Порождающие модели

Московский Физико-Технический Институт

2021

Генеративные и дискриминативные модели

Разделяющие модели

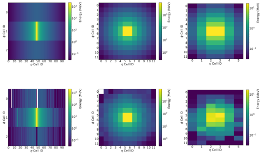
Моделируют: $p(y|\boldsymbol{x})$.

Порождающие модели

Моделируют: $p(y, \boldsymbol{x})$.

Порождающие модели:

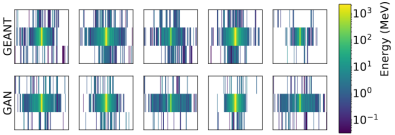
- Порождение новых элементов выборки (когда генерация — самоцель)
- Создание синтетических данных для обучения/дообучения
- Получение скрытых свойств выборки (например, через латентные переменные)

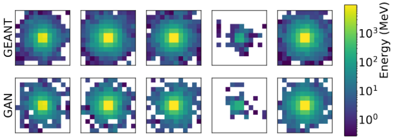


Порождение данных: пример

Paganini et al., 2017:

- Моделируются энергии частиц, попадаемых в калориметр
- Для моделирования используется GAN
- Сравнение происходит не с реальными данными, а со специализированным ПО GEANT
- Итог: качество приемлемое, но генерация происходит быстрее в 100-1000 раз

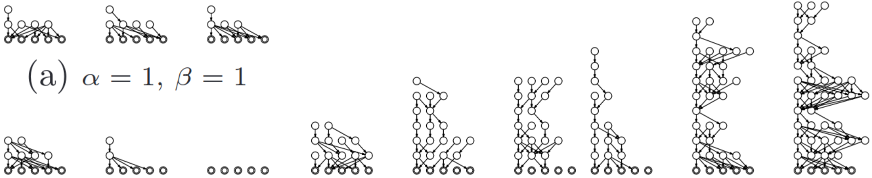




Порождение моделей: пример

Adams et al., 2010:

- Порождаются глубокие сети доверия (Deep belief networks)
- структура модели **Г** — последовательность матриц инцидентности для каждого слоя
- Порождение через Монте-Карло с использованием процесса индийского буффета в качестве априорного с параметрами α, β
- Интерпретация параметров: ширина и разреженность структуры



(a) $\alpha = 1,\ \beta = 1$

(b) $\alpha = \frac{1}{2},\ \beta = 1$

(c) $\alpha = 1,\ \beta = 2$

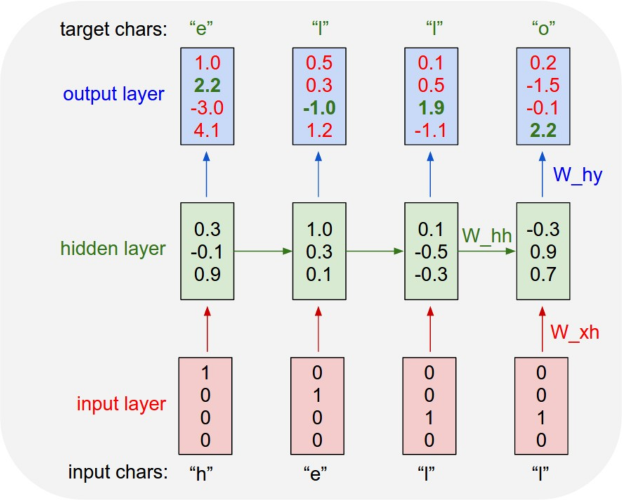
(d) $\alpha = \frac{3}{2},\ \beta = 1$

Как строить порождающие модели?

- **Метод 1:** задать явную функцию правдоподобия (“Fully-observed likelihood”), декомпозирующую правдодобие всего объекта на мелкие части (“Autoregressive models”).

Пример: CharRNN

Karpathy, 2015



Как строить порождающие модели?

- **Метод 1:** задать явную функцию правдоподобия (“Fully-observed likelihood”), декомпозирующую правдодобие всего объекта на мелкие части (“Autoregressive models”).
  Проблемы:
  - сложно назначить адекватную функцию правдоподобия.
  - вычислительно сложный вывод.

