VOLGOGRAD STATE TECHNICAL UNIVERSITY

Машинное обучение и нейросетевые модели

Лекция 5. Вариационный инференс

Лектор: Кравченя Павел Дмитриевич

Волгоград 2025

VOLGOGRAD STATE TECHNICAL UNIVERSITY

План лекции

1. Альтернативный способ решения задачи байесовского вывода.
2. Идея и сущность вариационного инференса.
3. Дивергенция Кульбака-Лейблера. Прямая и обратная дивергенция.
4. Оценка дивергенции Кульбака-Лейблера. Вариационная нижняя оценка.
5. Вариационные семейства. Подход Mean-Field. Покоординатный спуск.
6. Стохастический вариационный инференс.
7. Виды вариационного инференса: BBVI и ADVI, их особенности и алгоритмы.
8. Преобразования ограниченных переменных в ADVI.
9. Амортизированный вариационный инференс, особенности его реализации.
10. Выполнение вариационного инференса во фреймворке Pyro.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Задача байесовского вывода и способы её решения

- При выполнении *байесовского моделирования* возникают задачи определения апостериорного и предиктивного распределений, требующие вычисления *intractable*-интегралов.
- Один из способов решения таких задач состоит в применении *марковских цепей*, позволяющих эффективно получать *семплы* из *сложных многомерных распределений*.
- Данный метод позволяет получить *точные результаты* (при достаточно большом числе семплов), однако, требует *больших вычислительных ресурсов* и приводит к *значительному времени моделирования*.
- Данный метод *не является* единственным. Другим популярным методом решения задачи байесовского вывода является **вариационный инференс**.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Идея вариационного инференса

- Метод *вариационного инференса* базируется на следующей идее: если мы не можем семплировать из *сложного целевого многомерного распределения*, давайте **заменим это распределение** другим, *более простым*, вычисление семплов из которого будет возможным.
- При этом вновь введенное распределение (оно называется *вариационным*) должно быть *максимально похожим на целевое*, что позволит в дальнейшем применять эстиматор Монте-Карло для оценки интегралов с *наименьшей* погрешностью.
- *Если удастся* подобрать такое *вариационное распределение*, его можно будет использовать вместо целевого в дальнейших расчетах. Оно будет *отличаться* от целевого, но это различие несущественно на практике.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Дивергенция Кульбака-Лейблера

- Возникает вопрос: как оценить «*похожесть*» распределений?
- Для измерения схожести распределений $p(x)$ и $q(x)$ будем использовать **дивергенцию Кульбака-Лейблера**:

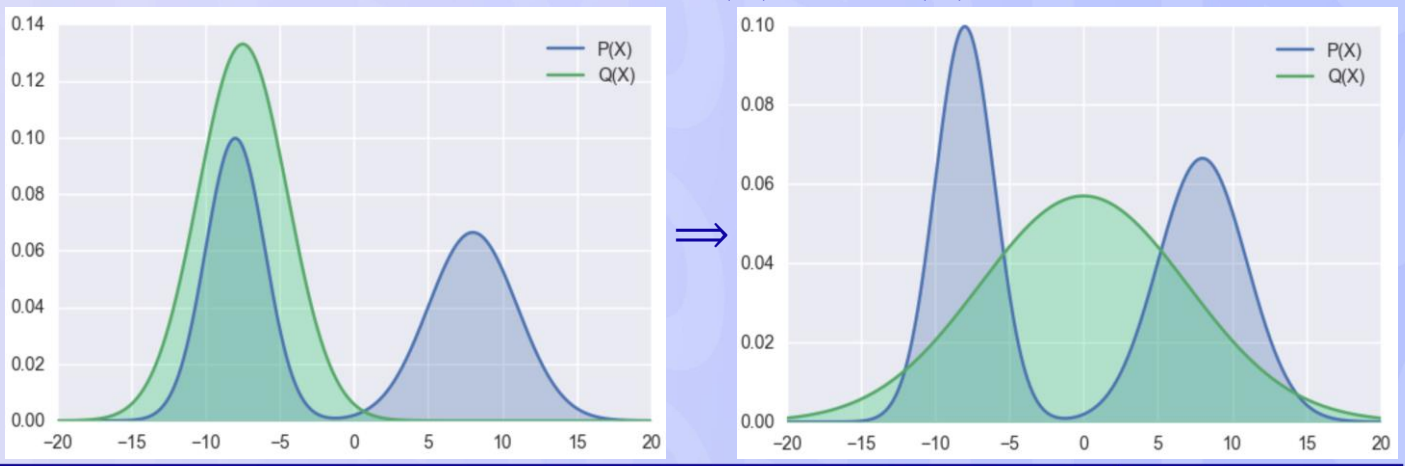
$$\mathrm{KL}\big(p(x)\,||\,q(x)\big) = \mathbb{E}_{p(x)}\left[\log\frac{p(x)}{q(x)}\right] = \int \log\frac{p(x)}{q(x)}\cdot p(x)dx\,.$$

