T\})$;
4) Генерируется случайный шум $\epsilon \sim N(0, 1)$;
5) Производится шаг прямого процесса диффузии $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon_t$
6) Выполняется шаг градиентного спуска для минимизации функции потерь

$$\nabla_\theta \frac{1}{2\sigma_t^2} ||\widetilde{\mu}(x_t, x_0, t) - \mu_\theta(x_t, t)||^2.$$

Здесь $\widetilde{\mu}(x_t, x_0, t)$ — это "истинное"распределение, или функция, которая описывает обратный процесс, а $\mu_\theta(x_t, t)$ — это предсказание модели для обратного процесса. Модель пытается минимизировать разницу между тем, как она предсказывает восстановление данных из x_t, и тем, как это должно происходить в идеале.

Алгоритм сэмплирования (получения x_0)
1) Начинаем с того, что сэмплируем случайное значение x_T из стандартного нормального распределения $N(0, I)$. Это значит, что на первом шаге T мы начинаем с полностью зашумленных данных, которые представляют собой чистый шум;
2) Далее запускается цикл по шагам времени t от максимального значения T до 1;
3) На каждом шаге t генерируется шум $\epsilon \sim N(0, I)$, но если $t = 1$, то вместо шума используется значение $\epsilon = 0$, т.к. на последнем шаге хотим получить восстановленные данные без дополнительного шума;
4) Обратный процесс диффузии: $x_{t-1} = \mu_\theta(x_t, t) + \sigma_t\epsilon$;
5) В результате работы алгоритма на выходе мы получаем восстановленные данные x_0, которые должны напоминать реальные данные, на которых была обучена модель.

Модель диффузии — это класс вероятностных моделей, используемых для генерации данных, которые основаны на процессе диффузии, представляемом в виде последовательности шагов, переходящих от простого распределения к сложному. Эти модели особенно популярны для генерации изображений и текста, и они становятся все более актуальными в области машинного обучения.

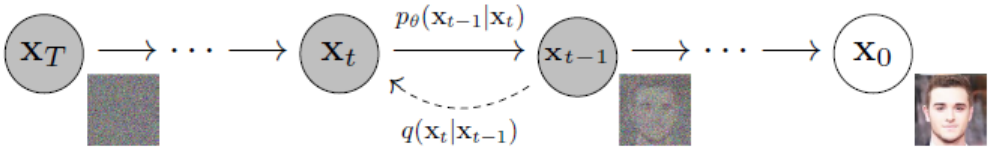


Рис. 1: Слева направо происходит процесс генерации изображения (Backward); Справа налево процесс разрушения изображения (Forward)

Модель диффузии работает на основе идеи, что процесс генерации сложных данных можно рассматривать как последовательность случайных изменений (шумов) в данных. Она описывает два ключевых процесса:

1) **Прямой процесс диффузии (forward diffusion process):** начинается с чистых данных и добавляет к ним шум в несколько шагов, превращая их в распределение, близкое к нормальному. Этот процесс обычно моделируется как последовательность гауссовских шумов
На каждом шаге добавляется шум, что можно записать следующим образом:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon_t,$$

где $\epsilon \in N(0, 1)$ – гауссовский шум, а α_t – коэфф-т, определяющий уровень шума на шаге t.
Эта модель добавляет шум на каждом шаге, таким образом, после шагов мы получаем x_t, которое близко к нормальному распределению.
Из прямого процесса также можем вычислить $q(x_t|x_{t-1})$ и $q(x_t|x_0)$.