Как строить порождающие модели?

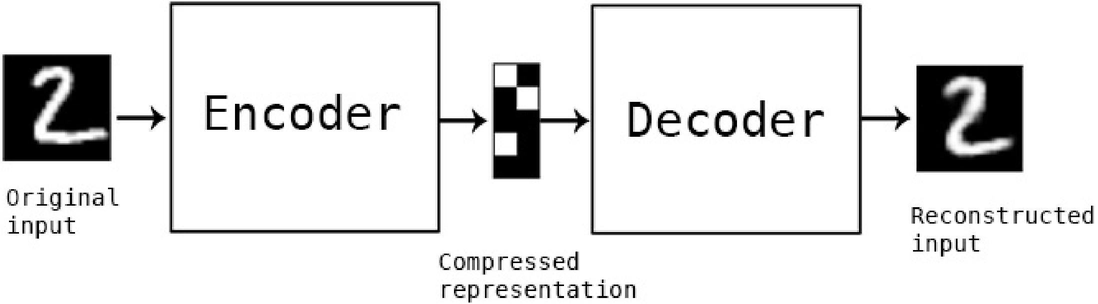
- **Метод 1:** задать явную функцию правдоподобия (“Fully-observed likelihood”), декомпозирующую правдодобие всего объекта на мелкие части (“Autoregressive models”).
  Проблемы:
  - сложно назначить адекватную функцию правдоподобия.
  - вычислительно сложный вывод.
- **Метод 2:** ввести предположение, что объекты порождаются скрытой переменной, исследовать свойства которой значительно проще (“Latent variable models”).

Пример: автокодировщик

Автокодировщик — модель снижения размерности:

$$\boldsymbol{H} = \boldsymbol{\sigma}(\boldsymbol{W}_e \boldsymbol{X}),$$

$$||\boldsymbol{\sigma}(\boldsymbol{W}_d \boldsymbol{H}) - \boldsymbol{X}||_2^2 \to \min .$$



Автокодировщик: порождающая модель?

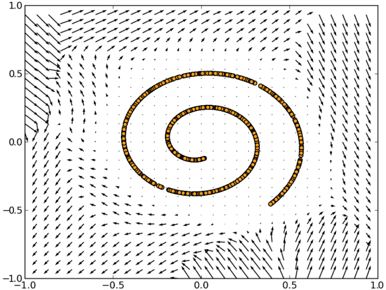
(Alain, Bengio 2012): рассмотрим модель автокодировщика с регуляризацией:

$$||\boldsymbol{f}(\boldsymbol{x}, \sigma) - \boldsymbol{x}||^2,$$

где σ — уровень шума, подаваемого на вход модели кодирования. Тогда

$$\frac{\partial \log p(x)}{\partial x} = \frac{||\boldsymbol{f}(\boldsymbol{x}, \sigma) - \boldsymbol{x}||^2}{\sigma^2} + o(1) \text{ при } \sigma \to 0.$$

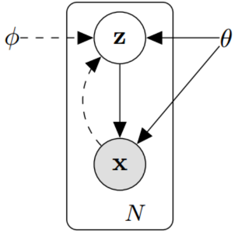
Векторное поле, индуцированное ошибкой реконструкции автокодировщика



Вариационный автокодировщик

Пусть объекты выборки $\boldsymbol{X}$ порождены при условии скрытой переменной $\boldsymbol{h} \sim \mathcal{N}(0, \boldsymbol{I})$:

$$\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{w}).$$



$p(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{w})$ — неизвестно.
Будем максимизировать вариационную оценку правдоподобия выборки:

$$\log p(\boldsymbol{x}|\boldsymbol{w}) \geq \mathsf{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{w}) - D_{\mathrm{KL}}(q_\phi(\boldsymbol{h}|\boldsymbol{x})||p(\boldsymbol{h})) \to \max.$$

