Bayesian multimodeling: Bayesian inference and basic sampling methods

MIPT

2024

Coin problem

A person flips a coin N times. What's the probability of getting tails on a coin?

Coin problem

A person flips a coin 3 times. All 3 times it comes up tails. What's the probability of getting tails on a coin?

Naive approach

$$\mathbf{X} = [1, 1, 1];$$

$$x \sim \mathrm{Bin}(w);$$

$$\hat{w} = \arg\max_{p} L(\mathbf{X}, w);$$

$$\rightarrow \hat{w} = 1.$$

Challenge: three events are not enough to estimate the distribution of heads and tails.

Frequentist and Bayesian statistics

Frequentist statistics

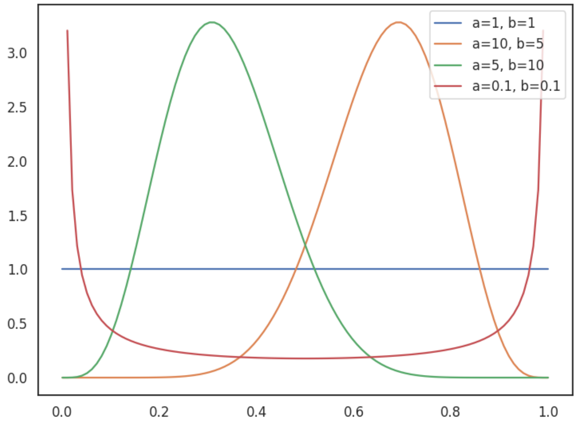
- Model parameter is a constant that is required to be estimated
- Probability is estimated purely from event frequency

Bayesian statistics

- Model parameter is a random value
  - We cannot "estimate" random value
  - But we can estimate its distribution parameters
- Probability is estimated w.r.t. our prior beliefs about data and parameter distribution
  - The more data we get the closer our estimation to MLE
  - In general our estimation is strongly relies on the prior

Beta-distribution: recap

- corresponds to the *prior* beliefs about Bernoulli distribution
- interpretation of parameters a, b: "effective number of events $w = 1, w = 0$"
- With $n \to \infty$ converges toδ-distribtuion with PDF concentration at MLE for Bernoulli.



Bayesian approach

Use beta-distribution as a *prior* distribution for our parameter w . From general considerations, the distribution should be symmetrical (unless we have more information):

$$p(w) \sim B(\alpha, \beta).$$

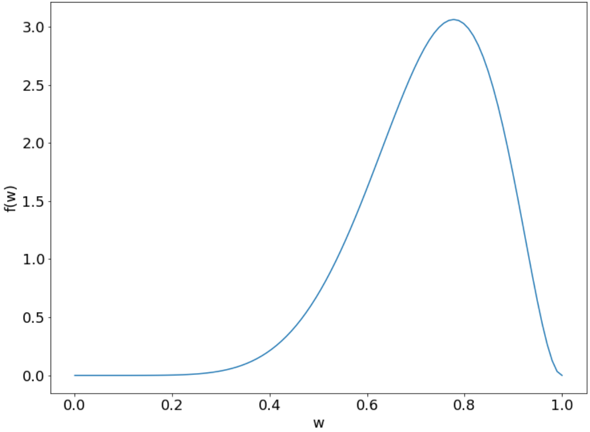
Find the *posterior* distribution of w using Bayes formula:

$$p(w|\mathbf{X}) = \frac{p(\mathbf{X}|w)p(w)}{p(\mathbf{X})} \propto p(\mathbf{X}|w)p(w);$$

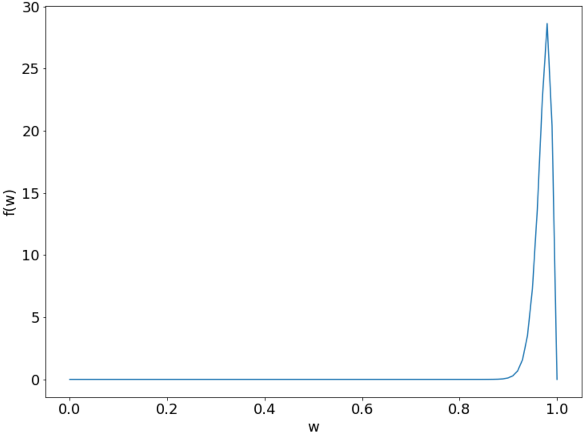
$$\log p(w|\mathbf{X}) = \log p(\mathbf{X}|w) + \log p(w) + \text{Const}.$$

Conclusion: roughly prior is a regularizer.

