# Метод Монте-Карло

Московский Физико-Технический Институт

2022

# Выбор модели: связанный байесовский вывод

*Первый уровень:* выбираем оптимальные параметры:

$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

*Второй уровень:* выбираем модель, доставляющую максимум обоснованности модели.
Обоснованность модели (“Evidence”):

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$

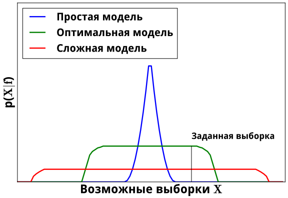
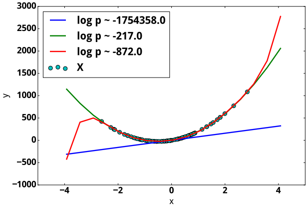



Схема выбора модели



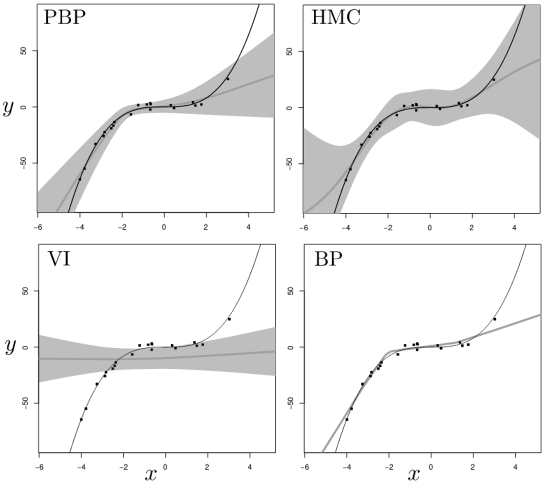

Пример: полиномы

# Оценки интеграла

$$\mathsf{E}f = \int_{\boldsymbol{w}} f(\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}.$$

- Аппроксимация Лапласа
  - Негибкий, аппроксимирует нормальным распределением
  - Есть проблемы с большими размерностями пространства
- Вариационный вывод
  - Гибкий, позволяет аппроксимировать неизвестное распределением широким классом распределений
  - Нижняя оценка интеграла, при некорректном выборе вариационного распределения может давать сильно заниженную оценку
- MC
  - Гибкий
  - Точный
  - Медленный

# VI vs MC



# Наивный метод

$$I = \mathsf{E}f = \int_{w} f(\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}.$$

Аппроксимируем:

$$\hat{I} = \frac{1}{N}\sum_{\boldsymbol{w}\sim p(\boldsymbol{w})} f(\boldsymbol{w}).$$

# Свойства

Оценка интеграла:

- Сильно состоятельна: $\hat{I} \to^{\text{п.н.}} I$
- Несмещена: $\mathrm{E}\hat{I} = I$
- Асимптотически нормальна;
- $\mathrm{D}\hat{I} = O(\frac{1}{N})$.
- **Проблема:** нужно уметь сэмплировать из $p$.

# Наивный метод

Пусть существует обратимая функция $T$ из $\mathcal{U}(0,1)$ в некоторое распределение $w$. Тогда

$$F_w(t) = p(w \leq t) = p(T(w') \leq t) = p(w' \leq T{-}1(t)) = T{-}1(w).$$

Отсюда $F_w^{-1} = T$.

**Пример**

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

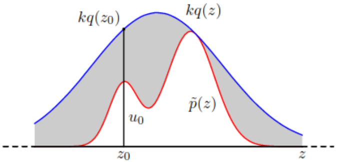
$$F_w(t) = 1 - \text{exp}(-\lambda t).$$

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

# Сэмплирование с отклонением

- Задана плотность $p(w)$ (может быть задана с точностью до нормировочной константы)
- Введем распределение $q$
- Подберем множитель $k$ таким образом, чтобы $kq(w) \geq p(z)$ для всех $z$
- В цикле
  - Просэмплируем $w_0 \sim q$
  - Просэмплируем $u \sim \mathcal{U}(0, kq(w_0))$
  - Если $u \leq p(w_0)$ — считать его сэмплом из $p(w)$

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




Bishop, 2006

# Сэмлирование по значимости

Пусть мы не можем сэмплировать из $p(w)$, но можем оценивать правдоподобие в каждой точке, и хотим получить интерал

$$\mathsf{E} f = \int f(w) p(w) dz.$$

Тогда введем распределение $q$:

$$\mathsf{E} f = \int f(w) p(w) dw = \int f(w) \frac{p(w)}{q(w)} dz \approx \frac{1}{L} \sum_{l=1}^{L} \frac{p(w^l)}{q(w^l)} f(w^l).$$

# Nested sampling

Приближение интеграла вида:

$$\int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

- В базовом виде — приближение через априорное распределение $p(\mathbf{w})$
- Можно ли сэмплировать из апостериорное распределение?