2) **Обратный процесс диффузии (backward diffusion process):** это генеративный процесс, который начинает с нормально распределенного шума и пытается восстанавливать данные, постепенно удаляя шум, используя нейронную сеть для предсказания, каково было состояние данных на предыдущем шаге.
В диффузионных моделях для генерации определяется Backward процесс, именно с помощью него мы будем получать изображения. Пусть $x_T \sim N(x_T|0, I)$, а в Backward процессе мы знаем как перейти от x_T в x_{T-1}, как из x_{T-2} в x_{T-3} и так далее вплоть до перехода $x_1 \to x_0$. Наша цель, чтобы x_0 был из распределения данных, то есть очень похожим на изображение из датасета.
Обратный процесс описывается следующим образом:

$$q(x_{t-1}|x_t, x_0) = N(x_{t-1}|\mu(x_t, x_0, t), \sigma_t I),$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}|\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)),$$

где μ_θ и Σ_θ – параметры, предсказываемые нейронной сетью. Эти параметры определяют, как восстанавливать данные из состояния x_t в состояние x_{t-1}.

Глобальная цель

Найти такой процесс, чтобы x_0 было из распределения данных. Рассматривать все возможные процессы трудно, поэтому ограничимся только марковскими (переход зависит только от последнего состояния).

ДОПОЛНИТЕЛЬНО

Пусть $f(x)$ — некоторая функция действительного переменного. Тогда семейство функций двух переменных $g(x, \xi)$, обладающее свойствами
1) $\forall x, \forall xi \quad f(x) \leq g(x, \xi)$
2) $\forall x \exists \xi(x) : \quad f(x) = g(x, \xi(x))$,
называется **вариоционной нижней** оценкой функции f.

Нижняя граница на обоснованность (ELBO, evidence lower bound):
$\mathcal{L}(q, \theta) = \int q(Z) \log \frac{p(X,Z|\theta)}{q(Z)} dZ$

8) **Уравнение детального баланса.** Если для эргодичной однородной Марковской цепи и для $p_*(x)$ верно

$$\forall x, x' \in \mathcal{X} \quad p_*(x)q(x'|x) = p_*(x')q(x|x'),$$

то p_* – инвариантное распределение.

8) **Марковской цепью** назовём бесконечную последовательность случайных величин $\{x_n\}_{n=1}^{\infty}$ такую, что $\forall n$ верно разложение:

$$p(x_1, \ldots, x_n) = p_1(x_1)p_2(x_2|x_1)p_3(x_3|x_2) \ldots p_n(x_n|x_{n-1}),$$

где функции $p_i(\cdot|\cdot)$ называют **функциями перехода**.

8) Важными видами марковских процессов являются те, у которых функция перехода не зависит от номера перехода.

8) Марковская цепь называется **гомогенной (или однородной)**, если

$$\forall n \quad p_n(x_n|x_{n-1}) = q(x_n|x_{n-1}).$$

В этом случае Марковская цепь определяется стартовым распределением $p_1(x_1)$ и вероятностью перехода $q(x'|x)$.

8) Цепи, для которых существует единственное инвариантное распределение $p_*(x)$ будем называть **эргодическими**, если $\forall p_1(x) \quad \lim\limits_{n\to\infty} p_n(x) = p_*(x)$.

9) **Метод главных компонент (PCA)**
Метод главных компонент решает задачу уменьшения размерности признакового пространства. Оказывается, то же самое можно сделать на вероятностном языке. Вводим модель с латентными переменными:

$$p(x, z|\theta) = p(x|z, \theta)p(z) = N(x|\mu + Wz, \sigma^2 I)N(z|0, I),$$

где $z \in \mathbb{R}^d$ и играет роль сжатого представления исходного вектора $x \in \mathbb{R}^D$. В роли параметров модели θ выступают вектор $\mu \in \mathbb{R}^D$, линейный оператор $W \in \mathbb{R}^{D\times d}$ и скаляр σ. Эта вероятностная модель говорит, что у каждого x размерности D есть некоторое латентное представление z размерности d такое, что x является результатом действия линейного оператора W на z плюс какой-то сдвиг μ и плюс какой-то шум.
Поскольку мы наблюдаем только $X = (x_1, \ldots, x_n)$, в модели переменные z являются скрытыми.