- Данный термин был взят из *теории информации*, в которой он измеряет разницу количества информации, представленной двумя распределениями.
- Дивергенция Кульбака-Лейблера обладает *следующими свойствами*:
  - ✓ $\mathrm{KL}(p||q) \geq 0 \;\; \forall p, q.$
  - ✓ $\mathrm{KL}(p||q) = 0$ только в случае, когда $p = q$.
  - ✓ $\mathrm{KL}(p||q) \neq \mathrm{KL}(q||p)$, т.е., дивергенция *не является симметричной*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Прямая (forward) дивергенция Кульбака-Лейблера

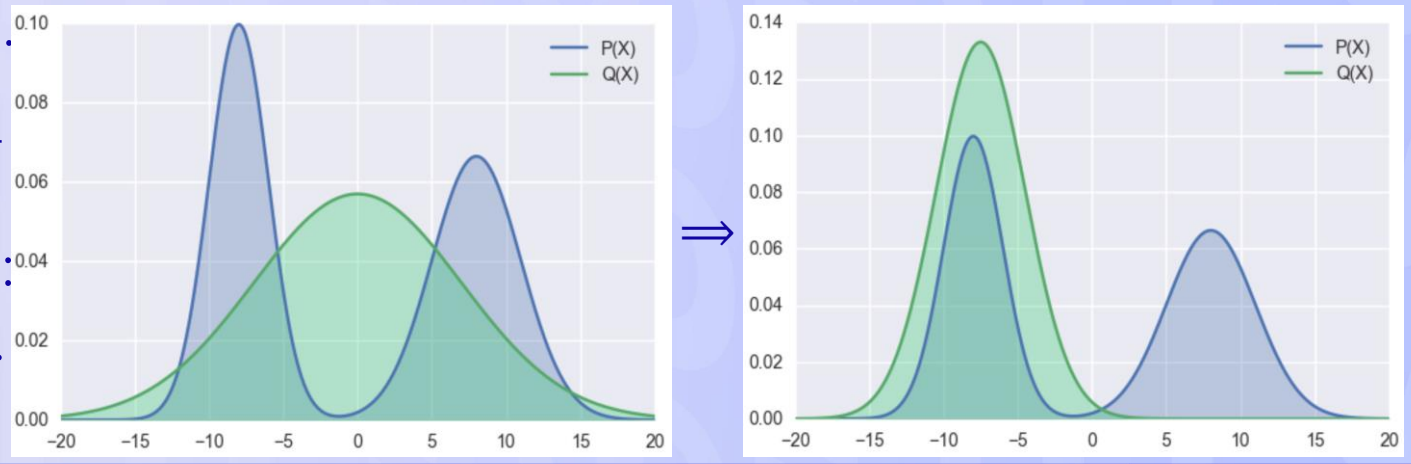
- Поскольку дивергенция Кульбака-Лейблера *не является симметричной*, возникает вопрос о том, чем отличаются *прямая* $\mathrm{KL}(p||q)$ и *обратная* $\mathrm{KL}(q||p)$.
- Для прямой дивергенции: $\mathrm{KL}(p||q) = \int \log \frac{p(x)}{q(x)} \cdot p(x)dx.$
- Видно, что если $p(x) = 0$, то величина $q(x)$ не влияет на значение дивергенции. Различие между распределениями $p(x)$ и $q(x)$ *имеет смысл определять*, если $p(x) > 0$.
- К прямой дивергенции применим термин **zero avoiding** (*избегание нуля*):
- $q(x) \neq 0$ там, где $p(x) = 0$.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Обратная (reverse) дивергенция Кульбака-Лейблера