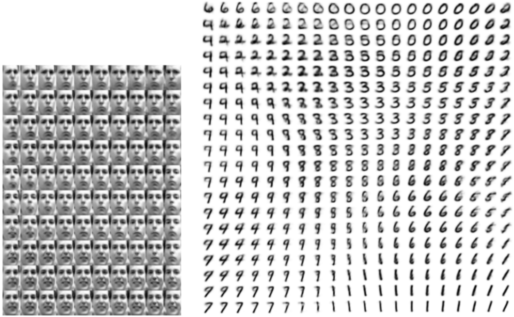
Распределения $q_\phi(\boldsymbol{h}|\boldsymbol{x})$ и $p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{w})$ моделируются нейросетью:

$$q_\phi(\boldsymbol{h}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})),$$

$$p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{w}) \sim \mathcal{N}(\boldsymbol{\mu}_w(\boldsymbol{h}), \boldsymbol{\sigma}^2_w(\boldsymbol{h})),$$

Вариационный автокодировщик: процесс порождения

Процесс порождения заключается в сэмплировании скрытой переменной из априорного расспределения: $\boldsymbol{z} \sim p(\boldsymbol{z})$ и действии на него декодером.



(a) Learned Frey Face manifold (b) Learned MNIST manifold

Как строить порождающие модели?

- **Метод 1:** задать явную функцию правдоподобия (“Fully-observed likelihood”), декомпозирующую правдодобие всего объекта на мелкие части (“Autoregressive models”).
  Проблемы:
  - сложно назначить адекватную функцию правдоподобия.
  - вычислительно сложный вывод.
- **Метод 2:** ввести предположение, что объекты порождаются скрытой переменной, исследовать свойства которой значительно проще (“Latent variable models”).
  Проблемы:
  - $p(\boldsymbol{x})$ не вычислимо аналитически
- Проблема обоих методов: высокое правдоподобие и хорошее качество сэмплирования могут быть не взаимосвязаны (Theis et al., 2015).
- Пусть задана шумовая смесь моделей:

$$p_w(x) = 0.01 p_{\text{data}}(x) + 0.99 p_{\text{noise}}(x), \log p_w(x) \geq \log p_{\text{data}}(x) - \log 100$$

- в другую сторону: переобучение.

Как строить порождающие модели?

- **Метод 1:** задать явную функцию правдоподобия (“Fully-observed likelihood”), декомпозирующую правдодобие всего объекта на мелкие части (“Autoregressive models”).
  Проблемы:
  - сложно назначить адекватную функцию правдоподобия.
  - вычислительно сложный вывод.
- **Метод 2:** ввести предположение, что объекты порождаются скрытой переменной, исследовать свойства которой значительно проще (“Latent variable models”).
  Проблемы:
  - $p(\boldsymbol{x})$ не вычислимо аналитически
- **Метод 3:** отказаться от правдоподбия и работать напрямую с порождением и отлонением порожденных объектов (по сути: переход к стат-критериям типа критерия отношения правдоподобия).

Генеративно-состязательные сети (Goodfellow et al., 2014)

Общий принцип: тренируем две модели, генератор G и дискриминатор D:

$$\min_{\boldsymbol{W}_G} \max_{\boldsymbol{w}_D} \mathrm{E}_{\boldsymbol{x} \in \mathfrak{D}} \log p(\boldsymbol{x} | \boldsymbol{w}_D, D) + \mathrm{E}_{\boldsymbol{x} \in p_G} \log(1 - p(\boldsymbol{x} | \boldsymbol{w}_D, D)).$$

Алгоритм оптимизации итеративной

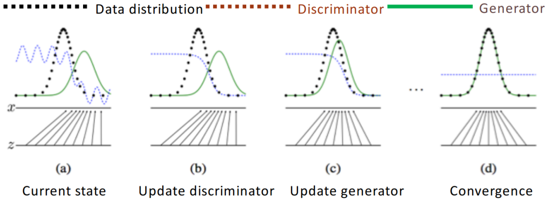
- $\mathrm{E}_{\boldsymbol{x} \in \mathfrak{D}} \log p(\boldsymbol{x} | \boldsymbol{w}_D, D) \to \max_{\boldsymbol{w}_D}$
- $\mathrm{E}_{\boldsymbol{x} \in p_G} \log(1 - p(\boldsymbol{x} | \boldsymbol{w}_D, D)) \to \min_{\boldsymbol{w}_G}$
- Альтернатива: $\mathrm{E}_{\boldsymbol{x} \in p_G} \log p(\boldsymbol{x} | \boldsymbol{w}_D, D) \to \max_{\boldsymbol{w}_G}$

