Bayesian Inference and Deep Learning

Richard E. Turner (ret26@cam.ac.uk)

Machine Learning Group, University of Cambridge

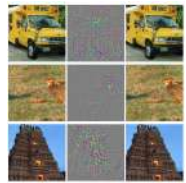
Robust Deep Learning

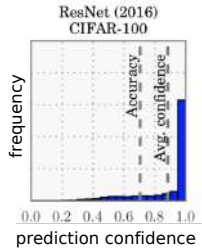
fragile (adversarial examples)

Robust Deep Learning

fragile (adversarial examples)

well calibrated uncertainty estimates:
deep learning is often confidently wrong

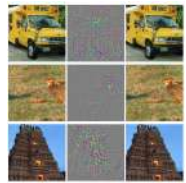


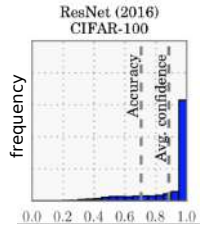


Robust Deep Learning

fragile (adversarial examples)

well calibrated uncertainty estimates:
deep learning is often confidently wrong





Data-Efficient Deep Learning

small data, big models (few-shot learning
and reinforcement learning)

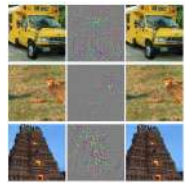
leverage heterogenous data sources (multi-task learning)

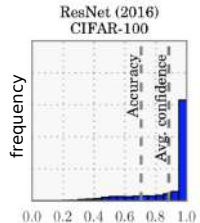
Limitations of Deep Learning

Robust Deep Learning

fragile (adversarial examples)

well calibrated uncertainty estimates:
deep learning is often confidently wrong





Data-Efficient Deep Learning

small data, big models (few-shot learning
and reinforcement learning)

leverage heterogenous data sources (multi-task learning)

Flexible Deep Learning

continual learning (online learning & model building)

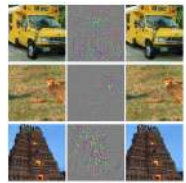
active learning (RL exploration-exploitation)

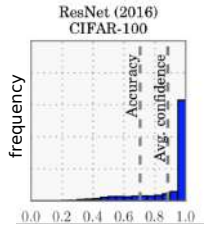
Limitations of Deep Learning

Robust Deep Learning

fragile (adversarial examples)

well calibrated uncertainty estimates:
deep learning is often confidently wrong





Data-Efficient Deep Learning

small data, big models (few-shot learning
and reinforcement learning)

leverage heterogenous data sources (multi-task learning)

Flexible Deep Learning

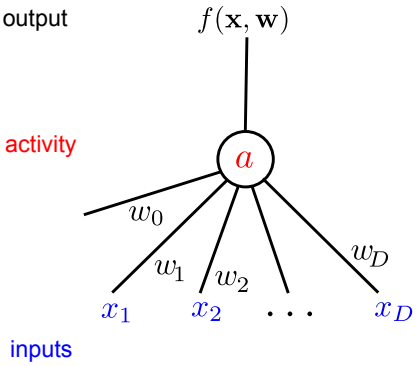
continual learning (online learning & model building)

active learning (RL exploration-exploitation)

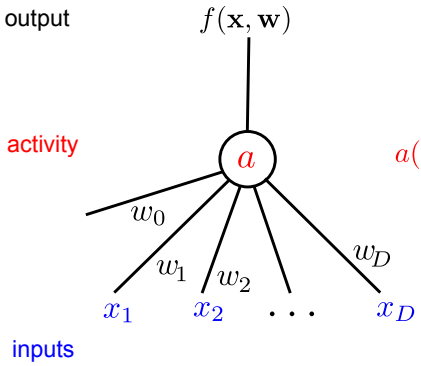
probabilistic modelling
+
probabilistic inference

Logistic Regression as a motivating example

Logistic regression: A single neuron

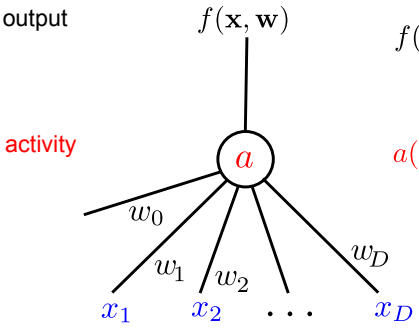


Logistic regression: A single neuron



$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$

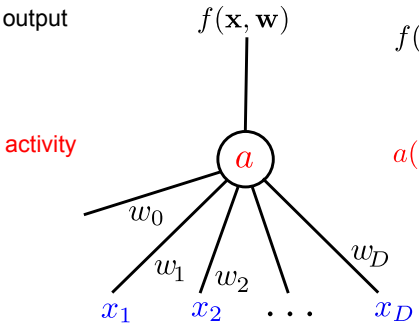
Logistic regression: A single neuron



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

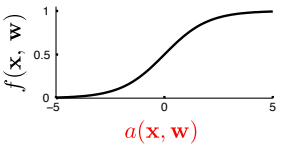
$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$

Logistic regression: A single neuron

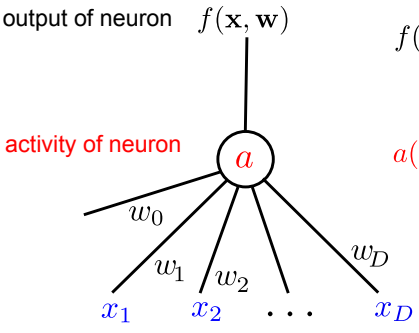


$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$

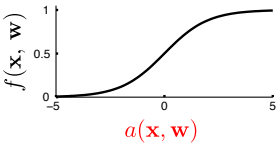


Logistic regression: A single neuron

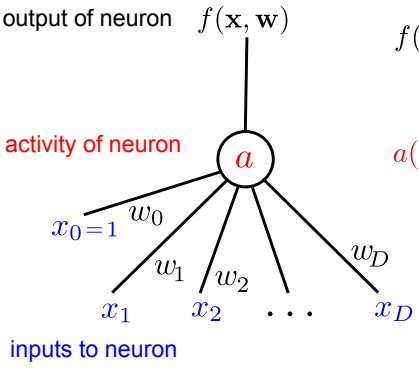


$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$

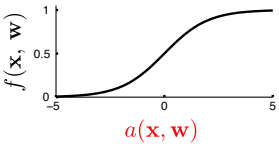


Logistic regression: A single neuron

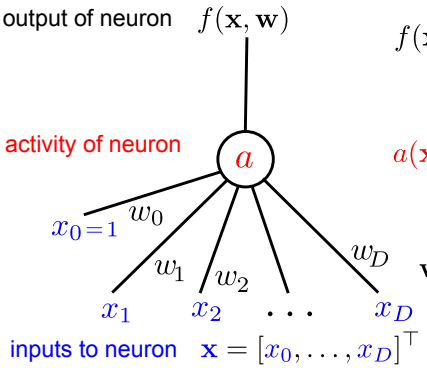


$$f(\mathbf{x}, \mathbf{w}) = P(y = 1 | \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$

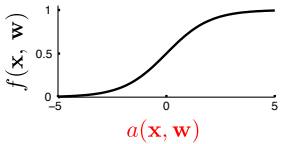


Logistic regression: A single neuron



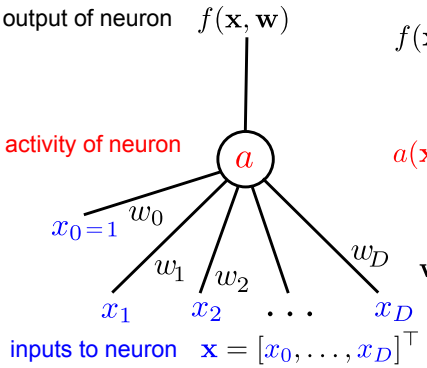
$$f(\mathbf{x}, \mathbf{w}) = P(y = 1 | \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d$$



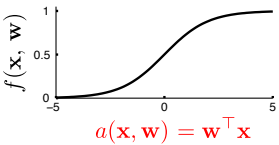
$$\mathbf{w} = [w_0, \ldots, w_D]^\top$$

Logistic regression: A single neuron



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1 | \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

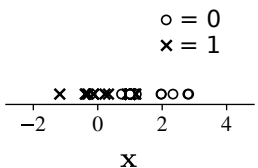
$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$



$$\mathbf{w} = [w_0, \ldots, w_D]^\top$$

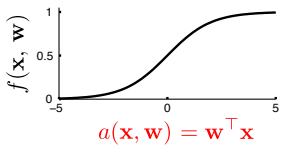
Logistic regression: Maximum Likelihood Estimation

$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$



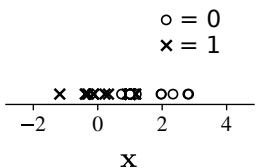
$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$



Logistic regression: Maximum Likelihood Estimation

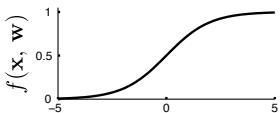
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$

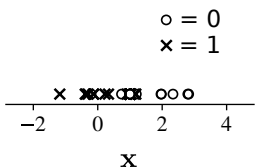
$$= w_0 + w_1 x_1$$



Logistic regression: Maximum Likelihood Estimation

observe data: estimate parameters

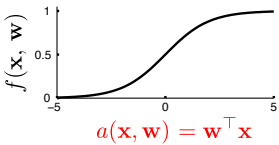
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$

$$= w_0 + w_1 x_1$$

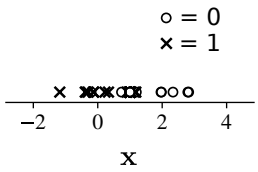


maximum likelihood estimate: parameters that make observed data most probable

Logistic regression: Maximum Likelihood Estimation

observe data: estimate parameters

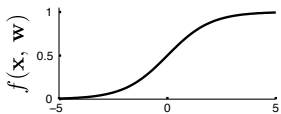
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$

$$= w_0 + w_1 x_1$$



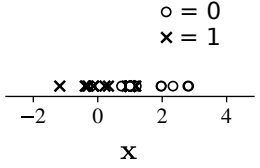
maximum likelihood estimate: parameters that make observed data most probable

$$P(\mathcal{Y}|\mathbf{w}, \mathcal{X}) = \prod_{n=1}^{N} P(y_n|\mathbf{w}, \mathbf{x}_n) = \prod_{n=1}^{N} f(\mathbf{w}^\top \mathbf{x}_n)^{y_n} (1 - f(\mathbf{w}^\top \mathbf{x}_n))^{(1-y_n)}$$

Logistic regression: Maximum Likelihood Estimation

observe data: estimate parameters

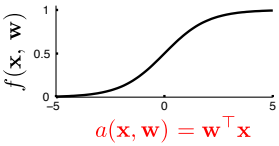
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$



$$f(\mathbf{x}, \mathbf{w}) = P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

$$a(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{d=1}^{D} w_d x_d = \mathbf{w}^\top \mathbf{x}$$

$$= w_0 + w_1 x_1$$



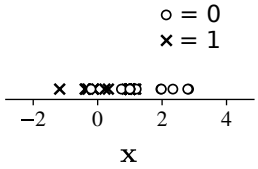
maximum likelihood estimate: parameters that make observed data most probable

$$P(\mathcal{Y}|\mathbf{w}, \mathcal{X}) = \prod_{n=1}^{N} P(y_n|\mathbf{w}, \mathbf{x}_n) = \prod_{n=1}^{N} f(\mathbf{w}^\top \mathbf{x}_n)^{y_n} (1 - f(\mathbf{w}^\top \mathbf{x}_n))^{(1-y_n)}$$

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w}, \mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top \mathbf{x}_n) + (1 - y_n) \log(1 - f(\mathbf{w}^\top \mathbf{x}_n)) \right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x}, \mathbf{w}) = P\left(y = 1 | \mathbf{w}, \mathbf{x}\right) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})} = \frac{1}{1 + \exp(-(w_0 + w_1 x_1))}$$

data:

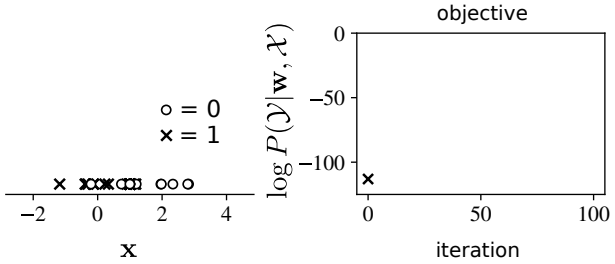
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w}, \mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top \mathbf{x}_n) + (1 - y_n) \log(1 - f(\mathbf{w}^\top \mathbf{x}_n)) \right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