- Для обратной дивергенции: $\mathrm{KL}(q||p) = \int \log \frac{q(x)}{p(x)} \cdot q(x)dx.$
- Теперь видно, что если $q(x) = 0$, то величина $p(x)$ не влияет на значение дивергенции. Различие между распределениями $p(x)$ и $q(x)$ *имеет смысл определять*, если $q(x) > 0$.
- Для *обратной дивергенции* является оптимальным соответствие $q(x)$ только некоторой части $p(x)$.
- К обратной дивергенции применим термин **zero forcing** (*принуждение нуля*):
- $q(x) = 0$ даже если $p(x) = 0$.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

- При решении задач используется **обратная дивергенция**, поскольку *матожидание* в ней рассчитывается по вариационному распределению $q(x)$, из которого *легко семплировать*:

$$\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z} \mid \mathbf{x})) = \mathbb{E}_{q(z)}\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z} \mid \mathbf{x})}\right].$$

- Таким образом, задача вариационного вывода сводится к экстремальной:

$$q(z): \quad \mathcal{L}[q(\mathbf{z})] = \mathrm{KL}[q(\mathbf{z}) \,||\, p(\mathbf{z} \mid \mathbf{x})] \to \min_{q(z) \in \mathcal{Q}}.$$

- Сущность «функция от функции» $\mathcal{L}[f(\mathbf{z})]$ называется **функционалом**.
- Раздел математики, изучающий задачи нахождения экстремумов функционалов, называется **вариационным исчислением**. Он и дал имя *рассматриваемому методу байесовского вывода*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Подход к оценке дивергенции Кульбака-Лейблера

- Таким образом, искомое *вариационное распределение* соответствует минимальному значению *обратной дивергенции* Кульбака-Лейблера.
- Однако, вычислить дивергенцию *напрямую* не получится, поскольку в её выражении содержится *неизвестное* целевое распределение $p(z \mid x)$.
- Перепишем *обратную дивергенцию Кульбака-Лейблера* таким образом:

$$\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z} \mid \mathbf{x})) = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z} \mid \mathbf{x})] = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})].$$

$$\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z} \mid \mathbf{x})) = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) .$$

- *Вероятность данных* $p(\mathbf{x})$ не зависит от $q(\mathbf{z})$, поэтому, *минимизация* дивергенции эквивалентна *максимизации* выражения:

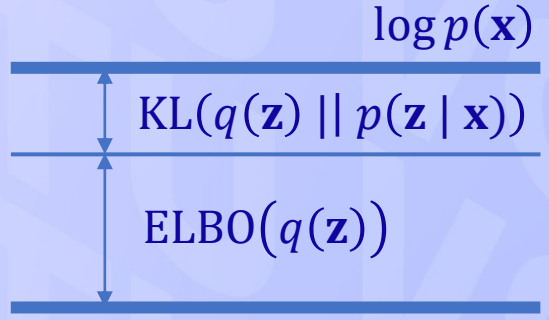
$$\mathrm{ELBO}\big(q(\mathbf{z})\big) = \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\mathbf{z})].$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Вариационная нижняя оценка (ELBO)

- Учитывая введенное обозначение, можно записать:

$$\log p(\mathbf{x}) = \mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z} \mid \mathbf{x})) + \mathrm{ELBO}\big(q(\mathbf{z})\big).$$



- Принимая во внимание, что $\mathrm{KL}(q||p) \geq 0$, получаем:

$$\log p(\mathbf{x}) \geq \mathrm{ELBO}\big(q(\mathbf{z})\big).$$

- Величина ELBO называется **вариационной нижней оценкой** *вероятности данных* (*evidence lower bound*).