Генеративно-состязательные сети: оптимальность

При достижении дискриминатором глобального оптимума, генератор минимизирует JS:

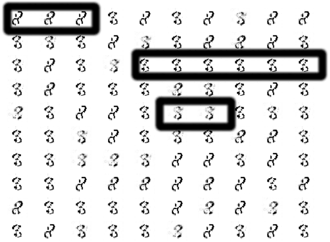
$$-\log(4) + KL\left(p(\mathbf{x}|\frac{p(\mathbf{x}) + p_G(\mathbf{x})}{2})\right) + KL\left(p_G\mathbf{x}|\frac{p(\mathbf{x}) + p_G(\mathbf{x})}{2})\right) \to \min_{\mathbf{w}_G}.$$

Следствие: оптимальное распределение генератора: $p_G = p(\mathbf{x})$.



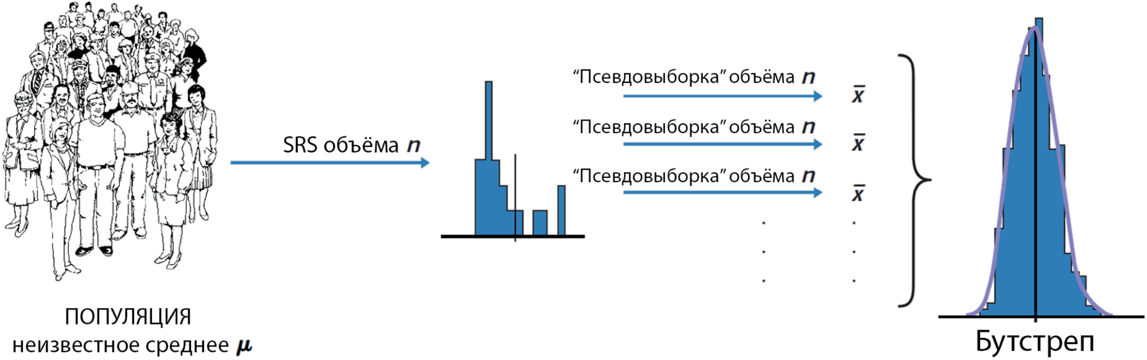
Особенности оптимизации GAN

- Оптимизация генератора может производиться в двух режимах: $\mathsf{E}_{\boldsymbol{x}\in p_G}\log(1-p(\boldsymbol{x}|\boldsymbol{w}_D,D))\to\min_{\boldsymbol{w}_G}$ или $\mathsf{E}_{\boldsymbol{x}\in p_G}\log p(\boldsymbol{x}|\boldsymbol{w}_D,D)\to\max_{\boldsymbol{w}_G}$: оптимум совпадает, но градиент в первом случае значительно более пологий.
- Генератор может сойтись в локальный экстремум и генерировать однотипные объекты (mode collapse).



https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/

Порождение данных: бутстрэп

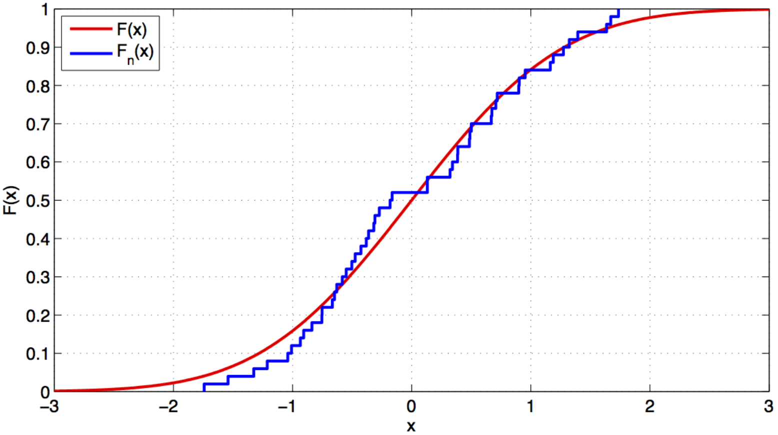


Сгенерировать N «псевдовыборок» объёма n и оценить выборочное распределение $\hat{\theta}_n$ «псевдоэмпирическим».