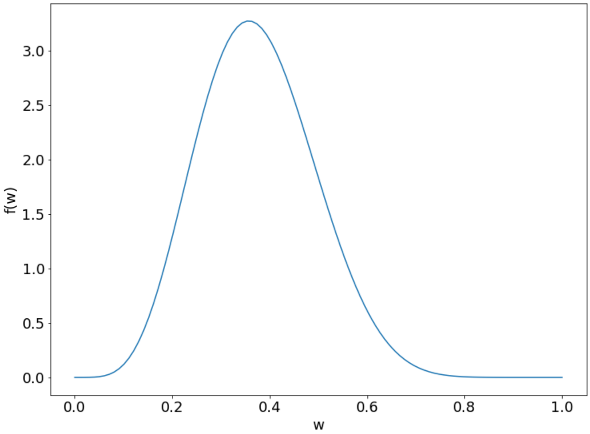
Posterior, $\alpha = 3, \beta = 3$



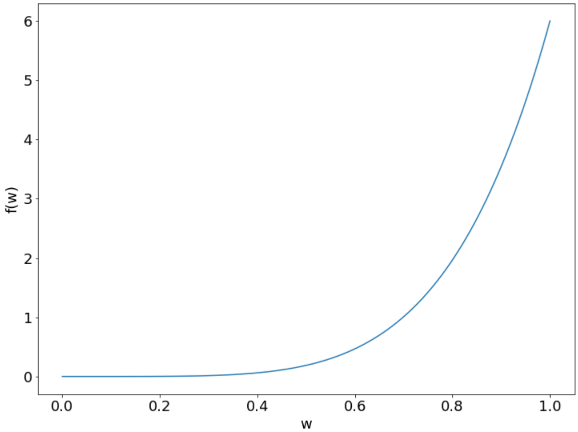
Posterior, 100 elements



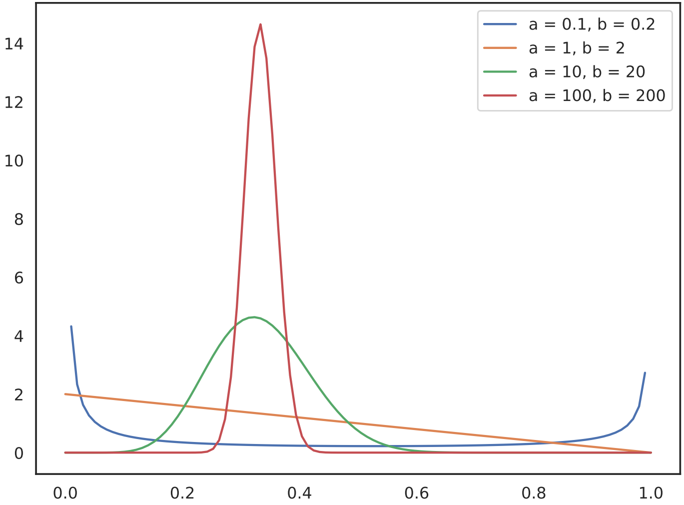
Posterior, $\alpha = 1, \beta = 10$



Posterior, $\alpha = 1, \beta = 1$



Beta-distribution for the sample α-β-ratio



Bayesian inference: first level

Given:

- likelihood $p(\mathbf{X}|\mathbf{w})$ of the dataset $\mathbf{X}$ w.r.t. parameters $\mathbf{w}$;
- prior distribution $p(\mathbf{w}|\mathbf{h})$
- prior parameters $\mathbf{h}$ (for the coin problem: $\mathbf{h} = [\alpha, \beta]$;)

Then the posterior for $\mathbf{w}$ w.r.t. $\mathbf{X}$:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{h}) = \frac{p(\mathbf{X}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathbf{X}|\mathbf{h})} \propto p(\mathbf{X}|\mathbf{w})p(\mathbf{w}|\mathbf{h}).$$