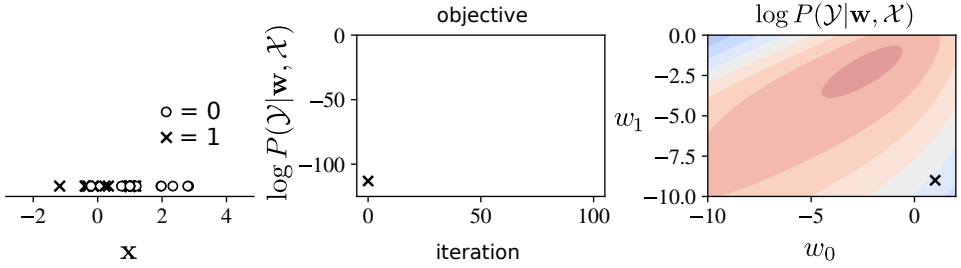
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P(y=1|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x})} = \frac{1}{1+\exp(-(w_0 + w_1 x_1))}$$

data:

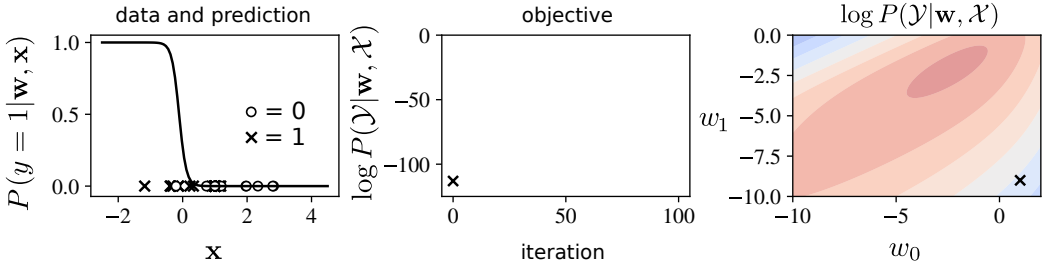
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top \mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top \mathbf{x}_n)) \right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

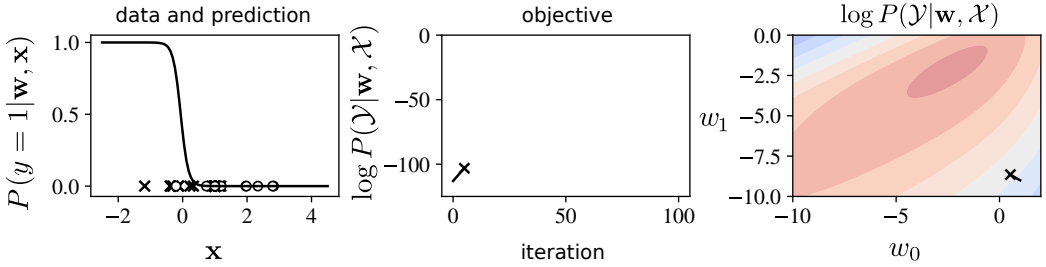
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

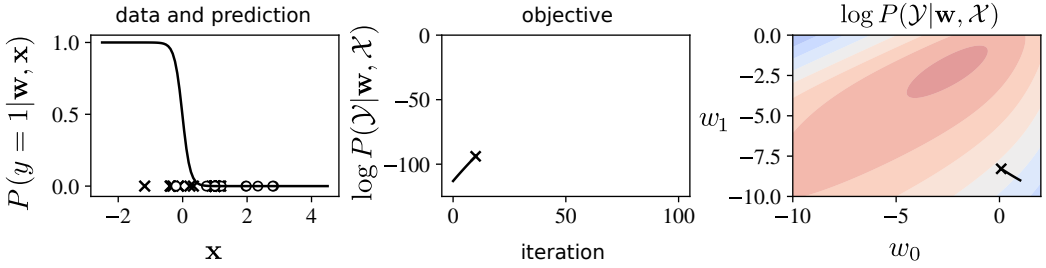
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

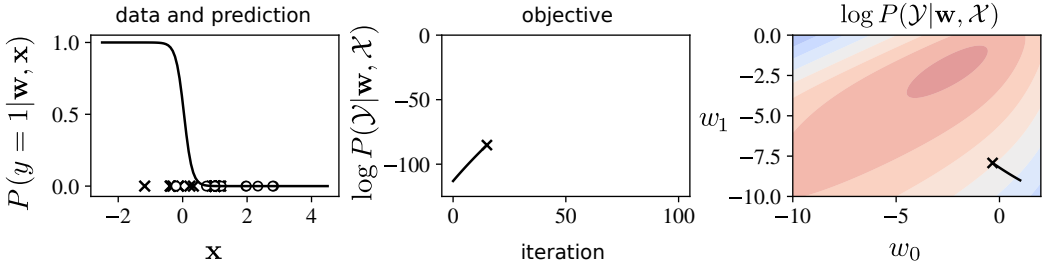
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

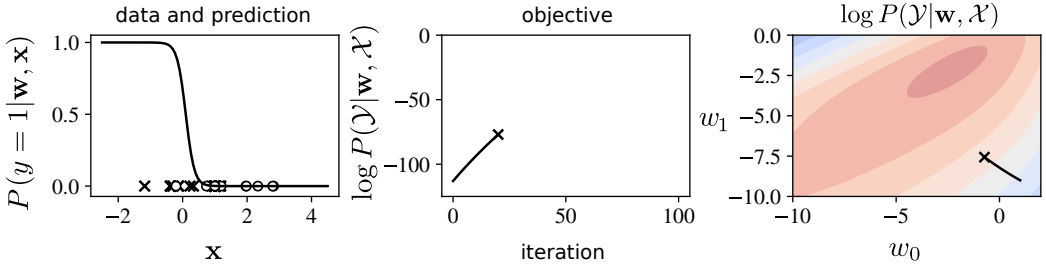
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

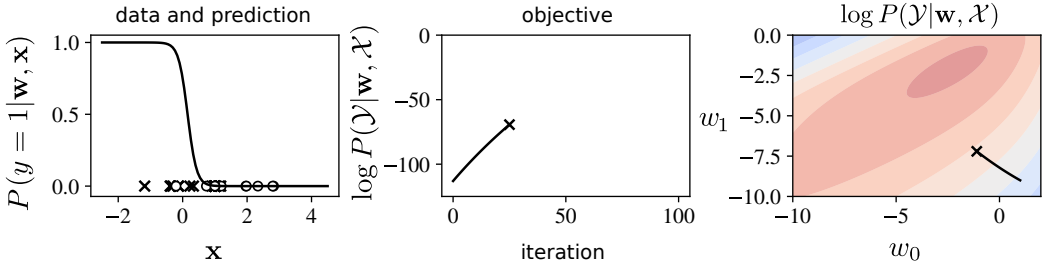
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

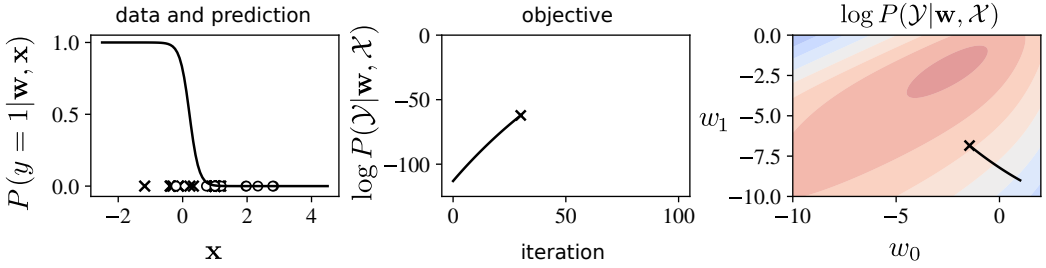
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

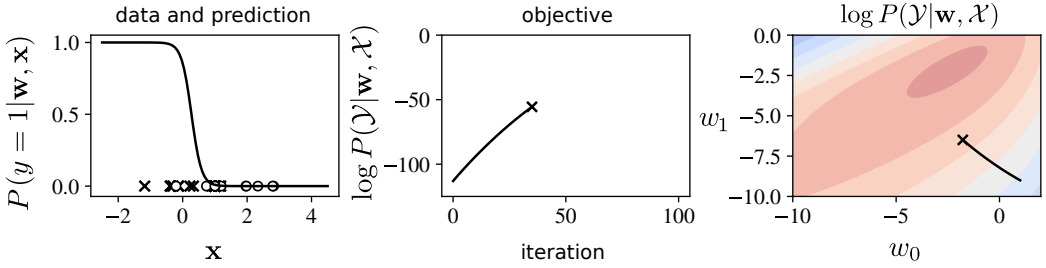
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\mathsf{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

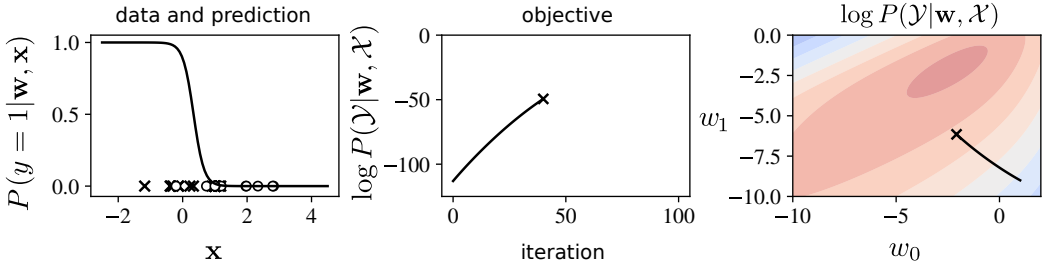
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

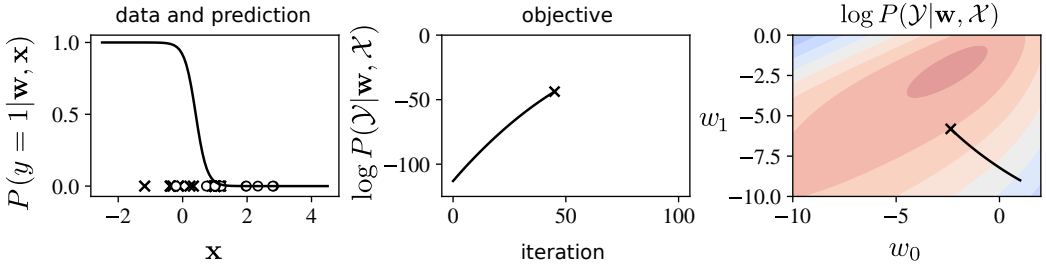
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

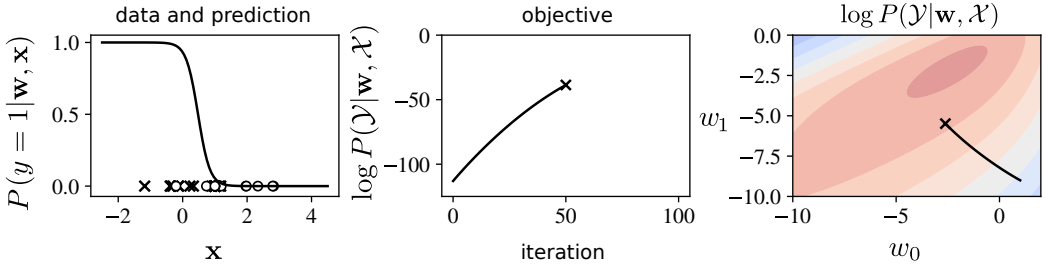
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

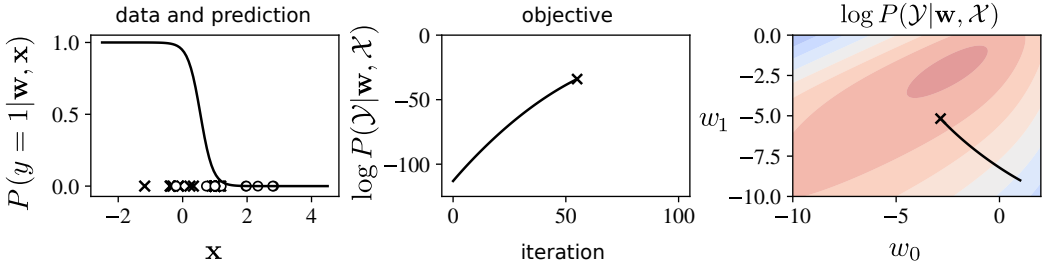
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