**Идея:**

- Введем $I_t = \int_{\mathbf{w}:p(\mathfrak{D}|\mathbf{w})>t} p(\mathbf{w})d\mathbf{w}$
- Вычисление интеграла сводится к интегрированию по контуру: $I = \int_0^\infty I_t dt$.

# MCMC

**Основная идея:** Сэмплируем аналогично сэмплированию с отклонениями, но q — марковское распределение, обусловленное на предлыдущий успешный шаг
Хотим, чтобы предельное (стационарное) распределение соответствовало нашему распределению $p(w)$.
Достаточное условие
$$p(w)T(w|w') = p(w)T(w'|w).$$

# Алгоритм Метрополиса — Гастингса

- Сэмплируем новое значение $w' \sim q(w|w^t)$.
- Принимаем его с вероятностью $A(w'|w^t) = \min\left(1, \frac{p(w')q(w^t|w')}{p(w^t)q(w'|w^t)}\right)$.
- Если приняли: $w^{t+1} = w'$,
- иначе: $w^{t+1} = w^t$.

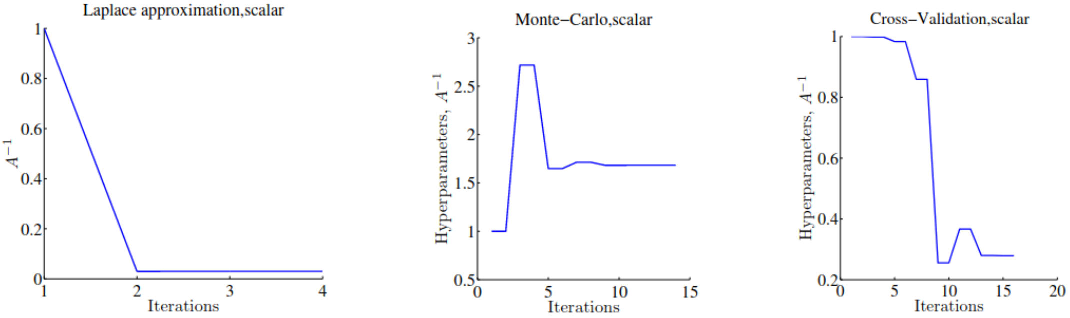
Условие предельного распределения выполняется:

$$p(w)T(w|w') = p(w)T(w'|w) = p(w')T(w'|w^t) = p(w')q(w'|w^t)A(w'|w^t) =$$

$$= p(w^t)q(w^t|w')A(w^t|w').$$

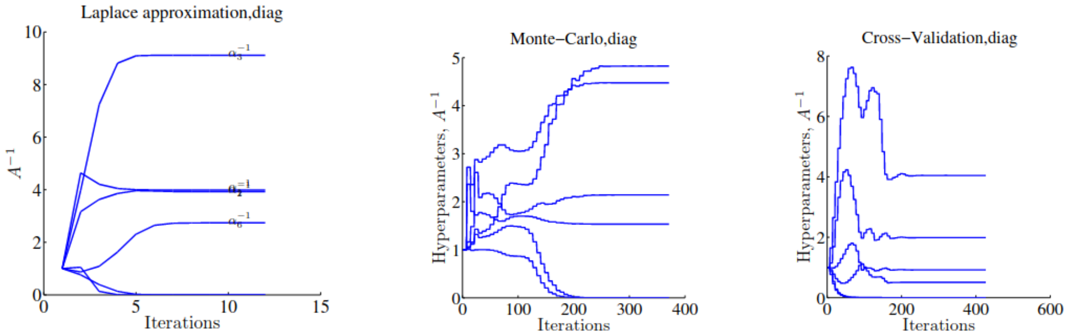
- Сэмплы скоррелированы. Если требуется декоррелировать сэмплы, можно брать каждый $k$-й сэмпл.
- Работает в пространствах выоской размерности значительно лучше, чем сэмплирование с отклонением.
- Выбор адекватного рапределения $q$ — главная оптимизационная задача алгоритма.