Find a point estimate as a maximum posterior probability (MAP):

$$\hat{\mathbf{w}} = \arg\max p(\mathbf{X}|\mathbf{w})p(\mathbf{w}|\mathbf{h}).$$

MAP-estimation is similar to MLE, if

- the dataset is large;
- prior is uniform in an infinitely large region (improper prior)

Why we used Beta-distribution?

$$p(w|\mathbf{X},\alpha,\beta) \propto p(\mathbf{X}|w)p(w|\alpha,\beta) \propto$$
$$\propto w^{\sum x}(1-w)^{m-\sum x} \times w^{\alpha-1}(1-w)^{\beta-1} =$$
$$= w^{\alpha-1+\sum x}(1-w)^{m+\beta-\sum x-1} \sim B(\alpha+\sum x, \beta+m-\sum x).$$

The distribution family is conjugate prior to the likelihood distribution, if the posterior belongs to the same family.

Prior families

- **Discrete (labels, discrete parameters)**
  - Bernoulli
  - Categorial distributions

  Hyperparameters (parameters of the prior parameters):
  - $w \sim \text{Bin}(w)$: $w \sim B(\alpha, \beta)$: conjugate
  - $w \sim \text{Cat}(w)$: $w \sim \text{Dir}(\boldsymbol{\alpha})$: conjugate
- **Real-valued distributions**
  - $\mathcal{N}$
  - Laplace
  - $\mathcal{C}$

  Hyperparameters:
  - Precision, $w \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2), \sigma^{-1} \in \Gamma$: conjugate for Gaussian distribution
  - Expectation, $\boldsymbol{\mu} \in \mathcal{N}$: conjugate for Gaussian distribution

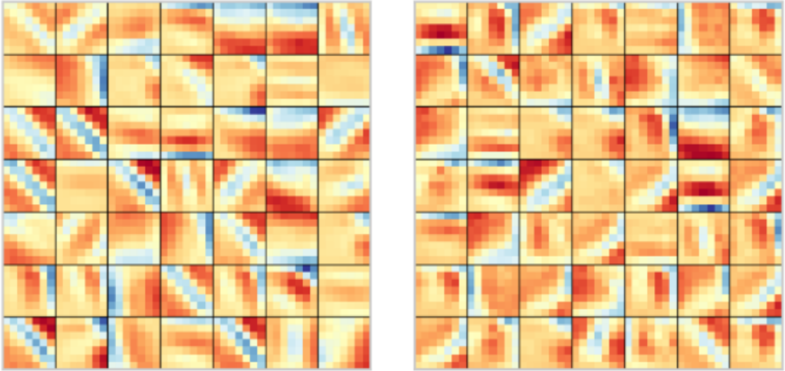
Informative prior vs Uninformative prior

- Informative prior: corresponds to some expert knowledge
  - Example: air temperature in some region: Gaussian variable with known mean and variance estimated from previous observations.
  - Mistake in informative prior estimation leads to poor models.
- Uninformative prior: corresponds to some basic knowledge
  - Example: air temperature in some region: uniform improper prior.
- Weakly-informative prior: somewhere in between
  - Example: air temperature in some region: uniform distribution in [-50, 50] degrees.

To discuss:

- $\mathbf{w} \sim \mathcal{N}(0, \mathbf{A}^{-1})$ — what type of the prior distribution?
- What if our prior and posterior are very close

The deep weight prior: Atanov et al., 2019



(b) Learned filters (c) Samples from DWP

The distribution can be modeled implicitlye by complex models and can generate rather informative samples.

