в бм известно два направления приближение: монте-карло и вар. вывод.

	MCMC	magic	VI
Bias	No	magic	Large
Sampling/Ensembling	Inefficient	magic	Efficient
Density	No	magic	Yes

Нельзя ли сделать что-то промежуточное? Ага. (S)IPM (Implicit Probabilistic Model).

Пусть есть DNN, которая принимает нормальный шум и выплевывает что-то со сложным распределением. Объекты генерировать можем, а плотность мы не знаем.

Почему не норм. потоки? Тяжело обучать и не очень гибкие. Слои должны быть такие, что якобиан считается за линейное время.

Здесь же мы снимаем эти ограничения.

Какие недостатки?

	MCMC	(S)IPM	VI
Bias	No	Small	Large
Sampling/Ensembling	Inefficient	Efficient	Efficient
Density	No	No	Yes
Likelihood	Explicit	Likelihood-free	Explicit

Самый известный пример — GAN. Как они работают?

$$X = (x_1 \dots x_n)$$
$$\xi \sim \mathcal{N}(\xi \mid 0, I) \to \text{Generator}_\theta \to X_{\text{syn}}$$
$$x_{\text{syn}}, x_i \to \text{Discriminator}_\eta \to \mathbb{P}\{x \in \text{real}\}$$
$$D(x) = \mathbb{P}\{x \in \text{real}\}$$
$$\eta = \arg\max_\eta \left(\frac{1}{n} \sum_{i=1}^{n} \log D(x_i) + \mathbb{E}_\xi \log\left(1 - D_\eta\left(G_\theta(\xi)\right)\right) \right)$$
$$\theta = \arg\max_\theta \left(-\mathbb{E}_\xi \log\left(1 - D_\eta\left(G_\theta\left(\xi\right)\right)\right)\right)$$

Формально так, но лучше обучать чуть-чуть по-другому:

$$\theta = \arg\max_\theta \left(\mathbb{E}_\xi \log D_\eta\left(G_\theta\left(\eta\right)\right)\right)$$

Почему? Раньше был единый функционал, сейчас же две разные оптимизационные задачи. Идейно то же самое.

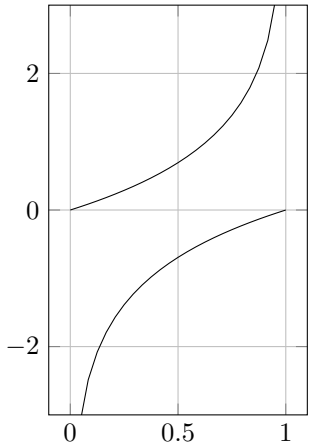


Схема крайне неустойчива. Попытки решения минимакса стохастическими методами приводят ко многим проблемам. Первые три года это было проблемой. Теперь менее. Но все равно, это заметно хуже по стабильности, чем вариационные автокодировщики. Это не означает, что они лучше.

Анализ. Пусть у нас есть идеальный дискриминатор:

$$p_{\text{real}}(x) \qquad p_{\text{syn}}(x)$$
$$D_*(x) = \frac{p_{\text{real}}(x)}{p_{\text{syn}}(x) + p_{\text{real}}(x)}$$

Посмотрим, какой функционал оптимизуруется. Заметим, что ид. дискр. зависит от генератора.

$$\theta = \arg\min_\theta \left(\frac{1}{n} \sum_{i=1}^{n} D_*(x_i) + \mathbb{E}_\xi \log(1 - D_*(G(\xi))) \right)$$

$$\approx \arg\min_\theta \int p_{\text{real}}(x) \log \frac{p_{\text{real}}(x)}{p_{\text{real}}(x) + p_{\text{syn}}(x)} dx + \int p_{\text{syn}}(x) \log \frac{p_{\text{syn}}(x)}{p_{\text{syn}}(x) + p_{\text{real}}(x)} fx =$$

$$= \arg\min_\theta \int p_{\text{real}}(x) \log \frac{p_{\text{real}}(x)}{\frac{p_{\text{real}}(x)+p_{\text{syn}}(x)}{2}} dx + \int p_{\text{syn}}(x) \log \frac{p_{\text{syn}}(x)}{\frac{p_{\text{real}}(x)+p_{\text{syn}}(x)}{2}} dx - \log 4 =$$

$$\arg\min_\theta \left[\mathrm{KL}(p_{\text{real}} \| \frac{p_{\text{real}} + p_{\text{syn}}}{2}) + \mathrm{KL}(\frac{p_{\text{real}} + p_{\text{syn}}}{2} \| p_{\text{real}}) - \log 4 \right] =$$

$$= \arg\min_\theta \left[2\mathrm{JS}(p_{\text{real}} \| p_{\text{sym}}) \right]$$

Это нанесло больше вреда, чем пользы. Тут неправильно почти все. Начиная с того, что ни доступа к ид. дискр. у нас нет, да выборка конечная.

Далеко не всегда маленькое значение дивергенции означает близость, которая нам нужна. Мы используем самую гибкую конструкцию — DNN. Если и она не может различить, то все одинаково. То есть мы выучиваем нашу дивергенцию.