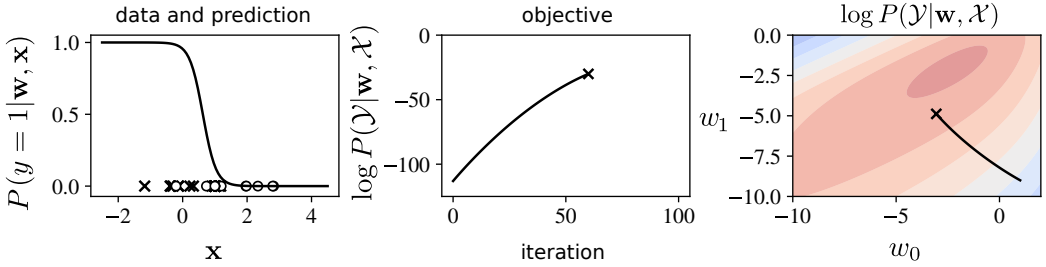
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

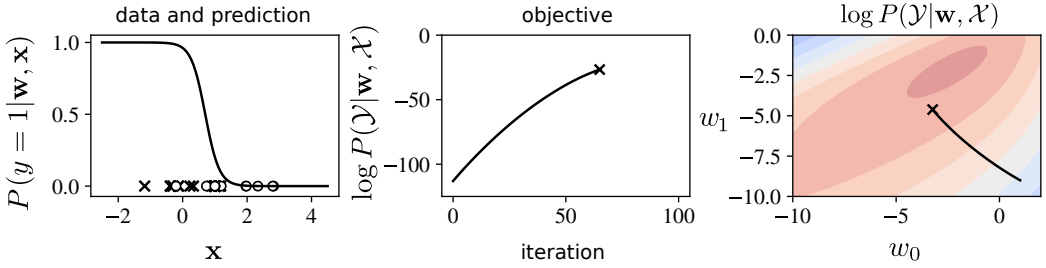
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

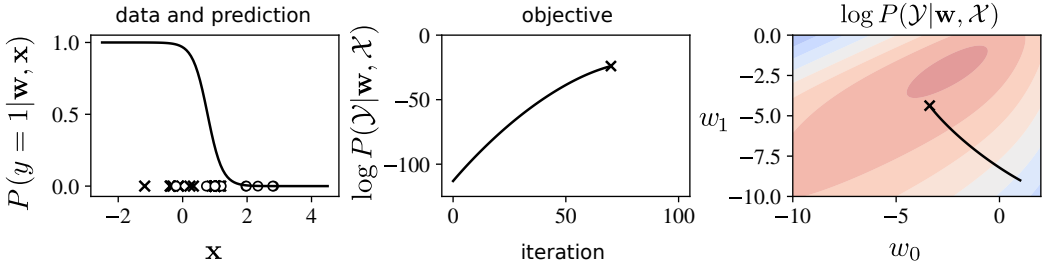
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

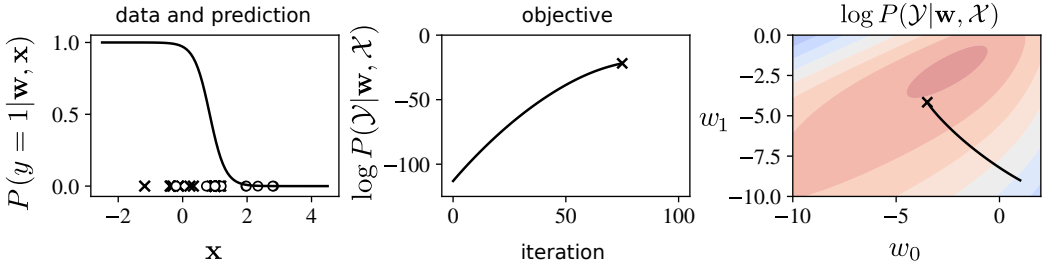
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

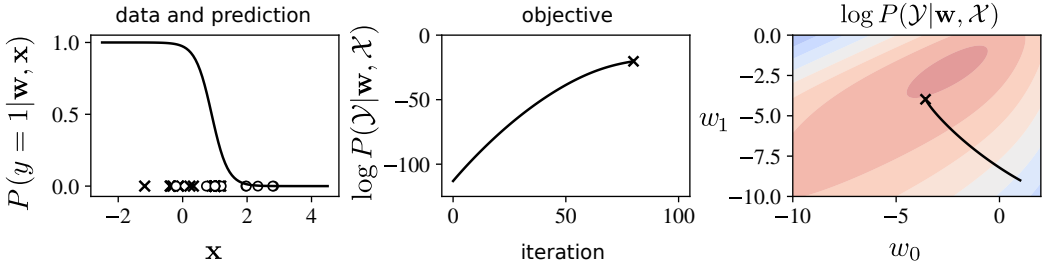
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

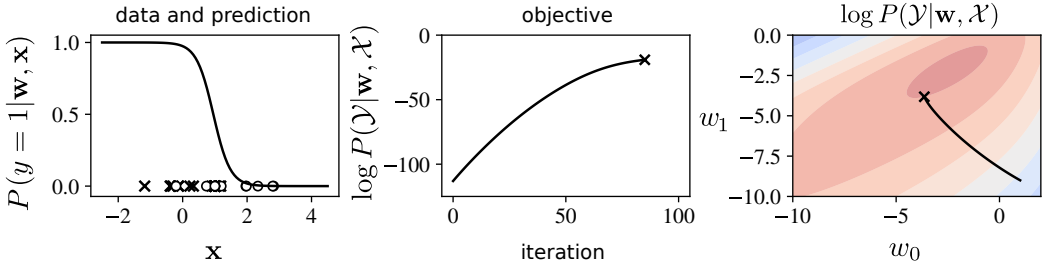
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\mathsf{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

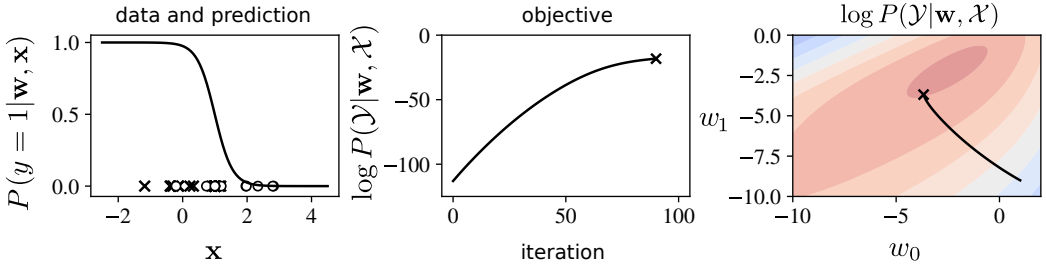
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

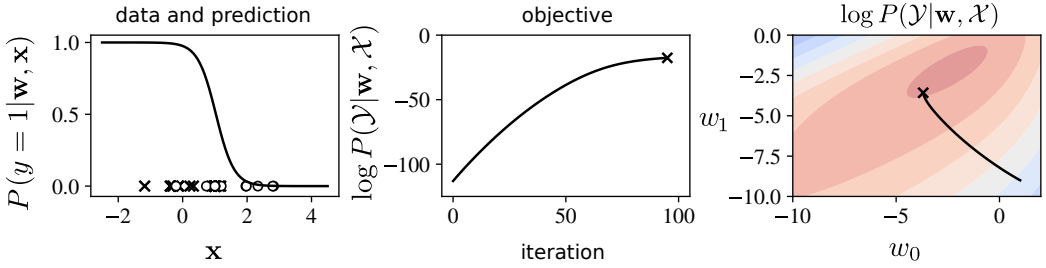
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

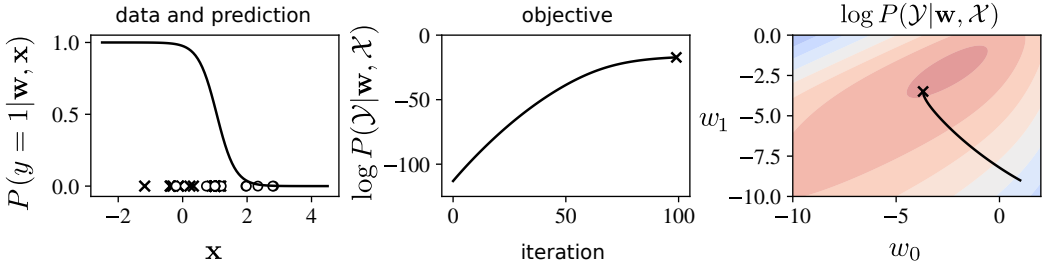
$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\mathsf{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Logistic regression: Maximum Likelihood learning

observe data: estimate parameters



model:

$$f(\mathbf{x},\mathbf{w}) = P\left(y=1|\mathbf{w},\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x})} = \frac{1}{1+\exp(-(w_0+w_1x_1))}$$

data:

$$\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$$
$$\mathcal{Y} = \{y_n\}_{n=1}^N$$

maximum likelihood estimate: parameters that make observed data most probable

$$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}|\mathbf{w},\mathcal{X}) = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \left[y_n \log f(\mathbf{w}^\top\mathbf{x}_n) + (1-y_n)\log(1-f(\mathbf{w}^\top\mathbf{x}_n))\right]$$

Bayesian approaches to logistic regression

Probabilistic model

encodes prior assumptions in a recipe for generating datasets

Bayesian approaches to logistic regression

Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$

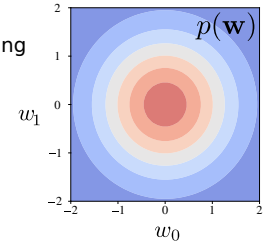
Bayesian approaches to logistic regression

Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma^2_{\mathbf{w}} \mathrm{I})$$

$$P(y = 1 | \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$



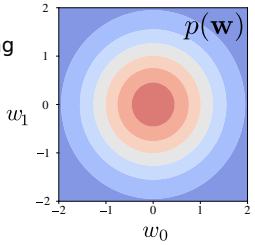
Bayesian approaches to logistic regression

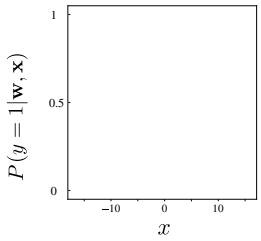
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





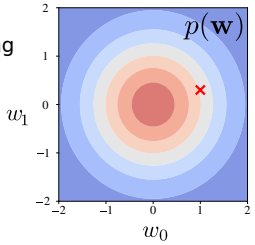
Bayesian approaches to logistic regression

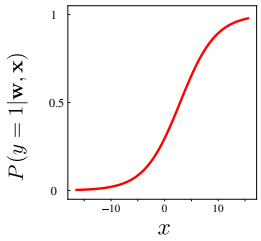
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





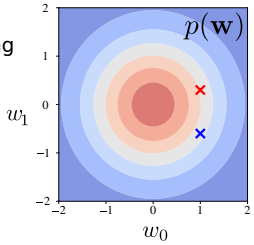
Bayesian approaches to logistic regression

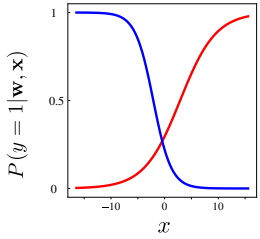
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





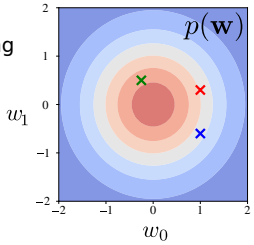
Bayesian approaches to logistic regression

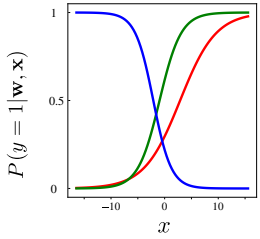
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





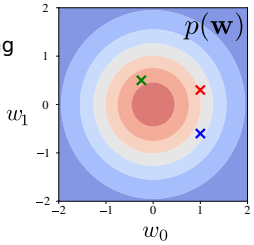
Bayesian approaches to logistic regression

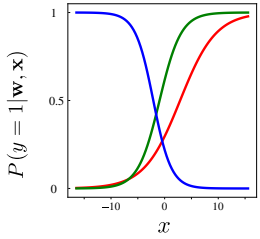
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

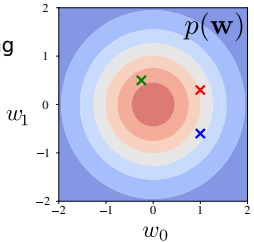
Bayesian approaches to logistic regression

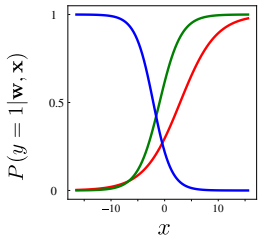
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data

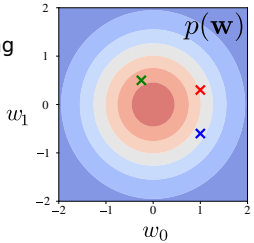
Bayesian approaches to logistic regression