# Пример: линейная модель



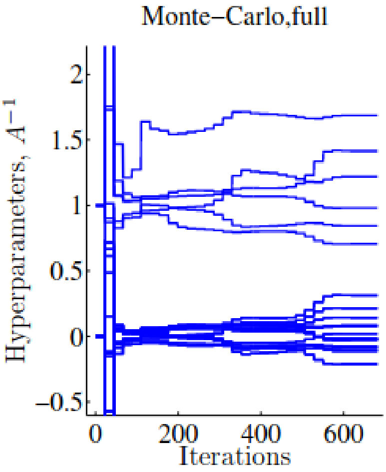

(a) Laplace approximation (b) Monte-Carlo (c) Cross validation

# Пример: линейная модель

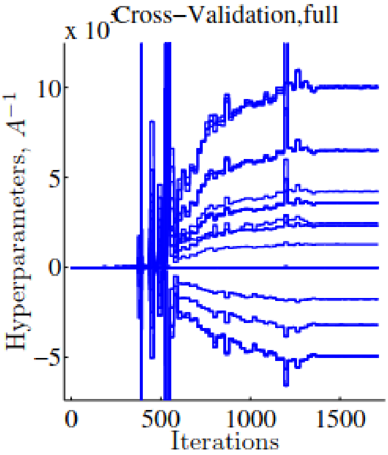



(a) Laplace approximation (b) Monte-Carlo (c) Cross validation

# Пример: линейная модель




(a) Monte-Carlo




(b) Cross validation

# Стохастическая динамика Ланжевена

Модификация стохастического градиентного спуска:

$$T = \mathbf{w} - \beta \nabla L + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \frac{\alpha}{2})$$

где шаг оптимизации $\alpha$ изменяется с количеством итераций:

$$\sum_{\tau=1}^{\infty} \beta_\tau = \infty, \quad \sum_{\tau=1}^{\infty} \beta_\tau^2 < \infty.$$

**Утверждение [Welling, 2011].** Распределине $q^\tau(\mathbf{w})$ сходится к апостериорному распределению $p(\mathbf{w}|\mathbf{X}, \mathbf{f})$.

Изменение энтропии с учетом добавленного шума:

$$\hat{\mathsf{S}}(q^\tau(\mathbf{w})) \geq \frac{1}{2}|\mathbf{w}|\log(\exp(\frac{2\mathsf{S}(q^\tau(\mathbf{w}))}{|\mathbf{w}|}) + \exp(\frac{2\mathsf{S}(\epsilon)}{|\mathbf{w}|})).$$

# Precondition-матрица для динамики Ланжевена

SGLD:

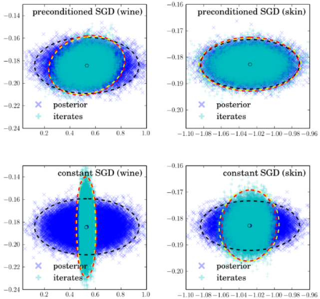
$$T = \boldsymbol{w} - \beta \nabla L + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \frac{\alpha}{2})$$

pSGLD:

$$T = \boldsymbol{w} - \beta \boldsymbol{M} \nabla L + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \frac{\alpha}{2}\boldsymbol{M}),$$

матрица $\boldsymbol{M}$ выбирается так, чтобы сделать шаг градиентного спуска равномерным по всем направлениям (с учетом дисперсии градиента).

*Пример для SGD, для SGLD — аналогично.*



# Сэмплирование по Гиббсу

Пусть задана графическая модель для набора переменных $w_1, \ldots, w_n$.
Тогда в цикле по всем переменным:

$$\hat{w}_i \sim p(w_i | w_1, w_{i-1}, w_{i+1}, w_n), w_i := \hat{w}_i$$

Сэмплирование Гиббса — частный случай MH-алгоритма.