Jeffreys prior

Uninformative prior:

$$p(\mathbf{w}) \propto \sqrt{\det I(\mathbf{w})} = \sqrt{\det\left(-\frac{\partial^2}{\partial w^2}\log L(w)\right)}.$$

- Invariant under the variable change:

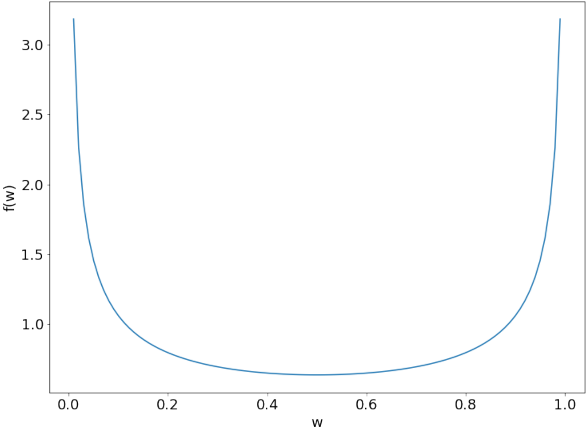
$$p(g(\mathbf{w})) = p(\mathbf{w})\left|\frac{dg}{d\mathbf{w}}\right| \rightarrow$$

$$p(g(\mathbf{w})) \propto \sqrt{\det I(g(\mathbf{w}))}.$$

- Interpretation: a value inverse to the amount of information obtained by our model from the dataset
- Examples:
  - $y \in \text{Bin}(w)$: $\quad p(w) \propto \frac{1}{\sqrt{p(1-p)}}$ — Beta-distribution (0.5, 0.5).
  - $w \in \mathcal{N}(\mu, \sigma)$: $\quad p(\mu) \propto \text{Const}$.
  - $w \in \mathcal{N}(\mu, \sigma)$: $\quad p(\sigma) \propto \frac{1}{|\sigma|}$.

See the talk of Galina Boeva, 2023, about learning Jeffreys prior

Uninformative prior $\neq$ flat prior!



See also the talk of Sergey Skorik, 2022

Model selection problem: Bayesian coherent inference

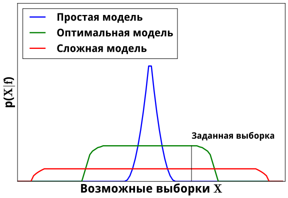
First level: find optimal parameters:

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

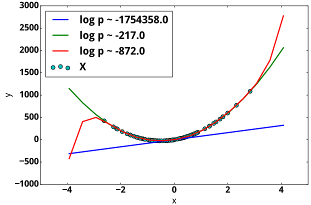
Second level: find model, that give optimal Evidence.

"Evidence":

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



Model selection scheme



Example: polynomial regression

Example: linear regression

Given m objects with n features

$\mathbf{f}(\mathbf{X}, \mathbf{w}) = \mathbf{X}\mathbf{w}$; $\mathbf{y} \sim \mathcal{N}(\mathbf{f}(\mathbf{X}, \mathbf{w}), \beta^{-1}), \mathbf{w} \sim \mathcal{N}(0, \mathbf{A}^{-1})$.

Write down the integral:

$$p(\mathfrak{D}|\mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \beta) = \frac{\sqrt{\beta \cdot |\mathbf{A}|}}{\sqrt{(2\pi)^{m+n}}} \int_{\mathbf{w}} \exp\left(-0.5\beta(\mathbf{y} - \mathbf{f})^{\mathsf{T}}(\mathbf{y} - \mathbf{f})\right) \exp\left(-0.5\mathbf{w}^{\mathsf{T}}\mathbf{A}\mathbf{w}\right) d\mathbf{w} =$$

$$= \frac{\sqrt{\beta \cdot |\mathbf{A}|}}{\sqrt{(2\pi)^{m+n}}} \int_{\mathbf{w}} \exp(-S(\mathbf{w})) d\mathbf{w}$$

Its value is tractable for the linear regression case:

$$\int_{\mathbf{w}} \exp(-S(\mathbf{w})) d\mathbf{w} = (2\pi)^{\frac{n}{2}} \exp(-S(\hat{\mathbf{w}})) |\mathbf{H}^{-1}|^{0.5},$$

where

$$\mathbf{H} = \mathbf{A} + \beta \mathbf{X}^{\mathsf{T}}\mathbf{X},$$

$$\hat{\mathbf{w}} = \beta \mathbf{H}^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$$

Conclusion: we can find the value of the Evidence for the linear models.