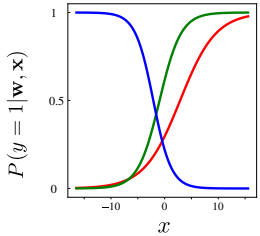
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data

only way to be coherent, Cox 1946

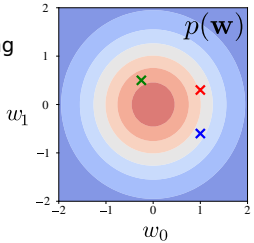
Bayesian approaches to logistic regression

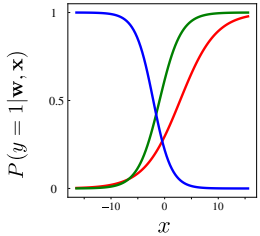
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data

only way to be coherent, Cox 1946

only way to protect against Dutch books, Ramsey 1926

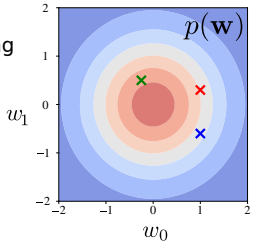
Bayesian approaches to logistic regression

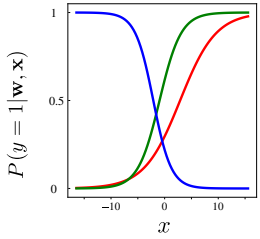
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data
2. apply the sum and product rules of probability to compute the plausibility of any setting of any unknown variable

only way to be coherent, Cox 1946

only way to protect against Dutch books, Ramsey 1926

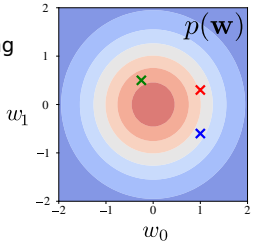
Bayesian approaches to logistic regression

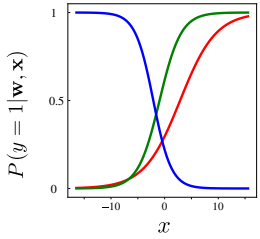
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data
2. apply the sum and product rules of probability to compute the plausibility of any setting of any unknown variable

only way to be coherent, Cox 1946

only way to protect against Dutch books, Ramsey 1926

sum rule: $P(A|C) = \sum_B P(A, B|C)$

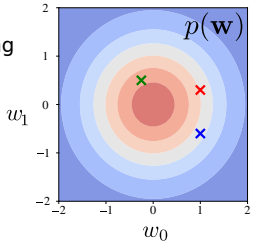
Bayesian approaches to logistic regression

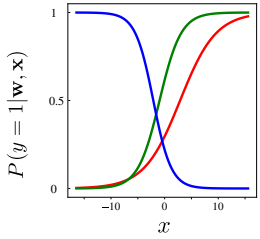
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data

   only way to be coherent, Cox 1946

   only way to protect against Dutch books, Ramsey 1926

2. apply the sum and product rules of probability to compute the plausibility of any setting of any unknown variable

sum rule: $P(A|C) = \sum_B P(A, B|C)$

product rule: $P(A, B|C) = P(B|C)P(A|B, C) = P(A|C)P(B|A, C)$

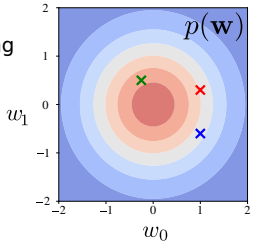
Bayesian approaches to logistic regression

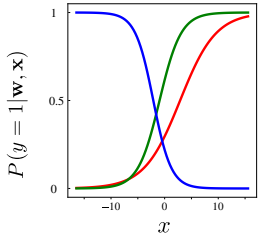
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

1. "right" answer is a probability distribution over all possible settings of the weights that indicates the plausiblity of that setting given data

   only way to be coherent, Cox 1946

   only way to protect against Dutch books, Ramsey 1926

2. apply the sum and product rules of probability to compute the plausibility of any setting of any unknown variable

implies

sum rule: $P(A|C) = \sum_B P(A, B|C)$

product rule: $P(A, B|C) = P(B|C)P(A|B, C) = P(A|C)P(B|A, C)$

Bayes' rule: $P(A|B, C) = \frac{1}{P(B|C)} P(A|C)P(B|A, C)$

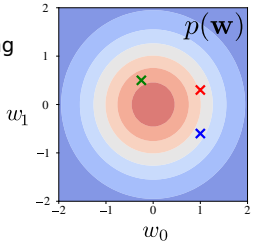
The posterior distribution over weights

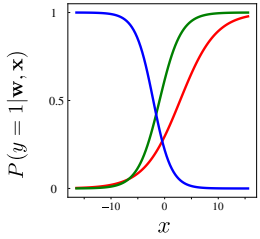
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$$

posterior
what we know
after seeing data

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

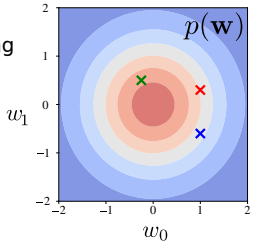
The posterior distribution over weights

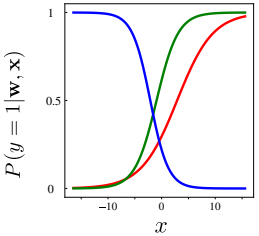
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})} p(\mathbf{w}|\mathcal{X}) P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

$A = \mathbf{w} \quad B = \mathcal{Y} \quad C = \mathcal{X}$

posterior
what we know
after seeing data

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

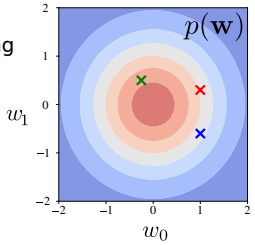
The posterior distribution over weights

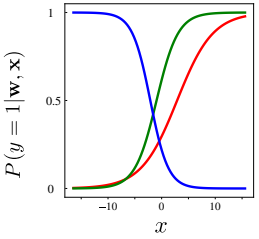
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

structure of model

$$A = \mathbf{w} \quad B = \mathcal{Y} \quad C = \mathcal{X}$$

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\, P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

posterior
what we know
after seeing data

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

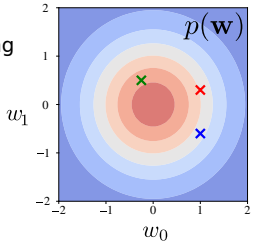
The posterior distribution over weights

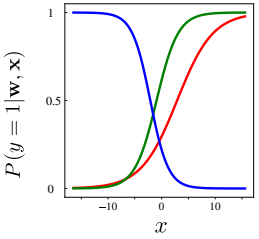
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

structure of model

$A = \mathbf{w}$ $B = \mathcal{Y}$ $C = \mathcal{X}$

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})} p(\mathbf{w}) \; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

posterior ∝ prior × likelihood of $\mathbf{w}$

posterior: what we know after seeing data

prior: what we knew before seeing data

likelihood of $\mathbf{w}$: what the data told us

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

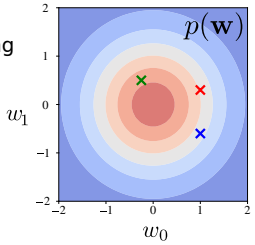
The predictive distribution over class labels

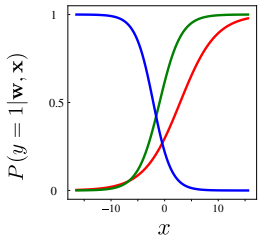
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

$$p(y^*|x^*, \mathcal{Y}, \mathcal{X})$$

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

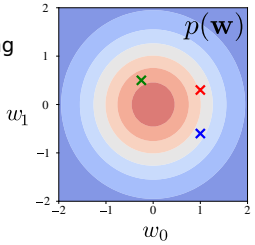
The predictive distribution over class labels

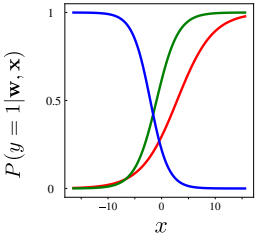
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

$$p(y^*|x^*, \mathcal{Y}, \mathcal{X})$$

$A = y^* \quad B = \mathbf{w} \quad C = x^*, \mathcal{Y}, \mathcal{X}$

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

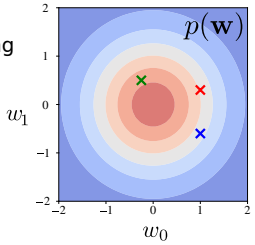
The predictive distribution over class labels

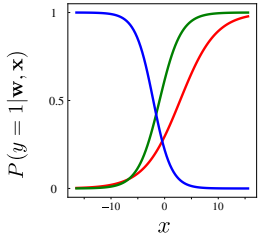
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

$$p(y^*|x^*, \mathcal{Y}, \mathcal{X}) = \int p(y^*, \mathbf{w}|x^*, \mathcal{Y}, \mathcal{X}) \mathrm{d}\mathbf{w}$$

$A = y^* \quad B = \mathbf{w} \quad C = x^*, \mathcal{Y}, \mathcal{X}$

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

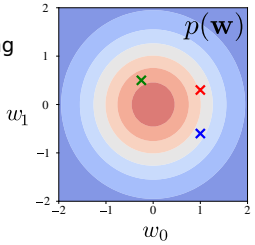
The predictive distribution over class labels

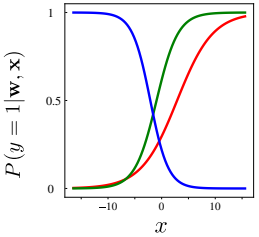
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

$$p(y^*|x^*, \mathcal{Y}, \mathcal{X}) = \int p(y^*, \mathbf{w}|x^*, \mathcal{Y}, \mathcal{X}) \mathrm{d}\mathbf{w} = \int p(\mathbf{w}|x^*, \mathcal{Y}, \mathcal{X})\, p(y^*|\mathbf{w}, x^*, \mathcal{Y}, \mathcal{X}) \mathrm{d}\mathbf{w}$$

$A = y^* \quad B = \mathbf{w} \quad C = x^*, \mathcal{Y}, \mathcal{X}$

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

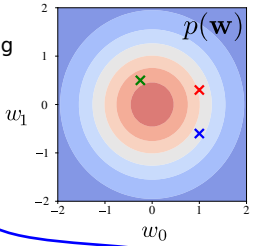
The predictive distribution over class labels

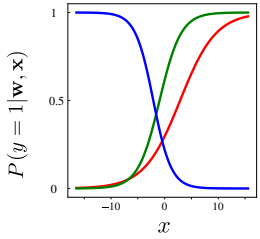
Probabilistic model

encodes prior assumptions in a recipe for generating datasets

$$p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$$

$$P(y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-a(\mathbf{x}, \mathbf{w}))}$$





Probabilistic inference

$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$$

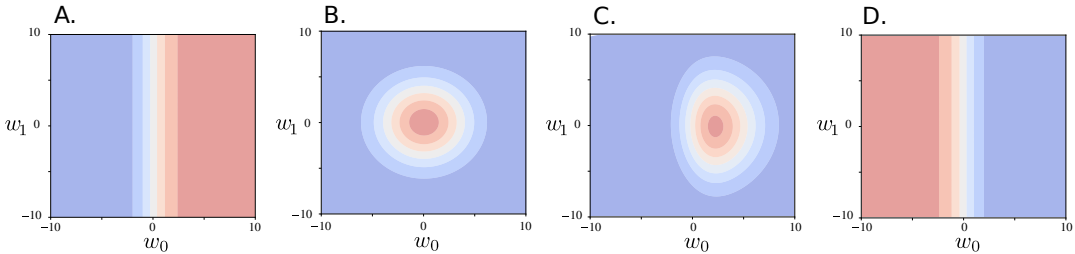
structure of model

$$p(y^*|x^*, \mathcal{Y}, \mathcal{X}) = \int p(y^*, \mathbf{w}|x^*, \mathcal{Y}, \mathcal{X}) \mathrm{d}\mathbf{w} = \int p(\mathbf{w}|\mathcal{Y}, \mathcal{X})\, p(y^*|\mathbf{w}, x^*) \mathrm{d}\mathbf{w}$$

$$A = y^* \quad B = \mathbf{w} \quad C = x^*, \mathcal{Y}, \mathcal{X}$$

sum rule:	$P(A\|C) = \sum_B P(A, B\|C)$
product rule:	$P(A, B\|C) = P(B\|C)P(A\|B, C) = P(A\|C)P(B\|A, C)$
Bayes' rule:	$P(A\|B, C) = \frac{1}{P(B\|C)} P(A\|C)P(B\|A, C)$

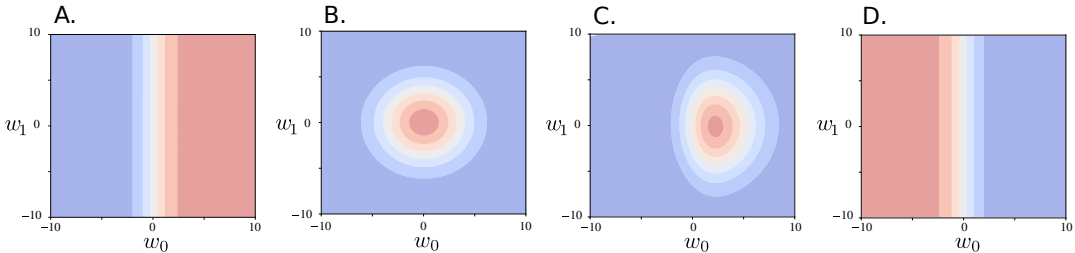
Quiz!