- *Вариационная нижняя оценка* ограничивает вероятность данных *снизу*, а её *максимизация* эквивалентна *минимизации дивергенции КЛ*. В общем виде данная величина записывается следующим образом:

$$\mathrm{ELBO}\big(q(\mathbf{z})\big) = \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\mathbf{z})] = \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x}) - \log q(\mathbf{z})] = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}\right].$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Вариационные семейства. Mean-field

- Один из способов максимизации ELBO предполагает задание *вариационного распределения* следующего вида:

$$q_{\boldsymbol{\phi}}(\mathbf{z}) = \prod_{i=1}^{n} q_{\phi_i}(z_i)$$

- В данном представлении предполагается, что все латентные переменные *независимы друг от друга* и зависят только от *«своих»* параметров. Такое семейство распределений носит название **mean-field**.
- Для дальнейших расчетов требуется задать распределения каждой латентной переменной. Часто их выражают с помощью *экспоненциального семейства функций*:

$$q_{\phi_i}(z_i) = h(z_i) \cdot e^{\phi_i^T t(z_i) - a(\phi_i)}$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Первоначальный вариант стохастического вариационного инференса

- Дальнейший расчет предполагает *аналитическое вычисление* ELBO и производных от него с учетом введенных *предположений*.
- Затем *оценка параметров* вариационного распределения выполняется с помощью **алгоритма покоординатного спуска** (*coordinate ascent inference*). Данный алгоритм применяется до тех пор, пока ELBO не *сойдется*.
- На каждой итерации алгоритма из датасета произвольно выбирается один объект $\mathbf{x}$; так реализуется *стохастическая оптимизация*.
- Данный алгоритм SVI был *исторически первым*. Он вводил *серьёзные ограничения* на используемые вариационные распределения и требовал *проведения аналитических выкладок* для *каждого* класса распределений.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Black Box Variational Inference (BBVI)

- Если аналитическое выражение градиентов получить *очень сложно* или *невозможно*, то можно воспользоваться подходом к проведению VI, который *не требует знания внутренней структуры модели* (*BBVI*).
- Данный способ предполагает задание *вариационного распределения* в **параметрическом виде**: $q(\mathbf{z} \mid \boldsymbol{\varphi}) = q_{\boldsymbol{\varphi}}(\mathbf{z})$, что позволяет свести *максимизацию* ELBO к *задаче оптимизации*, похожей на задачу *обучения модели* в *классических* ML и DL:

$$\boldsymbol{\varphi}^{opt} = \arg\min_{\boldsymbol{\varphi} \in \Phi} \left\{ -\text{ELBO}\left(p(\mathbf{z} \mid \mathbf{x}), q_{\boldsymbol{\varphi}}(\mathbf{z}) \right) \right\} = \arg\min_{\boldsymbol{\varphi} \in \Phi} \left\{ -\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \right] \right\}.$$

- Хорошо бы провести *оценку градиента* с помощью метода Монте-Карло. Но для этого нужно *уметь вычислять производные от матожидания*...
- Попробуем свести *градиент от ELBO* к виду, удобному для оценки.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Анализ градиента от ELBO

- Вычислим градиент от ELBO по параметрам распределения $q_{\boldsymbol{\varphi}}(\mathbf{z})$:

$$\nabla_{\boldsymbol{\varphi}}\mathrm{ELBO} = \nabla_{\boldsymbol{\varphi}} \int \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} = \int \nabla_{\boldsymbol{\varphi}}\big[\big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot q_{\boldsymbol{\varphi}}(\mathbf{z})\big]d\mathbf{z} =$$

$$= \int \nabla_{\boldsymbol{\varphi}}\big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} + \int \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot \nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} =$$

- Учтём, что $\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{x},\mathbf{z}) = 0$. Тогда получим:

$$\nabla_{\boldsymbol{\varphi}}\mathrm{ELBO} = -\mathbb{E}_q\big[\nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big] + \int \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot \nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z}\,.$$

- Рассмотрим *первое слагаемое*:

$$\mathbb{E}_q\big[\nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big] = \int \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\cdot q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} = \int \frac{\nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\cdot q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} = \nabla_{\boldsymbol{\varphi}} \int q_{\boldsymbol{\varphi}}(\mathbf{z})d\mathbf{z} = \nabla_{\boldsymbol{\varphi}} 1 = 0.$$

- Тогда в выражении градиента останется *только второй интеграл*:

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Анализ градиента от ELBO

$$\nabla_{\boldsymbol{\varphi}}\mathrm{ELBO} = \int \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot \nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z}\,.$$