Сам аппарат Adversarial Training гораздо более мощный, чем просто GAN. Суть в минимизации лосса дискриминатора. Из-за этого в любой модели, использующей AT, есть минимакс. Это плохо. это неустойчиво.

Основные механизмы:

- Обучаем генератор по конечной выборке
  Если слишком гибкий дискр., он может тупо переобучиться на конечную выборку. Это довольно печально. Максимум — мы научим генерировать из исходной выборки. Мы тогда пытаемся загрублять д. Но тогда мы не можем надеятся на идеальность тем более.

- Обучаем генератор по генератору
  Отличие? Мы имеем возможность получать бесконечную выборку. На всех операциях объекты будут уникальны. Риск переобучения снижается. Например, пусть генератор — MCMC. Тогда мы можем обучить что-то быстрее MCMC, другой генератор. Или например, дистилляция/transfer learning

- Обучаем генератор по плотности
  У нас есть доступ к ненормированной плотности. Отличие? В предыдущей не предполагалась плотность.

GAN лежат только в первой постановке.

Две наиболее поп. ген. модели: vae, gan.

Нельзя совместить?

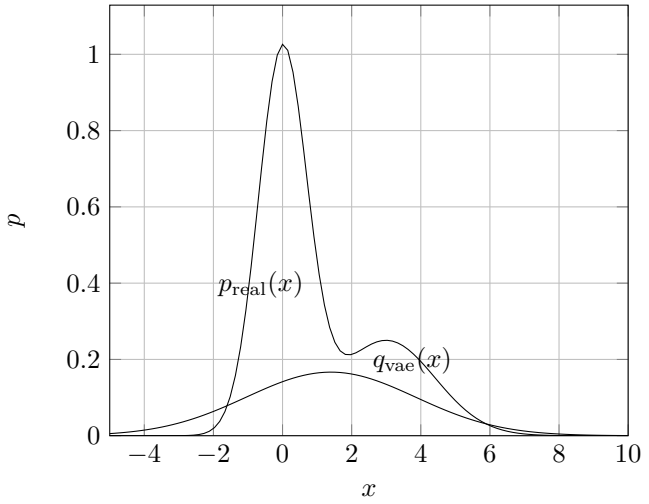
Что плохого в GAN? Мы никогда не поощряем генерацию всего. Поэтому mode collapsing. Не тотальный, но штраф довольно слабый.

В VAE же мы хотим генерацию всего:

$$\log p(X \mid \theta) \geq \mathcal{L}(\phi, \theta) =$$

$$= \sum_{i=1}^{n} \int q(z \mid x_i, \phi) \log p(x_i \mid z, \theta) dz - \mathrm{KL}(q(z|x_i, \phi) \| p(z)) \to \max_{\theta, \phi}$$

Здесь mode collapsing явно не может возникнуть.



Давайте использовать неявные модели в энкодере.
AAE.

$$q(z) = \int q(z \mid x, \phi) p(x) dx \approx p(z)$$

рисунок про дыры и покрытия в vae
пусть кодировщик — неявная вер. модель. Будем z из энк. и из априорного подавать дискриминатору, который будет выплевывать вероятность $\mathbb{P}\{z \sim p(z)\}$
Можем делать 3 разные модели:

1. сам вае
2. детерминированный
3. с впрыскиванием шума

Все три модели покрываются AAE.

$$\eta = \arg\max_{\eta} \left[\int p(z) \log D_\eta(z) dz + \frac{1}{n} \sum_{i=1}^{n} \int q(z \mid x_i, \phi) \log(1 - D(z)) dz \right]$$

$$\phi = \arg\max_{\phi} \sum_{i=1}^{n} \left[\int q(z \mid x_i, \phi) \log p(x_i \mid z, \theta) dz + \lambda \int q(z \mid x_i, \phi) \log D(z) dz \right]$$

Мы отказались от ELBO, теперь есть две части. Надо как-то балансировать: добавили λ. Это эвристика, но из соображений здравого смысла.
AVB

Итак. Возвращаемся к модели вае

$$\log p(X \mid \theta) \geq \mathcal{L}(\phi, \theta) =$$

$$= \sum_{i=1}^{n} \int q(z \mid x_i, \phi) \log p(x_i \mid z, \theta) dz - \mathrm{KL}(q(z|x_i, \phi) \| p(z)) \to \max_{\theta, \phi}$$

$$\mathrm{KL}(q\|p) = \int q(z|x_i, \phi) \log \frac{q(z|x_i, \phi)}{p(z)} dz$$

Допустим, мы обучили дискр, который предсказывает это отношение плотностей. Супер.