1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma^2_{\mathbf{w}}\mathrm{I})$
2. $P(y = 1|\mathbf{w}, x = 0)$
3. $P(y = 0|\mathbf{w}, x = 0)$
4. $p(\mathbf{w}|x = 0, y = 1)$

$$P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$$

Quiz!

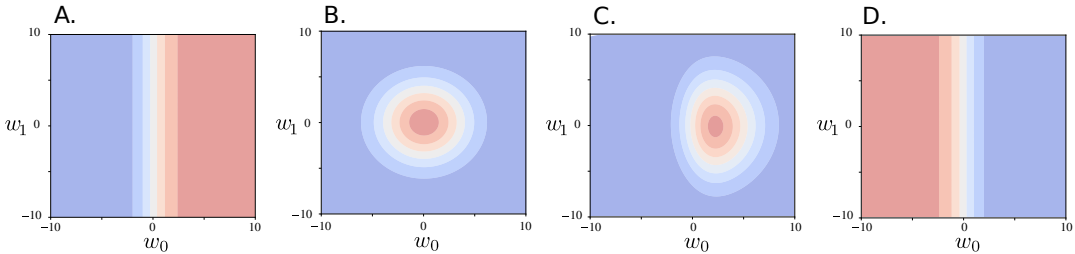


B.

1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma^2_{\mathbf{w}}\mathrm{I})$
2. $P(y = 1|\mathbf{w}, x = 0)$
3. $P(y = 0|\mathbf{w}, x = 0)$
4. $p(\mathbf{w}|x = 0, y = 1)$

$$P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$$

Quiz!



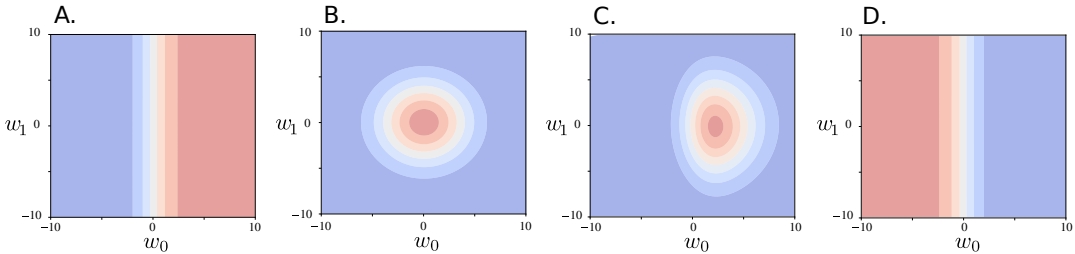
B. 1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma^2_{\mathbf{w}}\mathrm{I})$

A. 2. $P(y = 1|\mathbf{w}, x = 0)$ $\quad P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$

3. $P(y = 0|\mathbf{w}, x = 0)$

4. $p(\mathbf{w}|x = 0, y = 1)$

Quiz!



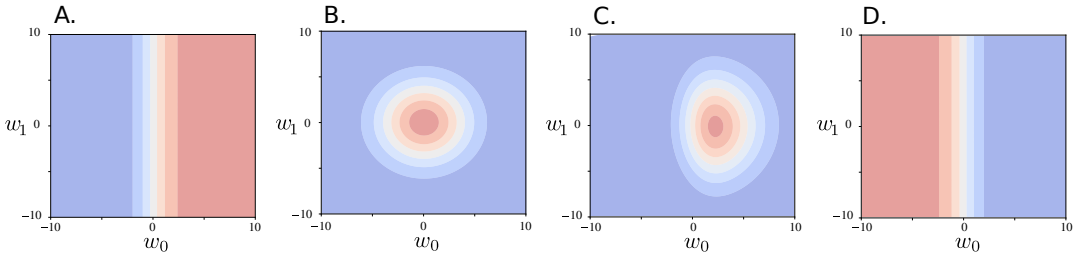
B. 1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma^2_{\mathbf{w}}\mathrm{I})$

A. 2. $P(y = 1|\mathbf{w}, x = 0)$ $\quad P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$

D. 3. $P(y = 0|\mathbf{w}, x = 0)$

4. $p(\mathbf{w}|x = 0, y = 1)$

Quiz!



B. 1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$

A. 2. $P(y = 1|\mathbf{w}, x = 0)$ $\qquad P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$

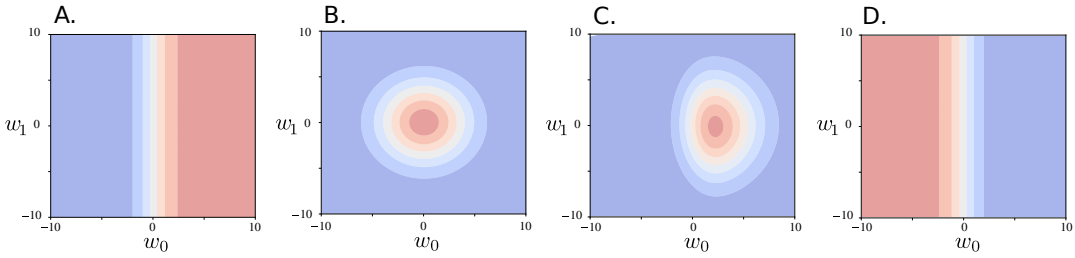
D. 3. $P(y = 0|\mathbf{w}, x = 0)$

4. $p(\mathbf{w}|x = 0, y = 1) \quad \propto \quad p(\mathbf{w}) \quad \times \quad P(y = 1|\mathbf{w}, x = 0)$

posterior — prior — likelihood of $\mathbf{w}$

what we know after seeing data — what we knew before seeing data — what the data told us

Quiz!



B. 1. $p(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathrm{I})$

A. 2. $P(y = 1|\mathbf{w}, x = 0)$ $\quad P(y = 1|\mathbf{w}, x) = \frac{1}{1+\exp(-(w_1 x + w_0))}$

D. 3. $P(y = 0|\mathbf{w}, x = 0)$

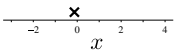
C. 4. $p(\mathbf{w}|x = 0, y = 1) \propto p(\mathbf{w}) \times P(y = 1|\mathbf{w}, x = 0)$

posterior — what we know after seeing data

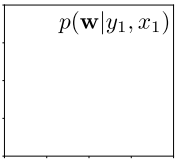
prior — what we knew before seeing data

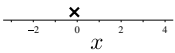
likelihood of $\mathbf{w}$ — what the data told us

Bayesian Inference in Action: 1D Classification Example

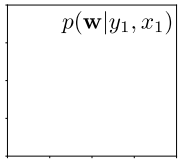


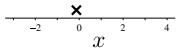
Bayesian Inference in Action: 1D Classification Example





Bayesian Inference in Action: 1D Classification Example





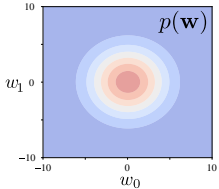
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

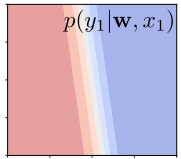
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

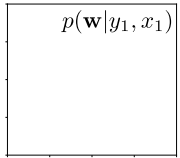
Bayesian Inference in Action: 1D Classification Example

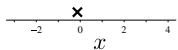


$\times$



$\propto$





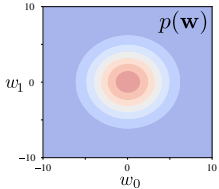
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

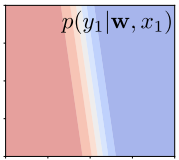
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

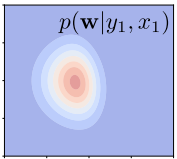
Bayesian Inference in Action: 1D Classification Example

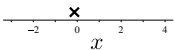


$\times$



$\propto$





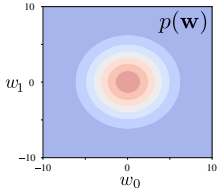
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

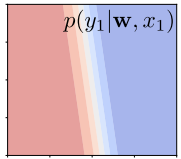
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

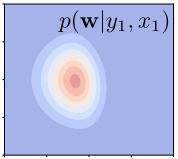
Bayesian Inference in Action: 1D Classification Example

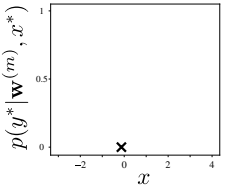


$\times$



$\propto$





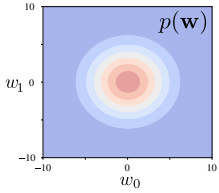
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

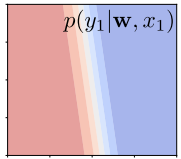
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

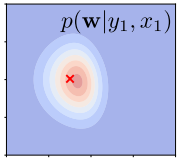
Bayesian Inference in Action: 1D Classification Example

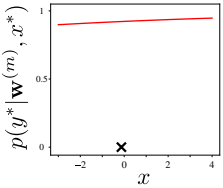


$\times$



$\propto$





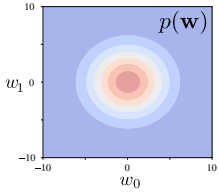
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

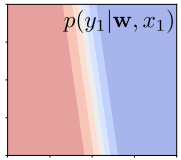
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

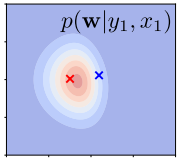
Bayesian Inference in Action: 1D Classification Example

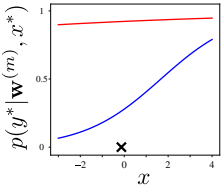


$\times$



$\propto$





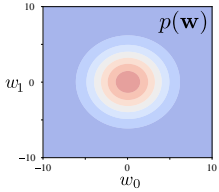
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

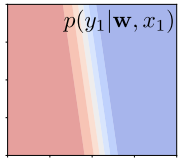
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

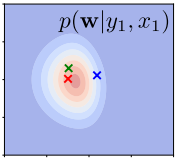
Bayesian Inference in Action: 1D Classification Example

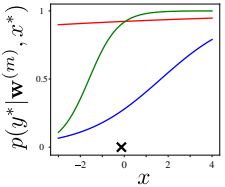


$\times$



$\propto$





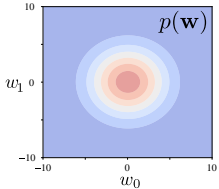
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

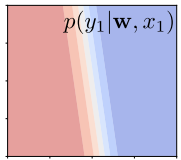
prior
what we knew
before seeing data

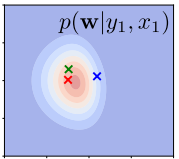
likelihood of $\mathbf{w}$
what the
data told us

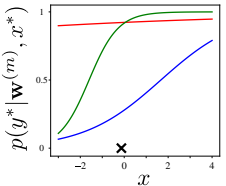
posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example

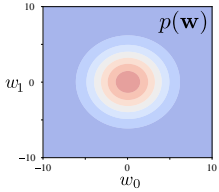


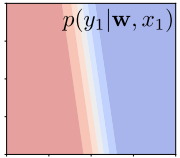


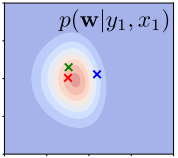


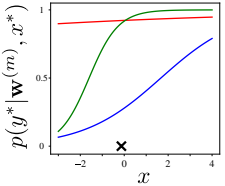


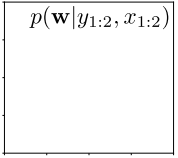
Bayesian Inference in Action: 1D Classification Example

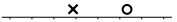




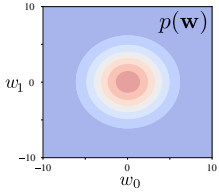




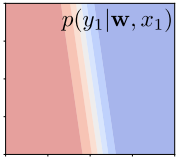




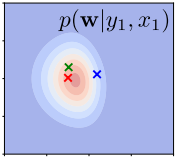
Bayesian Inference in Action: 1D Classification Example