- Заметим, что $\nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}) = \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}) \cdot q_{\boldsymbol{\varphi}}(\mathbf{z})$. Тогда градиент примет вид:

$$\nabla_{\boldsymbol{\varphi}}\mathrm{ELBO} = \int \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}) \cdot q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z} =$$

$$= \mathbb{E}_q\big[\nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}) \cdot \big(\log p(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\varphi}}(\mathbf{z})\big)\big].$$

- Итак, градиент от ELBO представим в виде матожидания, оценка которого с помощью метода Монте-Карло выражается в виде *эстиматора*:

$$\nabla_{\boldsymbol{\varphi}}\mathrm{ELBO} \approx \frac{1}{S}\cdot \sum_{s=1}^{S} \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s) \cdot \big(\log p(\mathbf{x},\mathbf{z}_s) - \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s)\big), \qquad z_s \sim q_{\boldsymbol{\varphi}}(\mathbf{z}).$$

- А знание градиента позволяет применить *метод градиентного спуска*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Стохастический вариационный инференс

- *Классический вариационный инференс* предполагает расчет ELBO:

$$\text{ELBO} = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right].$$

- *Совместная плотность вероятности* $p(\mathbf{x}, \mathbf{z})$ вычисляется *по всему датасету* $\mathcal{D} = \{\mathbf{x}\}$, который может иметь *внушительные размеры*.
- С целью *масштабируемости* можно рассмотреть *факторизацию*:

$$p(\mathbf{x} \mid \mathbf{z}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{z}), \qquad \mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}.$$

- Это позволяет *за один раз* использовать *оценку ELBO по мини-батчу* $\mathcal{B}_M$:

$$\log p(\mathbf{x} \mid \mathbf{z}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \mathbf{z}) \approx \frac{N}{M} \sum_{\mathbf{x}_i \in \mathcal{B}_M} \log p(\mathbf{x}_i \mid \mathbf{z}), \qquad \mathcal{B}_M \subset \mathcal{D}, \qquad M = |\mathcal{B}_M|.$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Алгоритм Black Box Variational Inference

- С учетом всего вышерассмотренного, алгоритм **BBVI** принимает вид:
  1) Выбор *семейства параметризованного вариационного распределения* $q(\mathbf{z} \mid \boldsymbol{\varphi}) = q_{\boldsymbol{\varphi}}(\mathbf{z})$. Оно может быть выбрано в целях простоты из MeanField-семейства, но бывает и *любым другим*.
  2) Формулировка задачи максимизации ELBO как *задачи оптимизации*:

$$\boldsymbol{\varphi}^{opt} = \arg\min_{\boldsymbol{\varphi} \in \Phi} \left\{ -\mathrm{ELBO}\left(p(\mathbf{z} \mid \mathbf{x}), q_{\boldsymbol{\varphi}}(\mathbf{z})\right) \right\} = \arg\min_{\boldsymbol{\varphi} \in \Phi} \left\{ -\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \right] \right\}.$$

  3) Использование *оценки Монте-Карло градиентов ELBO*:

$$\nabla_{\boldsymbol{\varphi}} \mathrm{ELBO} \approx \frac{1}{S} \cdot \sum_{s=1}^{S} \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s) \cdot \left(\log p(\mathbf{x}, \mathbf{z}_s) - \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s) \right), \qquad z_s \sim q_{\boldsymbol{\varphi}}(\mathbf{z}).$$

  4) Повторение шагов *стохастического градиентного спуска* с целью максимизации ELBO со скоростью обучения ρ (*learning rate*) до сходимости алгоритма:

$$\boldsymbol{\varphi}^{t} = \boldsymbol{\varphi}^{t-1} + \rho \frac{1}{S} \cdot \sum_{s=1}^{S} \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s^t) \cdot \left[\frac{N}{M} \sum_{\mathbf{x}_i \in \mathcal{B}_M} \log p(\mathbf{x}_i \mid \mathbf{z}_s^t) - \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s^t) \right], \qquad \mathbf{z}_s^t \sim q_{\boldsymbol{\varphi}}(\mathbf{z}).$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Automatic Differentiation Variational Inference (ADVI)

- Алгоритм BBVI требует *ручного расчёта градиента* от $\log q_{\boldsymbol{\varphi}}(\mathbf{z})$.
- Но современные фреймворки *дифференцирования функций по графу* (*TensorFlow*, *PyTorch*) позволяют *автоматизировать эту процедуру*.
- С их помощью градиент от *произвольной гладкой функции* $f_{\boldsymbol{\varphi}}(\mathbf{x}, \mathbf{z})$, заданной с использованием *вычислительного графа*, эффективнее всего вычислить с помощью **алгоритма обратного распространения ошибки**, лежащего в основе обучения современных *нейронных сетей*.
- В качестве вариационных распределений можно взять *классические*, *широко известные* семейства (*MeanField Gaussian*, *Full-rank Gaussian*).
- Такой *автоматизированный подход к вариационному выводу*, который *минимизирует вмешательство пользователя*, носит название **ADVI**.