# Contrastive Divergence: идея

Energy-based model:

$$p(x|w) = \frac{\exp\left(-E_w(x)\right)}{Z(w)}, \quad Z = \int_x \exp\left(-E_w(x)\right),$$

$$\frac{\partial \log p(x|w)}{\partial w} = \mathsf{E}_{x'} \frac{\partial E(x')}{\partial w} - \frac{\partial E(x)}{\partial w}$$

Алгоритм для RBM:

- Берем $x$ из выборки
- $h_0 \sim p(h_0|x)$
- $x_1 \sim p(x|h_0)$
- ...
- Получаем $x_k$
- $\frac{\partial \log p(x|w)}{\partial w} = \frac{\partial E(x_k)}{\partial w} - \frac{\partial E(x)}{\partial w}$

# Автокодировщик: порождающая модель?

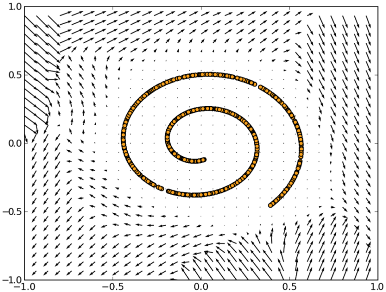
(Alain, Bengio 2012): рассмотрим модель автокодировщика с регуляризацией:

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

где $\sigma$ — уровень шума, подаваемого на вход модели кодирования. Тогда

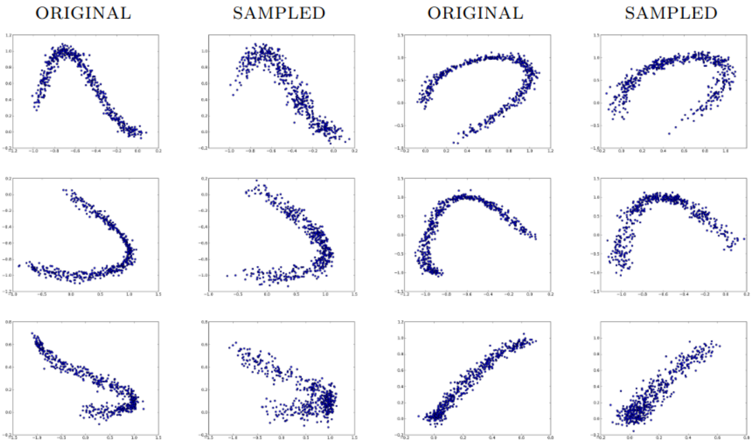
$$\frac{\partial \log p(x)}{\partial x} = \frac{||\boldsymbol{f}(\boldsymbol{x}, \sigma) - \boldsymbol{x}||^2}{\sigma^2} + o(1) \text{ при } \sigma \to 0.$$

Векторное поле, индуцированное ошибкой реконструкции автокодировщика



# Автокодировщик для оценки сэмплирования

$$A = \frac{p(x^*)}{p(x)} = \exp\left(E(x) - E(x^*)\right) \approx \frac{\partial E(x)}{\partial x}^{\mathsf{T}}(x^* - x) + o(||x - x^*||).$$



# Оптимизация распределения q

Распределение $q$ можно задавать не как статическое распределение, а как сложное нейросетевое распределение.

- **Основное требование:** определенность распределений $p(x|x'), p(x'|x) \rightarrow$ распределение должно быть инвертируемо.
- Нейросеть вида $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{w})$ является потоком и инвертируемо.

**Варианты оптимизации:**

- Энтропия * Acceptance rate (Li et al., 2020)
- Состязательная оптимизация между эмпирическим распределением и распределением $q$ (Song et al., 2017).

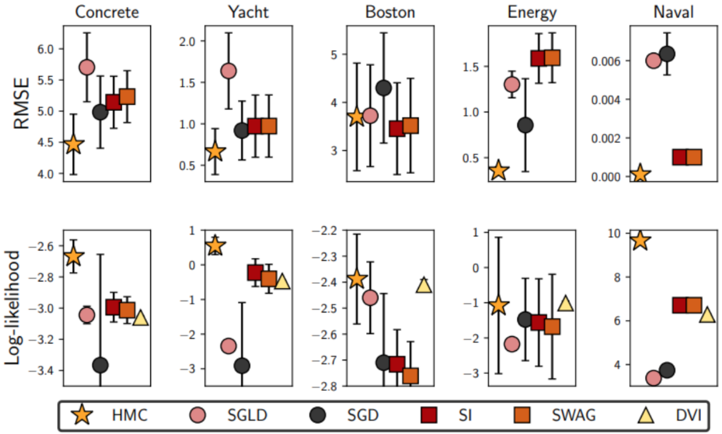
# What Are Bayesian Neural Network Posteriors Really Like?