Бутстреп: принцип работы

Извлечение выборок из генеральной совокупности — сэмплирование из неизвестного распределения $F_X(x)$.

Лучшая оценка $F_X(x)$, которая у нас есть — $F_{X^n}(x)$:



Порождение данных: сэмплирование

Базовый подход Пусть существует обратимая функция T из $x \in \mathcal{U}(0, 1)$ в некоторое распределение z. Тогда

$$F_z(t) = p(z \leq t) = p(T(t') \leq t) = p(t' \leq T^{-1}(t)) = T^{-1}(t).$$

Отсюда $F_z^{-1} = T$.

Пример

$$z = \lambda \text{exp}(-\lambda t).$$

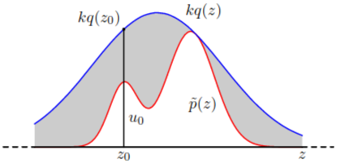
$$F_z(t) = 1 - \text{exp}(-\lambda t).$$

$$F_z^{-1}(t') = -1\frac{1}{\lambda}\text{log}(1 - t').$$

Сэмплирование с отклонением

- Задана плотность $p(z)$ (может быть задана с точностью до нормировочной константы)
- Введем распределение q
- Подберем множитель k таким образом, чтобы $kq(z) \geq p(z)$ для всех z
- В цикле
  - Просэмплируем $z_0 \sim q$
  - Просэмплируем $u \sim \mathcal{U}(0, kq(z_0))$
  - Если $u \leq p(z_0)$ — считать его сэмплом из $p(z)$

Идея метода: сэмплы u равномерно распределены в регионе, ограниченном кривой $p(z)$.



Bishop, 2006

Dataset shift

Dataset shift — явление, при котором распределение данных $p(\boldsymbol{X}, \boldsymbol{y})$ различается на этапе обучения и этапе контроля.

- Covariate shift — различие в $p(\boldsymbol{X})$
- Prior probability shift — различие в $p(\boldsymbol{y})$
- Concept shift — различие в $p(\boldsymbol{y}|\boldsymbol{X})$

Dataset shift

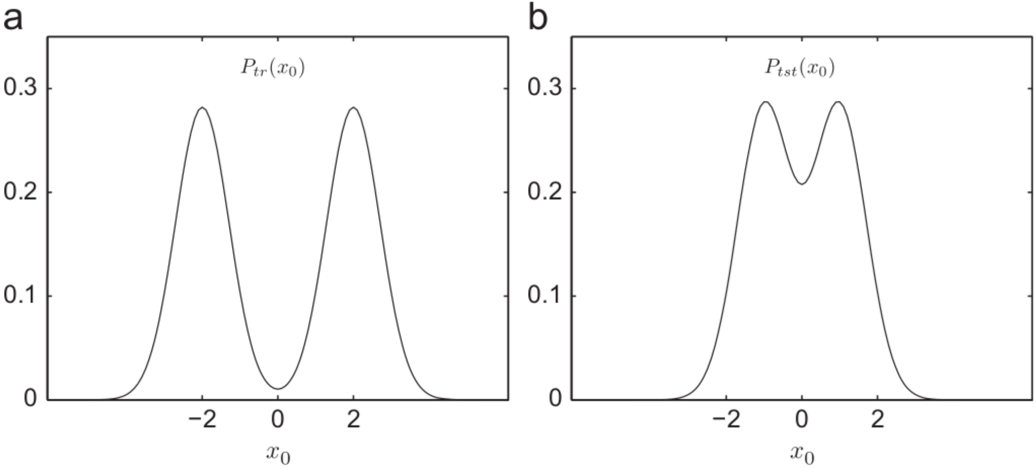
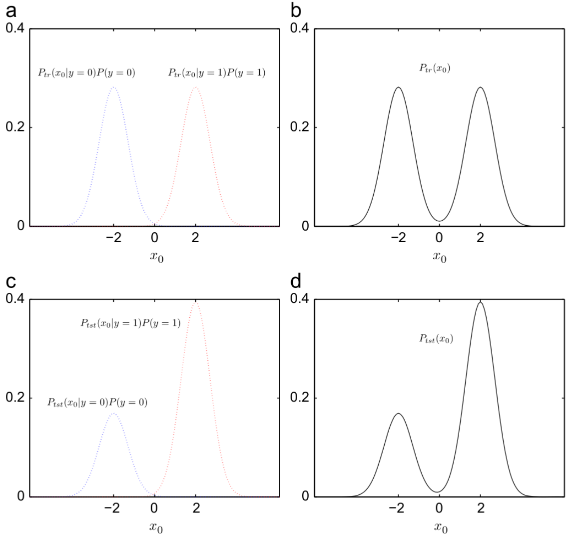