Example: Laplace approximation

Given m objects with n features
$\mathbf{y} \sim \mathcal{N}(\mathbf{f}(\mathbf{X}, \mathbf{w}), \beta^{-1}), \mathbf{w} \sim \mathcal{N}(0, \mathbf{A}^{-1})$.
Write down the integral:

$$p(\mathfrak{D}|\mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \beta) = \frac{\sqrt{\beta \cdot |\mathbf{A}|}}{\sqrt{(2\pi)^{m+n}}} \int_{\mathbf{w}} \exp(-S(\mathbf{w})) d\mathbf{w}.$$

Use Taylor serioes for S:

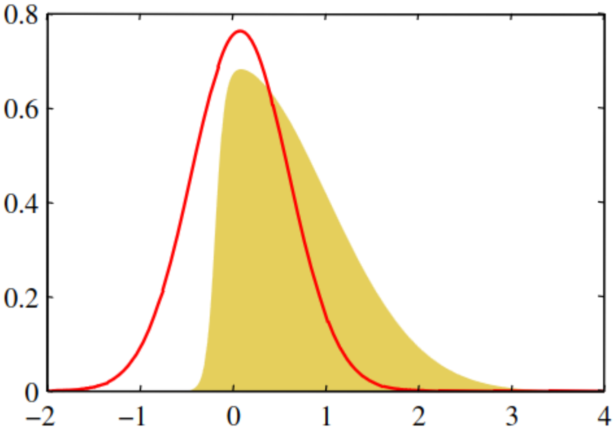
$$S(\mathbf{w}) \approx S(\hat{\mathbf{w}}) + \frac{1}{2} \Delta\mathbf{w}^{\mathsf{T}} \mathbf{H} \Delta\mathbf{w}$$

Then:

$$\frac{\sqrt{\beta \cdot |\mathbf{A}|}}{\sqrt{(2\pi)^{m+n}}} S(\hat{\mathbf{w}}) \int_{\mathbf{w}} \exp(-\frac{1}{2} \Delta\mathbf{w}^{\mathsf{T}} \mathbf{H} \Delta\mathbf{w}) d\mathbf{w}$$

The expression corresponds to the PDF for unnormalized Gaussian distribution.
Conclusion: we can use Laplace approximation for the non-linear models.

Laplace approximation: example



Bishop, 2006

Laplace approximation: drawbacks

- Only Gaussian distribution is available
  - No multimodality
- Hessian inversion: terribely slow
  - we can use diagonal matrix, but with worse approximation

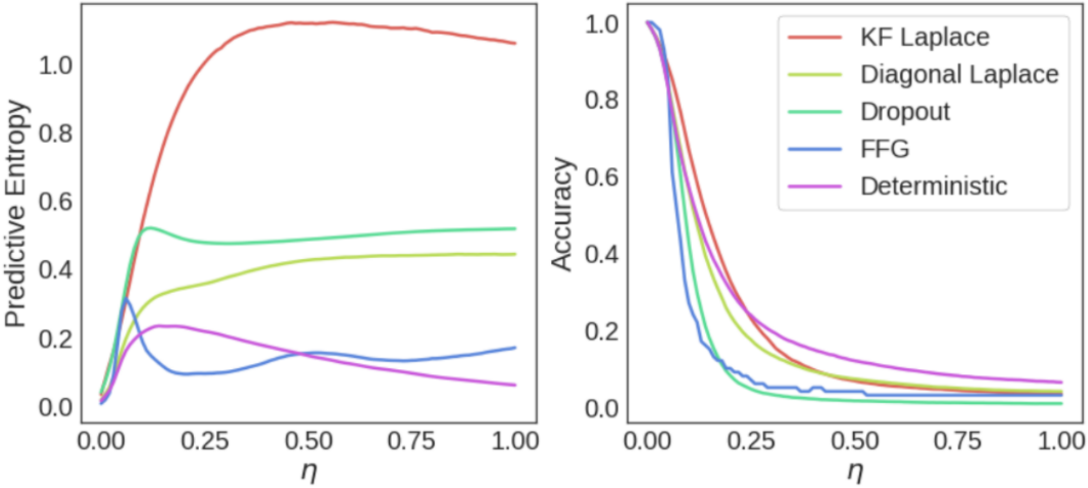
A scalable Laplace approximation for neural networks: Ritter et al., 2018