Izmailov et al., 2021:

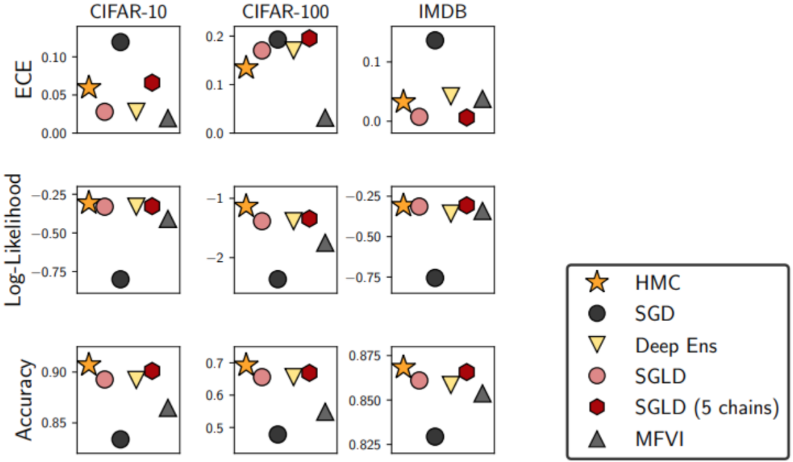
- Методами HMC решается задача нахождения апостериорного распределения для глубоких моделей на стандартных датасетах.
- Вычисления: 512 TPU

# What Are Bayesian Neural Network Posteriors Really Like?

## BNN evaluation: UCI

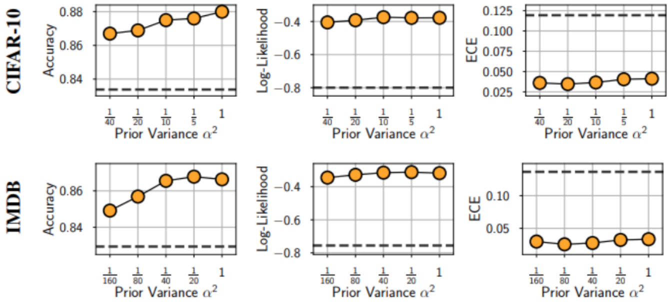


# BNN evaluation: CIFAR and IMDB



# What Are Bayesian Neural Network Posteriors Really Like?

## Effect of priors




HMC BNNs are fairly robust to Gaussian prior variance.

# Литература


- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – Т. 4. – №. 4. – С. 738.
- Hernández-Lobato J. M., Adams R. Probabilistic backpropagation for scalable learning of bayesian neural networks //International conference on machine learning. – PMLR, 2015. – С. 1861-1869.
- Kuznetsov M., Tokmakova A., Strijov V. Analytic and stochastic methods of structure parameter estimation //Informatica. – 2016. – Т. 27. – №. 3. – С. 607-624.
- Welling M., Teh Y. W. Bayesian learning via stochastic gradient Langevin dynamics //Proceedings of the 28th international conference on machine learning (ICML-11). – 2011. – С. 681-688.
- Mandt S., Hoffman M., Blei D. A variational analysis of stochastic gradient algorithms //International conference on machine learning. – PMLR, 2016. – С. 354-363.
- Alain G., Bengio Y. What regularized auto-encoders learn from the data-generating distribution //The Journal of Machine Learning Research. – 2014. – Т. 15. – №. 1. – С. 3563-3593.
- Li Z., Chen Y., Sommer F. T. A neural network mcmc sampler that maximizes proposal entropy //arXiv preprint arXiv:2010.03587. – 2020.
- Song J., Zhao S., Ermon S. A-nice-mc: Adversarial training for mcmc //Advances in Neural Information Processing Systems. – 2017. – Т. 30.
- Izmailov P. et al. What are Bayesian neural network posteriors really like? //International Conference on Machine Learning. – PMLR, 2021. – С. 4629-4640.