$$\text{Class1} : p(z); \quad \text{Class2} : q(z);$$

$$D_*(z) = \frac{p(z)}{p(z) + q(z)}$$

$$\frac{p(z)}{q(z)} = \frac{D_*(z)}{1 - D_*(z)}$$

$$D_\eta(x, z) : p(x)p(z) \text{vs.} p(x)q(z \mid x)$$

$$\eta = \arg\max_{\eta} \sum_{i=1}^{n} \left(\int p(z) \log D(x, z) dz + \int q(z|x_i, \phi) \log(1 - D(x_i, z)) dz \right)$$

$$\frac{D_\eta(x, z)}{1 - D_\eta(x, z)} \approx \frac{p(x)p(z)}{p(x)q(z \mid x)} = \frac{p(z)}{q(z \mid x)}$$

Получаем AVB

$$\theta = \arg\max_{\theta} \sum_{i=1}^{n} \int q(z|x_i, \phi) \log p(x_i \mid z, \theta) dz$$

$$\phi = \arg\max_{\phi} \sum_{i=1}^{n} \left[\int q(z \mid x_i, \phi) \log p(x_i \mid z, \theta) dz + \int q(z \mid x_i, \phi) \log \frac{D_\eta(x_i, z)}{1 - D_\eta(x_i, z)} dz \right]$$

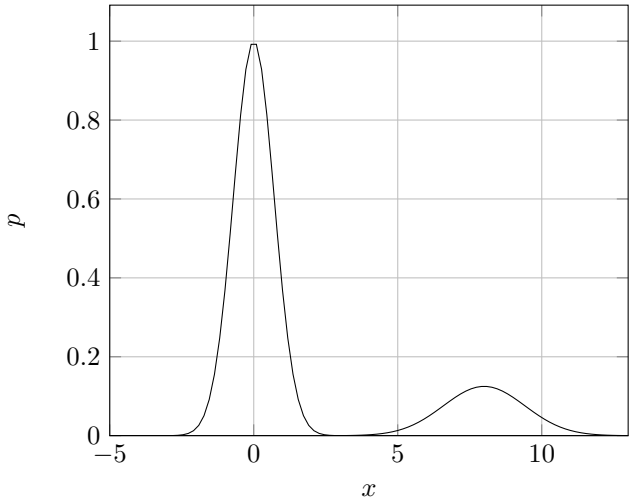
Это первый случай, когда мы не пытаемся обмануть дискриминатор. По сути, у нас должна быть ELBO, но мы KL посчитали через дискриминатор.

Да, мы рискуем, тк оптимизируем приближение. Проблема не закрыта, модель не идеальна.

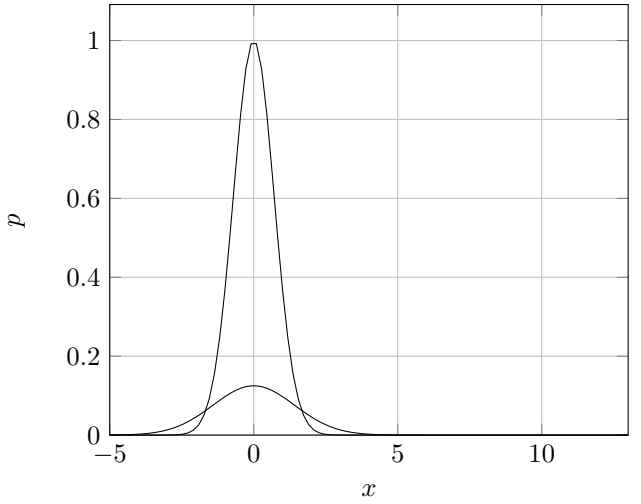
Встречаются модели с AT, в которых мы не пытаемся обмануть дискриминатор.

Возникает вопрос. Допустим, что дискр. обучился, а как его оценить? Какими свойствами должно обладать отношение, чтобы знать, что дискр. обучится хорошо.

Есть отличный прием



Так отношение плотностей чудовищно плохо оценивается.



А так супер.

Возникает идея. (опять рисунок с латентным пространством)

Носитель сильно отличается. Можно сделать лучше?

Заметим, что

$$\frac{p(z)}{q(z \mid x, \phi)} = \frac{p(z)}{r(z \mid x, \alpha)} \frac{r(z \mid x, \alpha)}{q(z \mid x, \phi)}$$

r — какое-то простое распределение, например, полностью факт. гауссиану. Вторую часть оцениваем DRE (Density Ration Estimation).

Это называется Adaptive Contrast.

$$r(z \mid x, \alpha) = \prod_{j=1}^{d} \mathcal{N}\left(z^j \mid \mu^j(x, \alpha), \left(\sigma^j(x, \alpha)\right)^2\right)$$

Как обучать?

$$\alpha = \arg\min_{\alpha} \sum_{i=1}^{n} \left\| \begin{bmatrix} \mu(x_i, \alpha) \\ \log \sigma(x_i, \alpha) \end{bmatrix} - \begin{bmatrix} \hat{\mu}(x_i, \phi) \\ \log \hat{\sigma}(x_i, \phi) \end{bmatrix} \right\|^2$$

$\hat{\mu}, \hat{\sigma}$ — считаем по выборке из $q(z \mid x_i, \phi)$

Если не жалко памяти, можно r и без сетки учить, для каждого объекта.

Вот такая вот модель. Она наиболее близко подошла к объединению vae и gan.