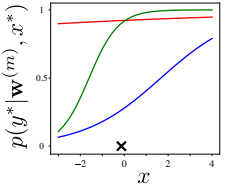


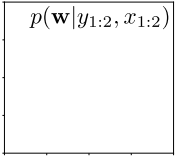
$\times$

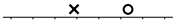


$\propto$









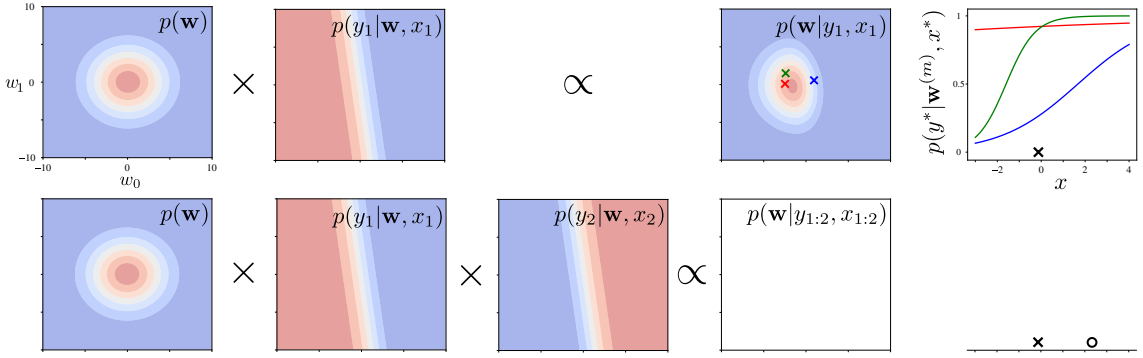
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



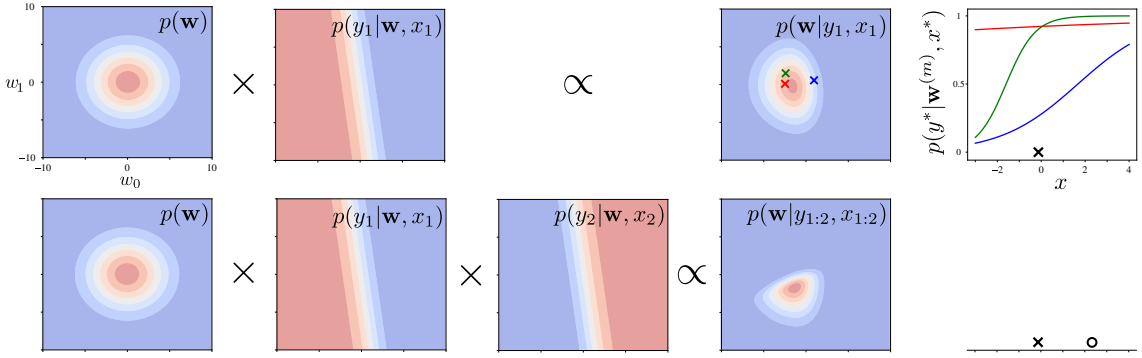
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



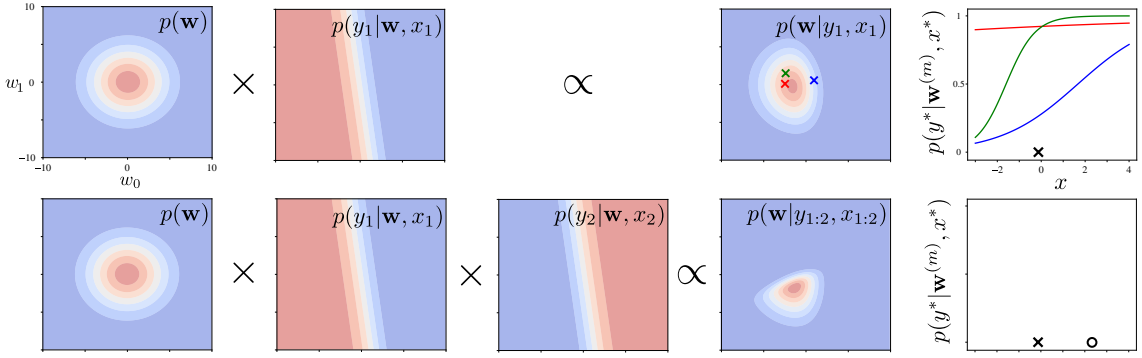
$p(\mathbf{w})$ × $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ ∝ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



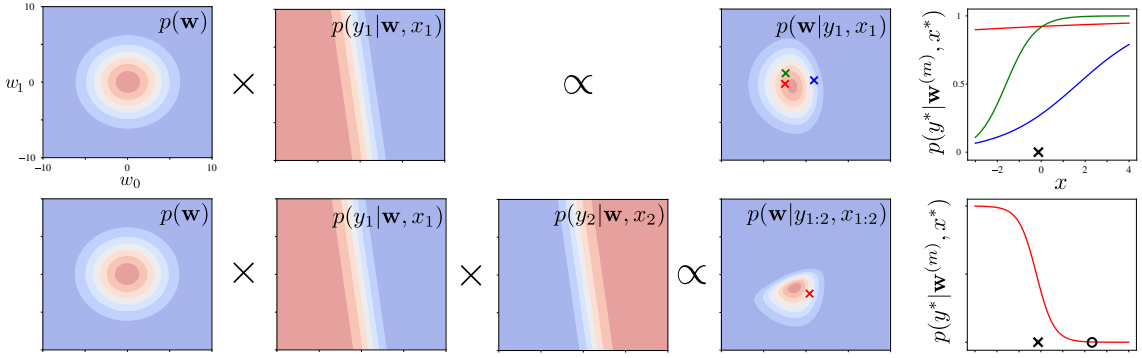
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



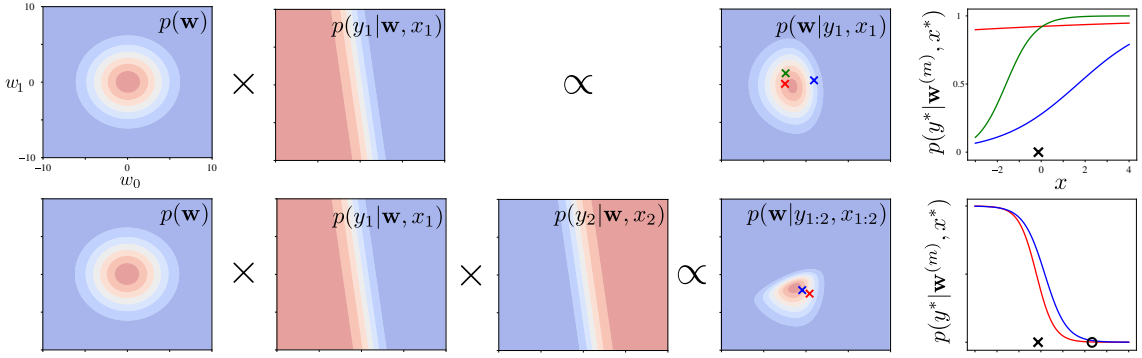
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



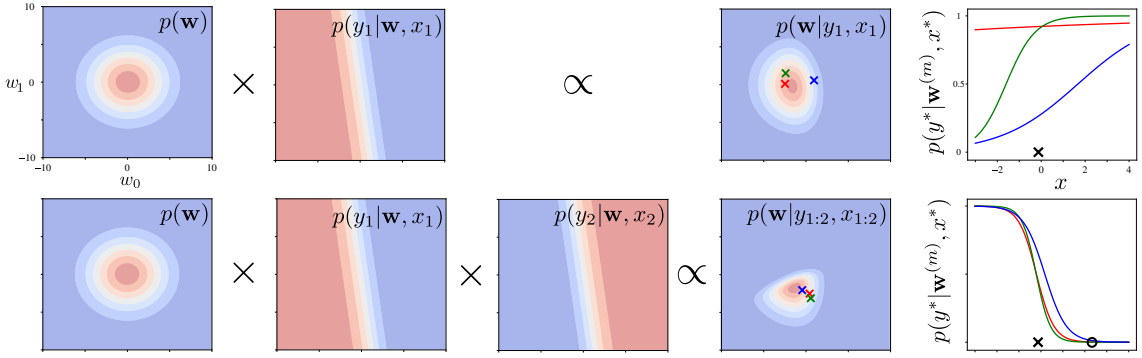
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



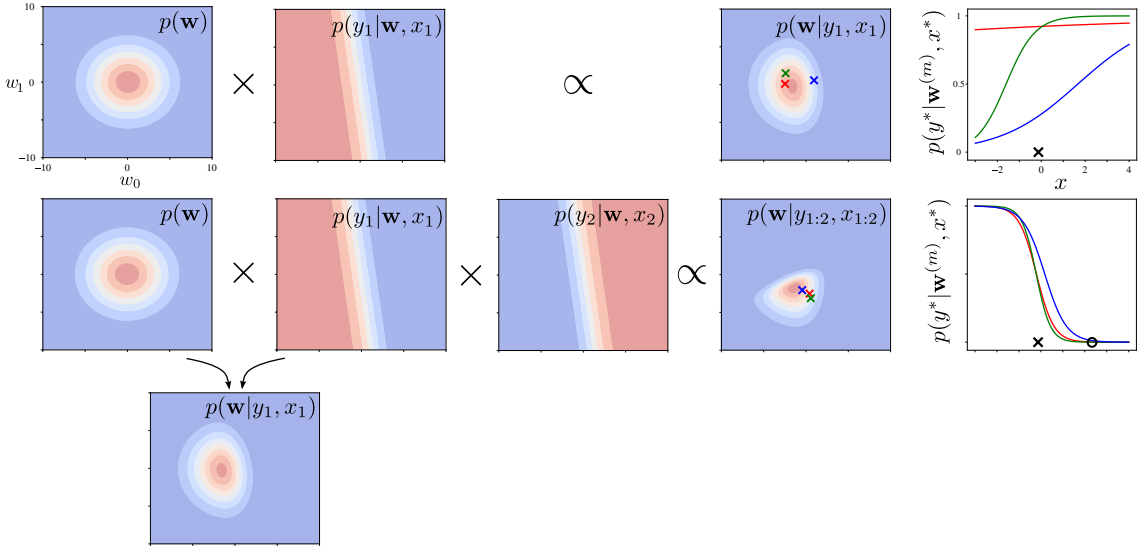
$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

prior
what we knew
before seeing data

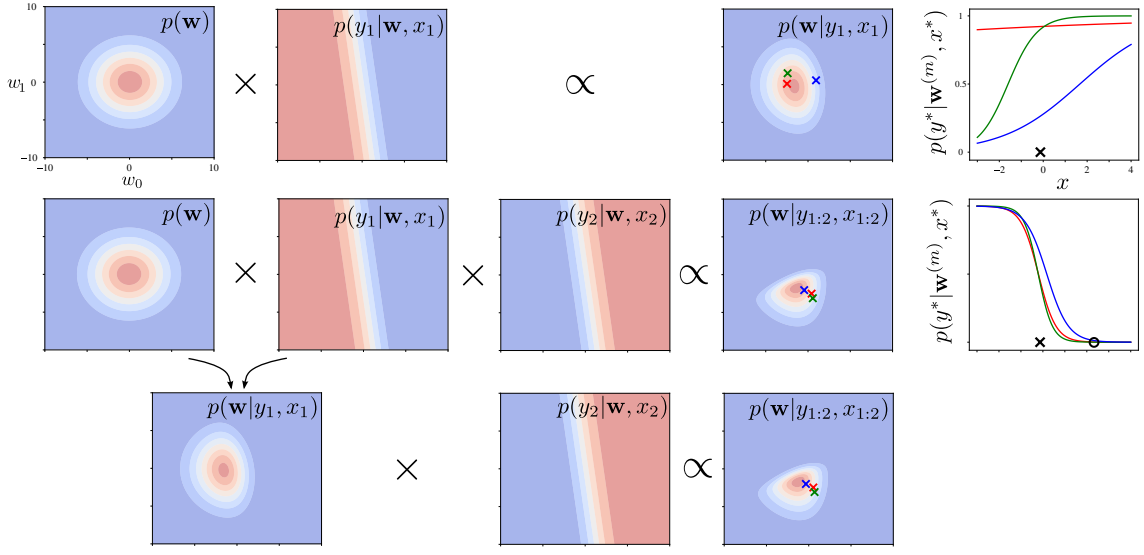
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example