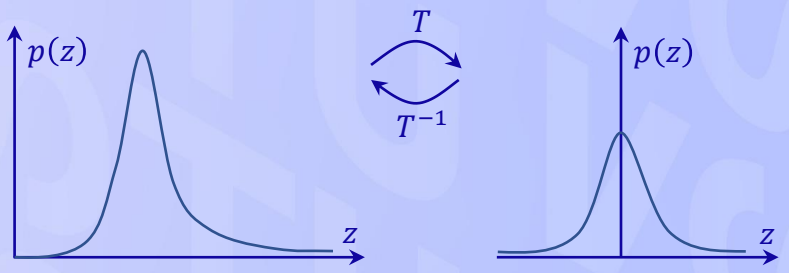
VOLGOGRAD STATE TECHNICAL UNIVERSITY

Автоматическое преобразование ограниченных переменных

- Если область значений переменной *ограничена*, то при градиентном спуске могут возникать *численные нестабильности и неустойчивости*.
- *Область возможных значений* вариационного распределения *должна соответствовать ограничениям* параметров, что *усложняет его подбор*.
- Для решения этих проблем ADVI использует *автоматическое преобразование ограниченных переменных* с помощью *биективных преобразований* T:

$$T : \mathrm{supp}\big(p(\mathbf{z})\big) \to \mathbb{R}^K.$$

- После этого оптимизация проводится в *неограниченном пространстве*, что *упрощает* вычисления и делает их *более стабильными*.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Примеры преобразований

- При *преобразовании* $\boldsymbol{\xi} = T(\mathbf{z})$ *плотность распределения* изменяется следующим образом:

$$p(\mathbf{x}, \boldsymbol{\xi}) = p\left(\mathbf{x}, T^{-1}(\boldsymbol{\xi})\right) \cdot \left|\det\left(\frac{\partial T^{-1}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right)\right|.$$

- К числу наиболее *широко применяющихся преобразований* относятся:

Название преобразования	Область значений аргумента	Функция преобразования	Область значений функции
Логарифмическое	$(0, +\infty)$	$\xi = \log(z)$	$(-\infty, +\infty)$
Логистическое	$(0, 1)$	$\xi = \log\left(\frac{z}{1-z}\right)$	$(-\infty, +\infty)$
Обобщённое логистическое	(a, b)	$\xi = \log\left(\frac{z-a}{b-z}\right)$	$(-\infty, +\infty)$

- В *Pyro преобразования* применяются для переменных с `constraints`.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Алгоритм Automatic Differentiation Variational Inference

1) Выбор *семейства параметризованного вариационного распределения* $q(\mathbf{z} \mid \boldsymbol{\varphi})$. Чаще всего используется *MeanField Gaussian* или *Full-rank Gaussian*: $q(\mathbf{z} \mid \boldsymbol{\varphi}) = q(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma})$.
2) *Подбор преобразований* $\boldsymbol{\xi} = T(\mathbf{z})$ для преобразования ограниченной переменной $\mathbf{z}$ в $p(\mathbf{x}, \mathbf{z})$ в неограниченную $\boldsymbol{\xi}$.
3) Стандартизация вариационного распределения, например: $\boldsymbol{\eta} = S_{\boldsymbol{\sigma},\boldsymbol{\mu}}(\boldsymbol{\xi}) = \mathrm{diag}(\exp(\boldsymbol{\sigma}))^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu})$.
4) Формулировка ELBO:

$$ELBO(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\eta};\, \mathbf{0}, \mathbb{I})} \left[\log p \left[\mathbf{x}, T^{-1}\left(S^{-1}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{\eta})\right)\right] + \log \left| J_{T^{-1}}\left(S^{-1}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{\eta})\right)\right| - \log\left(q\left(S^{-1}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{\eta})\right)\right)\right].$$