Fig. 1. Covariate shift: $P_{tst}(y|x_0) = P_{tr}(y|x_0)$ and $P_{tr}(x_0) \neq P_{tst}(x_0)$. (a) Training data and (b) test data.

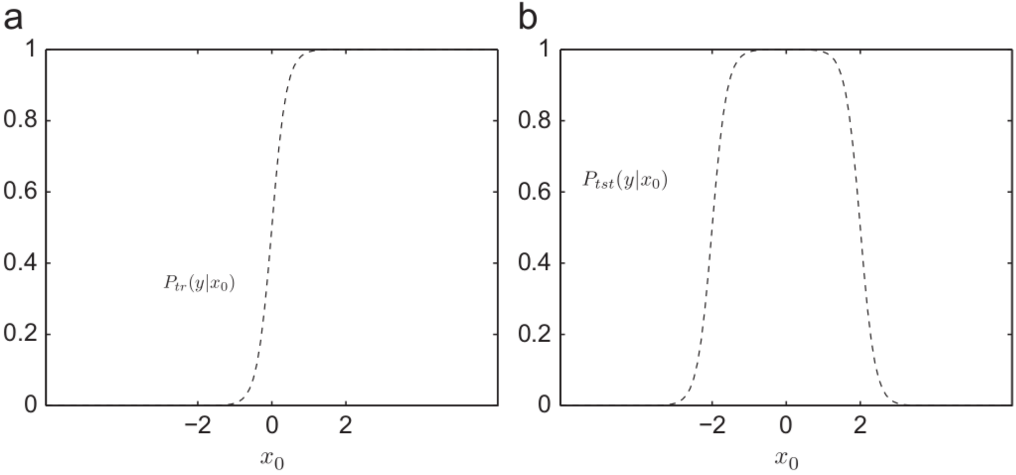
Moreno-Torres et al., 2012

Dataset shift



Moreno-Torres et al., 2012

Dataset shift



Moreno-Torres et al., 2012

Evidence vs Кросс-валидация

Оценка Evidece:

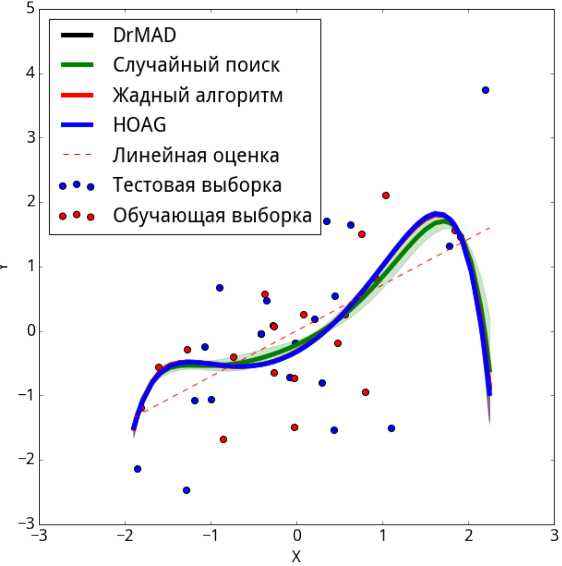
$$\log p(\boldsymbol{X}|\boldsymbol{f}) = \log p(\boldsymbol{x}_1|\boldsymbol{f}) + \log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{f}) + \dots + \log p(\boldsymbol{x}_n|\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-1}, \boldsymbol{f}).$$