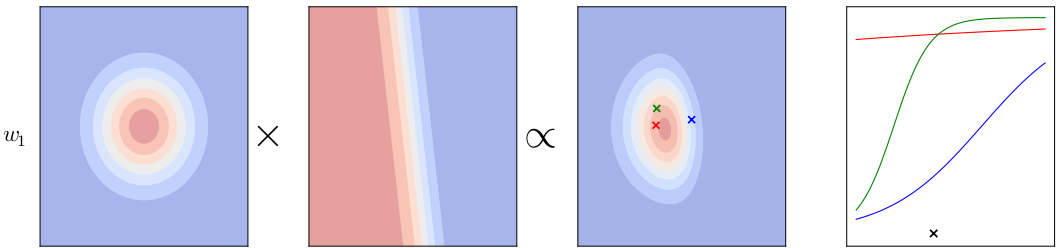


Bayesian Inference in Action: 1D Classification Example



Bayesian Inference in Action: 1D Classification Example

$p(\mathbf{w}|\mathbf{y}_{1:n-1}, \mathbf{x}_{1:n-1})$ $\quad$ $p(y_n|\mathbf{w}, x_n)$ $\quad$ $p(\mathbf{w}|\mathbf{y}_{1:n}, \mathbf{x}_{1:n})$ $\quad$ $p(y^*|\mathbf{w}^{(m)}, x^*)$

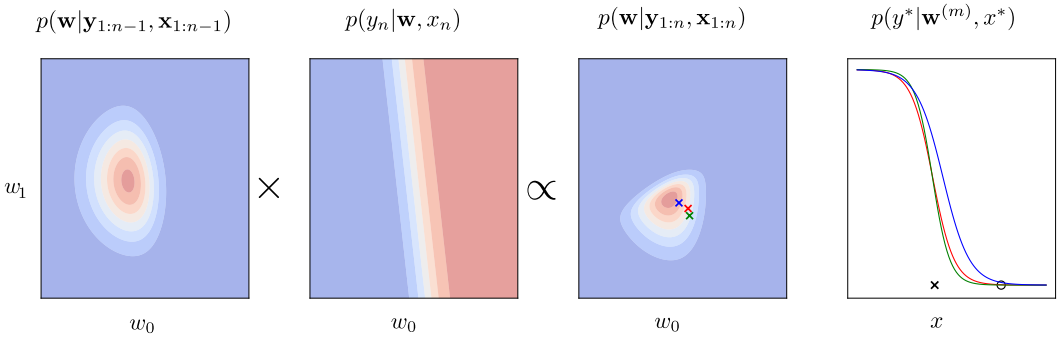


prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

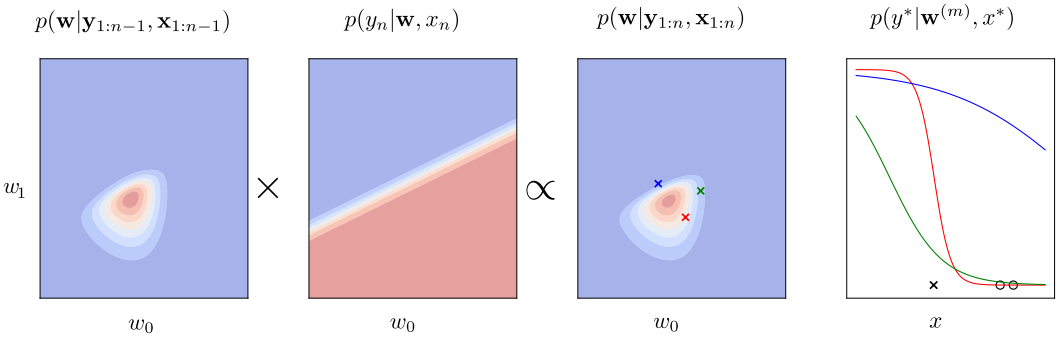
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

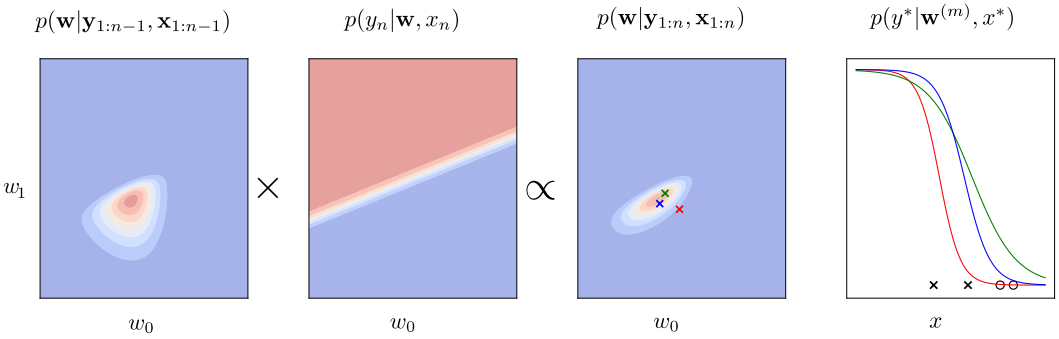
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

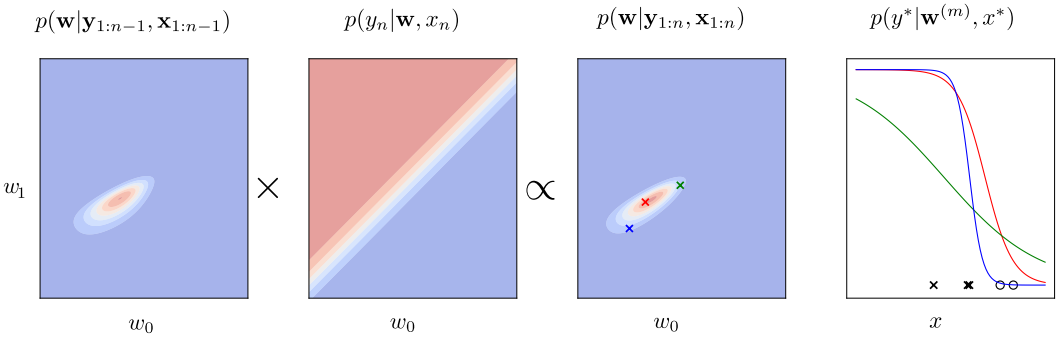
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

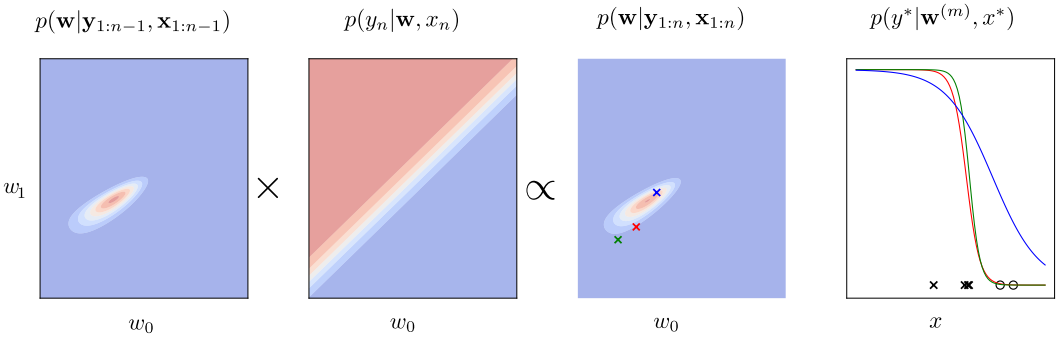
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

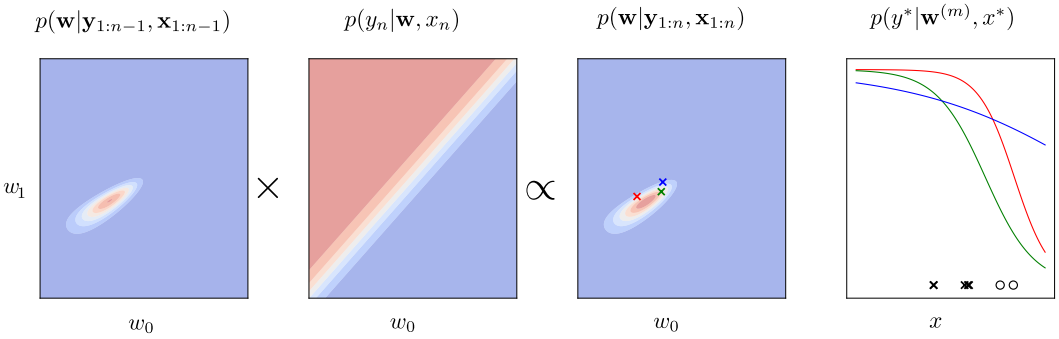
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

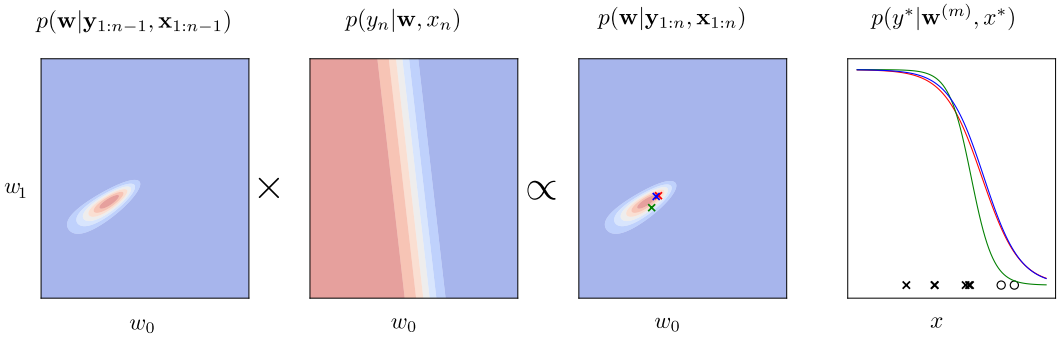
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

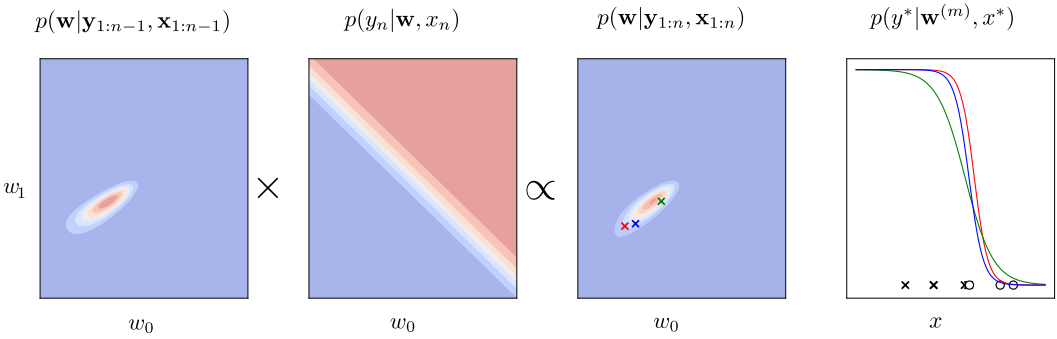
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

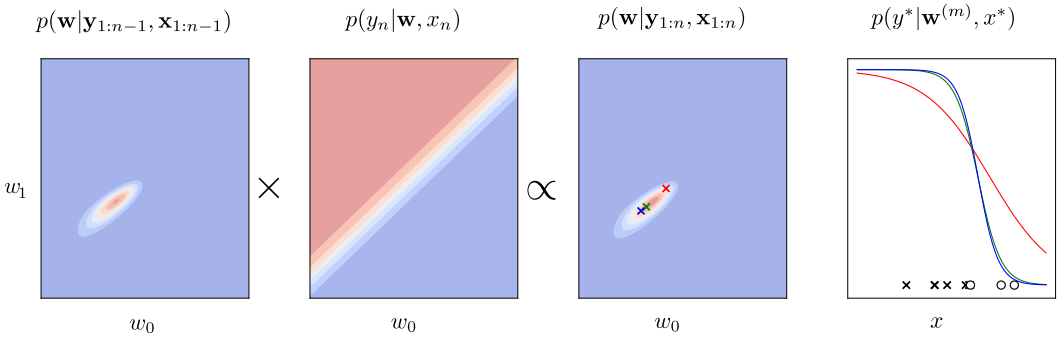
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

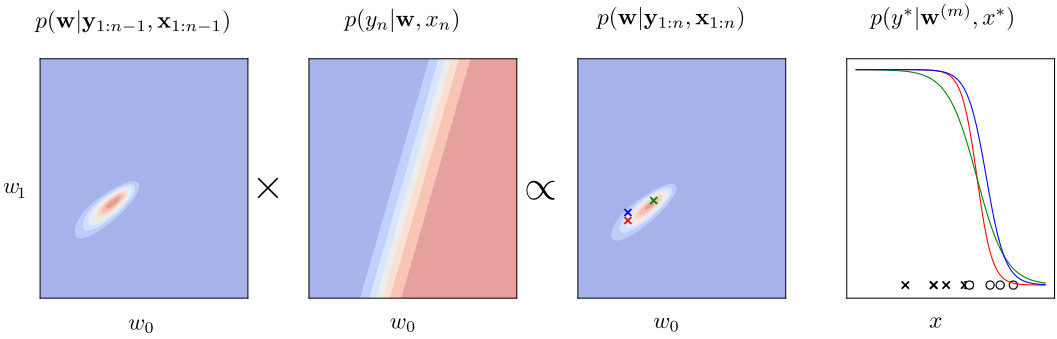
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