5) Вычисление *градиентов ELBO* с помощью *алгоритма обратного распространения ошибки*.
6) Повторение шагов *стохастического градиентного спуска* с целью максимизации ELBO со скоростью обучения ρ (*learning rate*) до сходимости алгоритма:

$$(\boldsymbol{\mu}, \boldsymbol{\sigma})^t = (\boldsymbol{\mu}, \boldsymbol{\sigma})^{t-1} + \rho \cdot \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} ELBO(\boldsymbol{\mu}, \boldsymbol{\sigma}), \qquad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta};\, \mathbf{0}, \mathbb{I}).$$

7) Формирование вариационного распределения:

$$q(\boldsymbol{\xi}) = S^{-1}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{\eta}), \qquad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta};\, \mathbf{0}, \mathbb{I}).$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Рекомендации по организации вариационного инференса в Pyro

- Начинать *вариационный инференс* следует с *небольших значений learning rate* ($10^{-3} - 10^{-4}$), увеличивая его по мере необходимости после получения адекватных результатов.
- По умолчанию следует использовать *оптимизаторы* *Adam* или *ClippedAdam*. Для *стохастических* моделей имеет смысл *увеличить значения коэффициентов момента* для Adam.
- Следует использовать алгоритмы *уменьшения скорости обучения* в процессе инференса (*decaying learning rate*), что позволит *улучшить сходимость*:

```
lrd = gamma ** (1 / num_steps)
optim = pyro.optim.ClippedAdam({'lr': initial_lr, 'lrd': lrd})
```

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Рекомендации по организации вариационного инференса в Pyro

- Следует удостовериться, что *модель* (*model*) и *вариационное распределение* (*guide*) имеют одинаковый support:

$$\mathrm{supp}(p) = \{x \in X : p(x) \neq 0\}.$$

- Параметры, которые могут принимать значения только из определенного множества, должны быть *ограничены* (*constrained*). Ограничение параметров Pyro вводится посредством параметра при его определении:

```
scale = pyro.param("scale", torch.tensor(0.05), constraint=constraints.positive)
```

- При создании *вариационного распределения* (*guide*) можно воспользоваться классом AutoGuide. Данный класс предлагает к использованию алгоритмы *автоматического построения вариационного распределения* на основе определенных *семейств* распределений.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

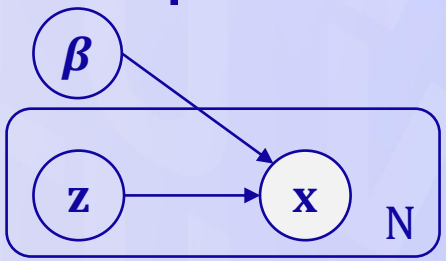
Стохастический вариационный инференс в Pyro

- *Стохастический вариационный инференс* в **Pyro** является основным алгоритмом инференса. Он реализуется с помощью *класса* **SVI**.
- Данный класс при создании принимает *параметры*, определяющие его работу:
  - ✓ **model** – *модель* Pyro;
  - ✓ **guide** – *вариационное распределение* в контексте модели;
  - ✓ **optim** – *оптимизатор*, реализующий один из алгоритмов SGD;
  - ✓ **loss** – *функция ошибки*. Как правило, один из вариантов ELBO:
    - ➢ Trace_ELBO;
    - ➢ TraceGraph_ELBO;
    - ➢ TraceMeanField_ELBO, и т.д.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Глобальные и локальные латентные переменные

- Вероятная модель может содержать *латентные* переменные двух типов.
- Если *латентная* переменная связана только с одним объектом в датасете, она называется *локальной*. На *распределение* локальной переменной влияет только один объект.
- Если *латентная* переменная связана со всеми объектами в датасете, она называется *глобальной*. На распределение глобальной переменной влияют все объекты.
- Пример: латентная переменная $\boldsymbol{\beta}$ является *глобальной*, а переменные $\mathbf{z}_i$ – *локальными*.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Переход к амортизированному вариационному инференсу