- Decompose the neural network parameters by the layers, make an assumption that parameters from different layers are not correlated
- $\mathbf{H}_l = (\mathbf{f}_l(\mathbf{h}_l)\mathbf{f}_l(\mathbf{h}_l)^{\mathsf{T}}) \circ \mathbf{H}(\mathbf{h}_l)$ with Kronecker product.
- Reduce the complexity because of blockwise posterior structure
- Inverse of Kronecker product is equal to the Kronecker product of the inverses

---

For details see the talk of Alekandr Kolesov, 2021

Approximation mode matters



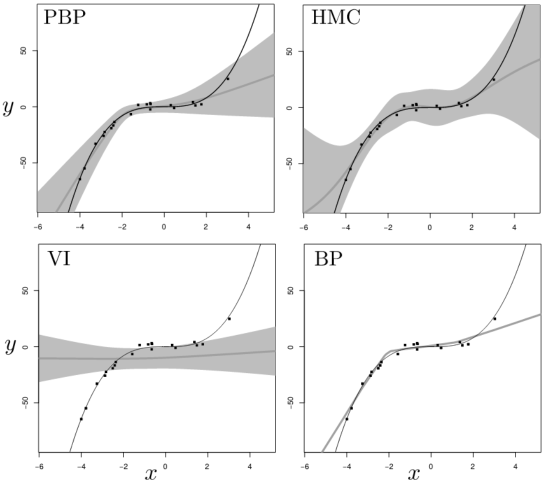
Evidence estimation

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

- Laplace approximation
  - Fixed form of approximation distribution
  - Poorly scales
- Variational inference
  - Well scales
  - Can use different forms of approximation distributions
  - Lower bound of evidence => biased
- MC
  - Can use different forms of approximation distributions
  - Approximates well
  - Slow

See the talk of Alekandr Kolesov about relation between VI and MC, 2021

VI vs MC



Naive method

$$I = \mathsf{E}f = \int_w f(\mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

Approximate:

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

Why this does not work?

Properties

Integral estimation:

- strongly consistent : $\hat{I} \to^{\text{a.s.}} I$
- Unbiased: $\mathrm{E}\hat{I} = I$
- Assymptotically normal;
- $\mathrm{D}\hat{I} = O(\frac{1}{N})$.
- **Challenge:** we need to sample from p.

Inverse transform sampling

Let T be an invertible function from $u \sim \mathcal{U}(0,1)$ to some random variable distribution $p(w)$. Then

$$F_w(t) = p(w \leq t) = p(T(u) \leq t) = p(u \leq T^{-1}(t)) = T^{-1}(t).$$

Therefore we can generate w using T^{-1}.

Example

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

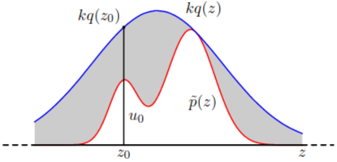
$$F_w(t) = 1 - \text{exp}(-\lambda t).$$

$$F_w^{-1}(u) = -1\frac{1}{\lambda}\text{log}(1-u).$$

Rejection sampling

- Given $p(w)$ (up to normalizing constant)
- Set distribution q
- Set value k so that $kq(w) \geq p(z)$ for all z
- In a loop:
  - Sample $w_0 \sim q$
  - Sample $u \sim \mathcal{U}(0, kq(w_0))$
  - If $u \leq p(w_0)$, use it as a sample from $p(w)$

Core idea: samples u are uniform in a region limited by $p(w)$.



Bishop, 2006

Importance sampling

Consider the case when we cannot sample from $p(w)$, but we can estimate likelihood and want to estimate the integral

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

Let q be an auxilary distribution:

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

MCMC

Basic idea: Sample similar to rejection sampling, but q is a Markov distribution with conditioning on the previous step.
We want the stationary (limiting) distribution to be equal to our $p(w)$.
Sufficient condition

$$p(w')T(w|w') = p(w)T(w'|w).$$

Metropolis-Hastings algorithm

- Sample new $w' \sim q(w|w^t)$.
- Accept with probability $A(w'|w^t) = \min\left(1, \frac{p(w')q(w^t|w')}{p(w^t)q(w'|w^t)}\right)$.
- If accepted: $w^{t+1} = w'$,
- Otherwise: $w^{t+1} = w^t$.