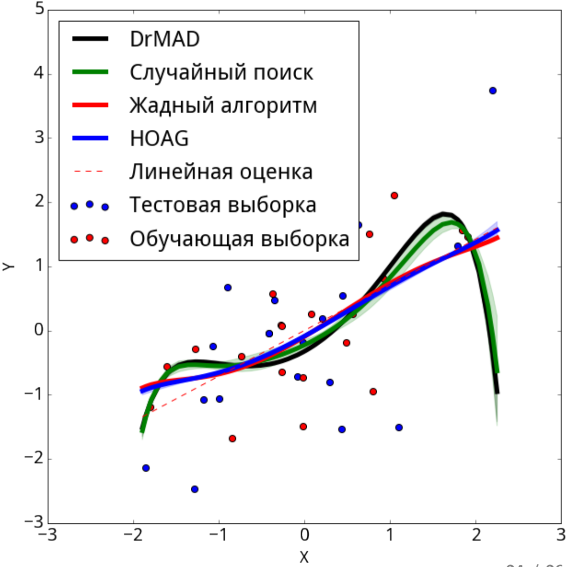
Оценка leave-one-out:

$$\mathrm{LOU} = \mathrm{E}\log p(\boldsymbol{x}_n|\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-1}, \boldsymbol{f}).$$

Кросс-валидация использует среднее значение последнего члена $p(\boldsymbol{x}_n|\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-1}, \boldsymbol{f})$ для оценки сложности.
Evidence учитывает **полную** сложность описания заданной выборки, определяющую предсказательную способность модели с самого начала.

Evidence vs Кросс-валидация: пример





Литература и прочие ресурсы

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – T. 128. – №. 9.
- Paganini M., de Oliveira L., Nachman B. Accelerating science with generative adversarial networks: an application to 3D particle showers in multilayer calorimeters //Physical review letters. – 2018. – T. 120. – №. 4. – C. 042003.
- Antoran J., Miguel A. Disentangling and learning robust representations with natural clustering //2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). – IEEE, 2019. – C. 694-699.
- Adams R. P., Wallach H., Ghahramani Z. Learning the structure of deep sparse graphical models //Proceedings of the thirteenth international conference on artificial intelligence and statistics. – JMLR Workshop and Conference Proceedings, 2010. – C. 1-8.
- Alain G., Bengio Y. What regularized auto-encoders learn from the data-generating distribution //The Journal of Machine Learning Research. – 2014. – T. 15. – №. 1. – C. 3563-3593.
- Theis L., Oord A., Bethge M. A note on the evaluation of generative models //arXiv preprint arXiv:1511.01844. – 2015.
- Kingma D. P., Welling M. Auto-Encoding Variational Bayes //stat. – 2014. – T. 1050. – C. 10.
- Efron B., Tibshirani R. *An Introduction to the Bootstrap*, 1993.
- Moreno-Torres J. G. et al. A unifying view on dataset shift in classification //Pattern recognition. – 2012. – T. 45. – №. 1. – C. 521-530.
- Bakhteev O. Y., Strijov V. V. Comprehensive analysis of gradient-based hyperparameter optimization algorithms //Annals of Operations Research. – 2020. – T. 289. – №. 1. – C. 51-65.

Литература и прочие ресурсы

- Aditya Grover et al., Deep Generative Models tutorial, 2018: goo.gl/H1prjP
- Fei-Fei Li et al., Generative Models tutorial, 2017, http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
- Shakir Mohamed et al., UAI 2017 Tutorial, 2017, https://www.youtube.com/watch?v=JrO5fSskISY
- Andrej Karpathy: The Unreasonable Effectiveness of Recurrent Neural Networks, 2015: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- Maxim Panov: Uncertainty, Out-of-distribution detection for NNs: https://www.youtube.com/watch?v=N-p_qSLzoAI
- https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/
- Генератор котиков: https://github.com/aleju/cat-generator