- Если модель содержит *локальные переменные*, то она плохо масштабируется, так как при увеличении количества объектов в датасете количество локальных латентных переменных тоже *увеличивается*.
- Более того, наблюдение одного объекта из датасета *не зависит от наблюдений* других объектов, что приводит к невозможности использовать информацию из *предыдущих* наблюдений в *последующих*.
- В случае *больших датасетов* вычислительная эффективность *падает*, поскольку результаты разных наблюдений *не могут переиспользоваться*.
- Стандартный подход к решению проблемы увеличивающегося датасета – переход к обработке данных *по батчам*. Однако, применение этого подхода в данном случае осложняется наличием локальных переменных.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Переход к амортизированному вариационному инференсу

- Как *решать* проблему?
- Можно перейти от множества локальных переменных к *глобальной* – но это приведёт к снижению точности, так как не будут учитываться *особенности* отдельного объекта.
- С другой стороны, вариационные распределения локальных переменных отличаются друг от друга только набором параметров, определяемых *отдельно для каждого объекта*.
- *Идея*: вместо того, чтобы определять *свой набор параметров* для каждой локальной переменной, определим *функцию*, которая по *признаковому описанию объекта* сможет формировать параметры распределения *соответствующей локальной переменной*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Амортизированный вариационный инференс

- Полученный *вариационный инференс* называется *амортизированным*, поскольку предполагает «*амортизацию*» процесса оптимизации.
- Вместо того, чтобы оптимизировать *каждый объект* датасета *отдельно*, стоимость оптимизации *распределяется между несколькими объектами*, что снижает общую вычислительную нагрузку при выполнении *инференса*.
- Для *амортизированного* инференса можно записать:

$$q(\mathbf{z}) = q(\mathbf{z} \mid f(\mathbf{x})) = q(\mathbf{z} \mid \mathbf{x}), \qquad \text{ELBO} = \mathbb{E}_{q(x)}\left[\frac{\log p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\right].$$

- Функция $f(\mathbf{x})$ определяет зависимость *параметров вариационного распределения* от *данных* и обычно выражается нейронной сетью, которая носит название inference network.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Организация амортизированного вариационного инференса в Pyro

- Для организации *амортизированного вариационного инференса* в Pyro требуется *определить условную независимость* между локальными переменными и нейросетевую функцию для *оценки параметров*.
- В Pyro *условная независимость* реализуется с помощью примитива plate, в конструкторе которого необходимо указать *размер датасета*. Глобальные переменные определяются *вне конструкции* plate.

```
def model(data):
    #семплирование глобальной переменной
    beta = pyro.sample("beta", ...)
    for i in pyro.plate("locals", len(data)):
        z_i = pyro.sample("z_{}".format(i), ...)
        theta_i = compute_something(z_i)
        pyro.sample("obs_{}".format(i), dist.MyDist(theta_i), obs=data[i])
```

- В Pyro определены два способа определения *условной независимости* с использованием plate: *последовательный* и *векторный*:

```
for i in pyro.plate("data_loop", len(data), subsample_size=5):
    pyro.sample("obs_{}".format(i), dist.Bernoulli(f), obs=data[i])

with pyro.plate('observe_data', size=10, subsample_size=5) as ind:
    pyro.sample('obs', dist.Bernoulli(f), obs=data.index_select(0, ind))
```

- При *необходимости* можно указать размер *мини-батча*. Он указывается только при определении *вариационного распределения* (*guide*). В *модель* (*model*) нужный размер подставится *автоматически* во время *инференса*:

```
with pyro.plate('observe_data', size=10) as ind:
    pyro.sample('obs', dist.Bernoulli(f), obs=data.index_select(0, ind))
```

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Демонстрация практических примеров

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Заключение

1. Рассмотрели вариационный инференс – альтернативный подход к решению задачи байесовского вывода.
2. Сформулировали идеи и подходы, лежащие в основе VI.
3. Поговорили по дивергенцию Кульбака-Лейблера и вариационную нижнюю оценку, выяснили их роль в решении задачи инференса.
4. Определили понятие стохастического вариационного инференса.
5. Рассмотрели варианты BBVI и ADVI, разобрали их особенности и алгоритмы работы.
6. Обговорили и рассмотрели на практических примерах реализацию классического и амортизированного вариационного инференса во фреймворке Pyro.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Спасибо за внимание!

Волгоград 2025