Sufficient condition is satisfied:

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

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

- Samples are correlated. We can decorrelate sample using each k sample.
- Works better in high-dimmensional settings than rejection sampling.
- Good choice of q is the main challenge for the algorithm.

Optimization of q

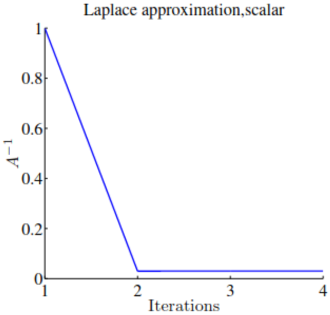
Distribution q can be set using neural networks.

- **Main requirements:** existance of $p(x|x'), p(x'|x) \to$ the distribution must be invertible.
- Neural network in a form of $\mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{x} + \mathbf{g}(\mathbf{x}, \mathbf{w})$ is a flow and invertible.

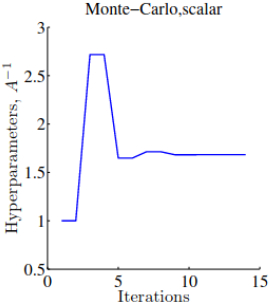
Optimization variants:

- Entropy * Acceptance rate (Li et al., 2020)
- GAN between empirical distribution and q (Song et al., 2017).

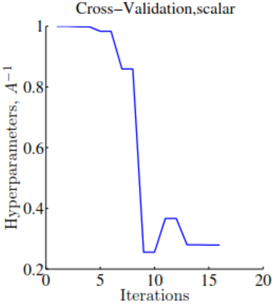
Hyperparameter selection for linear model



(a) Laplace approximation

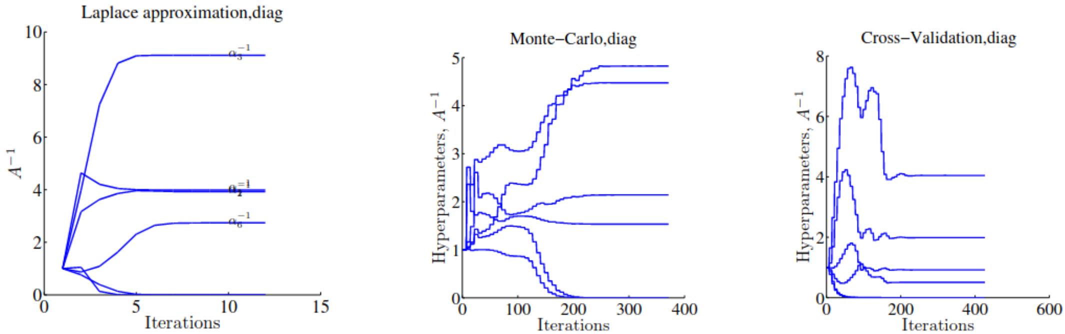


(b) Monte-Carlo



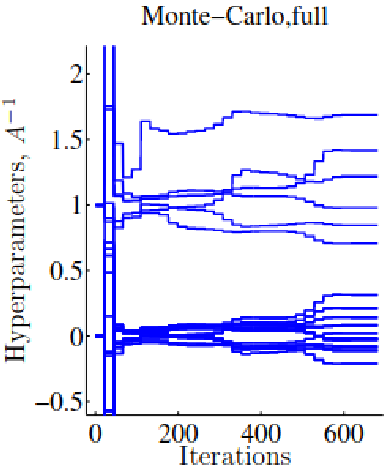
(c) Cross validation

Hyperparameter selection for linear model

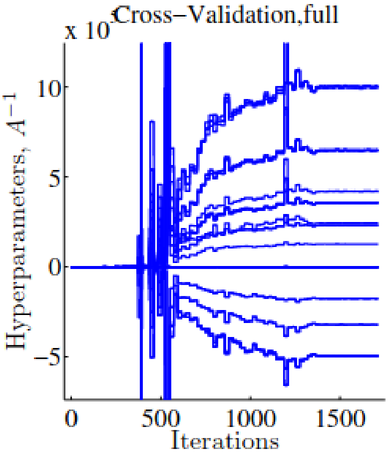


(a) Laplace approximation (b) Monte-Carlo (c) Cross validation

Hyperparameter selection for linear model



(a) Monte-Carlo



(b) Cross validation

References

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – T. 128. – №. 9.
- MacKay D. J. C., Mac Kay D. J. C. Information theory, inference and learning algorithms. – Cambridge university press, 2003.
- Atanov A. et al. The deep weight prior //7th International Conference on Learning Representations, ICLR 2019. – 2019.
- Kuznetsov M., Tokmakova A., Strijov V. Analytic and stochastic methods of structure parameter estimation //Informatica. – 2016. – T. 27. – №. 3. – C. 607-624.
- Coin example: https://towardsdatascience.com/visualizing-beta-distribution-7391c18031f1
- Jefreys distribution: https://medium.datadriveninvestor.com/firths-logistic-regression-classification-with-datasets-that-are-small-imbalanced-or-separated-49d7782a13f1
- Atanov A. et al. The deep weight prior //arXiv preprint arXiv:1810.06943. – 2018.
- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – T. 4. – №. 4. – C. 738.

- Kuznetsov M., Tokmakova A., Strijov V. Analytic and stochastic methods of structure parameter estimation //Informatica. – 2016. – T. 27. – №. 3. – C. 607-624.

- Li Z., Chen Y., Sommer F. T. A neural network mcmc sampler that maximizes proposal entropy //arXiv preprint arXiv:2010.03587. – 2020.

Organizational issues

Current hometask

- Nobody filled activities
- Please do it until next meeting, otherwise penalty will be applied
- Машалов and Насыров — ???
- The scores were corrected (see the course page)
- The project schedule (preliminary) is added (see the course page)

Next hometask: presentation

- For all teams and their members: make presentations of your projects
- The presentation must cover
  - Project description (maybe more detailed than I gave to you)
  - Name of the project library
  - Scheme of the project (what will be the classes, how it will be integrated, what's the stack)
  - Brief algorithm description (from 1 to 4 slides for all the algorithms, other people must be able to understand the idea of all the algorithms)
  - Idea for demo/basic code
- Time limit: 10 min

Next hometask: for people who are wrapping the library

- Create a repository in intsystems
- Keep in mind the future library must support documentation deploy and auto-testing. You can make the repository by yourself OR
  - Use Andrey Grabovoy's template: https://github.com/intsystems/ProjectTemplate
    - Please turn off autodeploy of github pages
  - Use my template: https://github.com/intsystems/SoftwareTemplate-simplified
  - Use any other template, see for example: https://github.com/LauzHack/pytorch_project_template
- Think about stack:
  - Codestyle (linters?)
  - Documentation engines (mkdocs? shpinx?)
  - Test libraries (built-in unittest? pytest?)
- Please make it w.r.t. to the manual

RTFM

Please make it w.r.t. to the manual

Next hometask: for people who are planning the library

- Create a document in the repository with the following information (the same as in the presentation, but maybe with more details)
  - Project name
  - Architecture of the project: what classes must be implemented? How they should interact?
  - Describe all the public functions and class methods you are planing to implement, with annotations.
  - What are the libraries you are planning to use and/or integrate?
- In perfect case, the member who is implementing the algorithm can write the code just by your architecture description.
- Note, the document can be improved/changed in the future, but I will score you and other members of the team on the correspondence of the proposed structure and the final algorithm implementation.

For other activites

- I will add some comments on the next meeting for all the activities.
- Basic code: deadline is 29th of October. You can start thinking together with demo code
- Blog post: very drafty version must be ready on the 29th of October. Have a look at https://github.com/intsystems/IDA/tree/main
- Documentation: structure of the documentation must be ready on the 29th of October. Discuss the engine of the documentation with a person responsible for the repo.
- Tests: must be ready on the 19th of November with some high coverage. Discuss the framework of the tests with a person responsible for the repo.
- Algorithms: must be ready on the 19th of November.