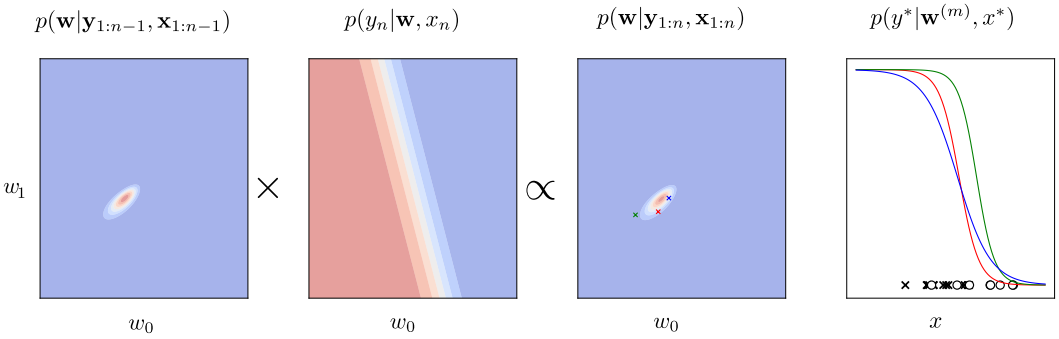
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

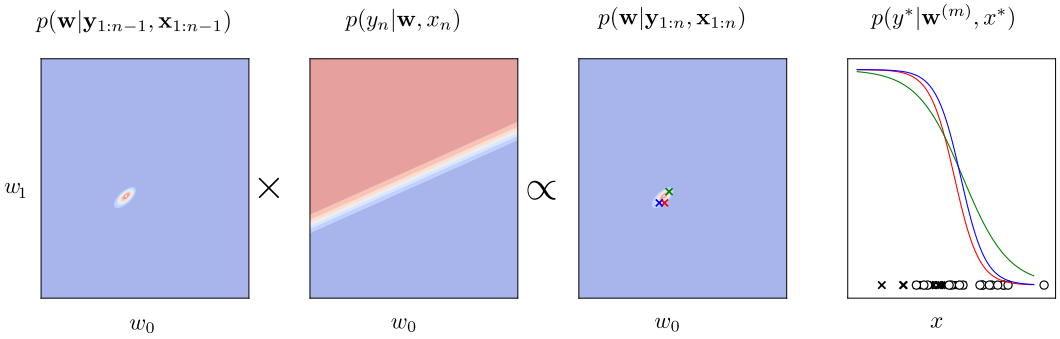
likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Bayesian Inference in Action: 1D Classification Example



prior

what we knew before seeing data

likelihood of $\mathbf{w}$

what the data told us

posterior

what we know after seeing data

Why be Bayesian (distributional estimates of weights)?

	maximum-likelihood	Bayes
learning	$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$	$p(\mathbf{w}\|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}\|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$
prediction	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) \approx p(y^*\|\mathbf{w}^{\text{ML}}, x^*)$	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) = \int p(\mathbf{w}\|\mathcal{Y}, \mathcal{X})\, p(y^*\|\mathbf{w}, x^*)\, \mathrm{d}\mathbf{w}$
	single weight setting	ensemble over weight settings
		requires approximation

Why be Bayesian (distributional estimates of weights)?

	maximum-likelihood	Bayes
learning	$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$	$p(\mathbf{w}\|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}\|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$
prediction	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) \approx p(y^*\|\mathbf{w}^{\text{ML}}, x^*)$	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) = \int p(\mathbf{w}\|\mathcal{Y}, \mathcal{X})\, p(y^*\|\mathbf{w}, x^*)\, \mathrm{d}\mathbf{w}$
	single weight setting	ensemble over weight settings
		requires approximation

Robust Deep Learning

point estimates over-confident, averaging over weight settings less so
Bayesian methods are more robust to adversarial examples (hard to fool ensemble of networks + uncertainty)

Why be Bayesian (distributional estimates of weights)?

	maximum-likelihood	Bayes
learning	$\mathbf{w}^{\mathsf{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$	$p(\mathbf{w}\|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}\|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$
prediction	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) \approx p(y^*\|\mathbf{w}^{\mathsf{ML}}, x^*)$	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) = \int p(\mathbf{w}\|\mathcal{Y}, \mathcal{X})\, p(y^*\|\mathbf{w}, x^*) \mathrm{d}\mathbf{w}$
	single weight setting	ensemble over weight settings
		requires approximation

Robust Deep Learning

point estimates over-confident, averaging over weight settings less so
Bayesian methods are more robust to adversarial examples (hard to fool ensemble of networks + uncertainty)

Data-efficient Deep Learning

small data, big model: build models 'the size of a house' & let data prune/learn structure
leverage heterogeneous data sources (multi-task learning) using shared parameters

Why be Bayesian (distributional estimates of weights)?

	maximum-likelihood	Bayes
learning	$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$	$p(\mathbf{w}\|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}\|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$
prediction	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) \approx p(y^*\|\mathbf{w}^{\text{ML}}, x^*)$	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) = \int p(\mathbf{w}\|\mathcal{Y}, \mathcal{X})\, p(y^*\|\mathbf{w}, x^*) \mathrm{d}\mathbf{w}$
	single weight setting	ensemble over weight settings
		requires approximation

Robust Deep Learning

point estimates over-confident, averaging over weight settings less so
Bayesian methods are more robust to adversarial examples (hard to fool ensemble of networks + uncertainty)

Data-efficient Deep Learning

small data, big model: build models 'the size of a house' & let data prune/learn structure
leverage heterogeneous data sources (multi-task learning) using shared parameters

Flexibile Deep Learning

continual learning: use old posterior as prior
active learning: select data that are expected to reduce uncertainty in parameter estimates the most

References: Introduction to the Bayesian approach

Excellent textbook introductions to Bayesian machine learning:

D. J. C. MacKay Information Theory, Inference and Learning Algorithms, 2003

C. Bishop Pattern Recognition and Machine Learning, 2006

K. Murphy Machine Learning: A Probabilistic Perspective, 2012

Beautiful seminal work on Bayesian neural networks:

R. M. Neal Bayesian Learning for Neural Networks, Lecture Notes in Statistics No. 118. New York: Springer-Verlag. 1996

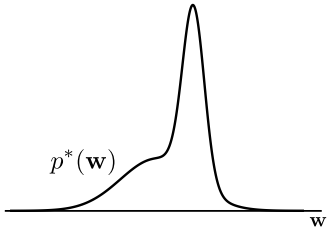
Approximate Bayesian Inference

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) \propto p(\mathcal{Y}, \mathbf{w}|\mathcal{X})$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

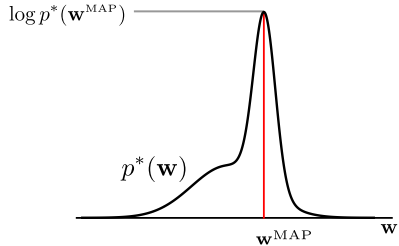
$$p(\mathbf{w}|\mathcal{Y}, \mathcal{X}) \propto p(\mathcal{Y}, \mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$



Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

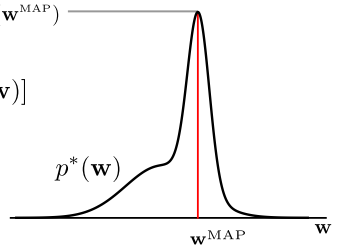
$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X})$$



Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

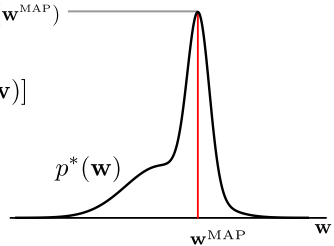
$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



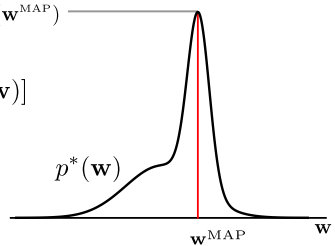
Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

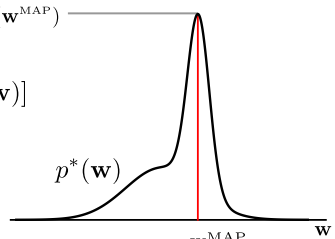
$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

what's the name for
this mathmatical object?

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

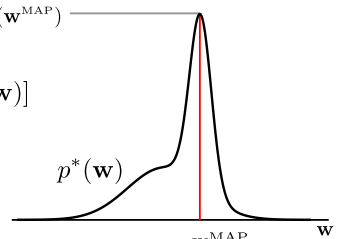
Hessian

what's the name for this mathmatical object?

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

Hessian

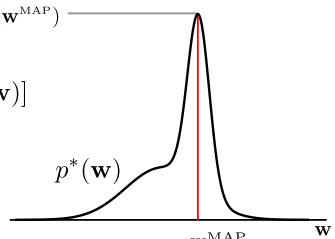
what's the numerical value of the first order term?

what's the name for this mathmatical object?

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

Hessian

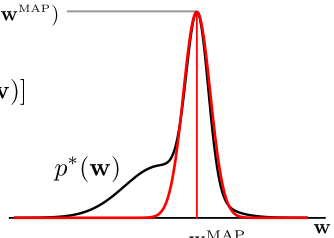
what's the numerical value of the first order term?

what's the name for this mathmatical object?

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

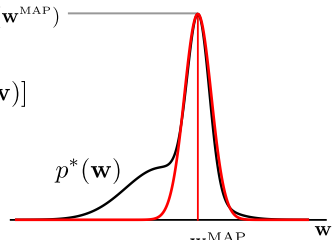
Hessian

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

Hessian

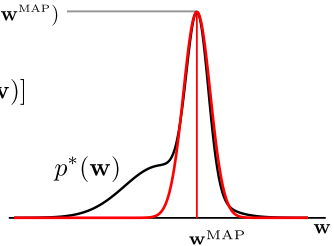
$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

what's the name of this type of function?

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

Hessian

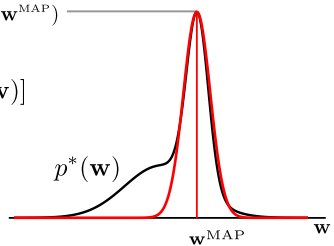
$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \det(2\pi\Sigma)^{1/2} \mathcal{G}(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma) \qquad \Sigma^{-1} = -\frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

Hessian

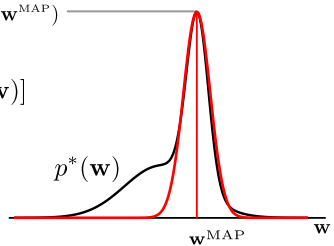
$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}\underbrace{\mathcal{G}(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma)}_{q(\mathbf{w})} \qquad \Sigma^{-1} = -\frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + \underbrace{(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}}_{0} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

Hessian

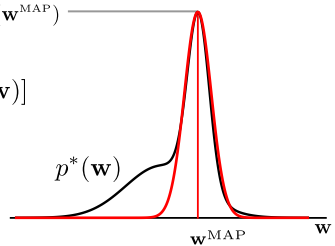
$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

$$p^*(\mathbf{w}) \approx \underbrace{p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}}_{\approx p(\mathcal{Y}|\mathcal{X})} \underbrace{\mathcal{G}(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma)}_{q(\mathbf{w})} \qquad \Sigma^{-1} = -\frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + (\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

0

Hessian

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

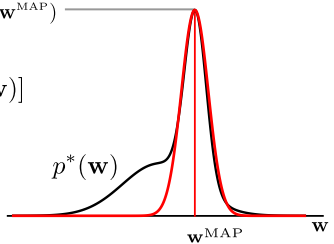
$$p^*(\mathbf{w}) \approx \underbrace{p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}}_{\approx p(\mathcal{Y}|\mathcal{X})} \underbrace{\mathcal{G}(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma)}_{q(\mathbf{w})} \qquad \Sigma^{-1} = -\frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

Prediction requires additional approx.: $p(y^*|\mathbf{x}^*,\mathcal{X},\mathcal{Y}) = \int p(y^*|\mathbf{w},\mathbf{x}^*)q(\mathbf{w})\mathrm{d}\mathbf{w}$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right]$$



Taylor expand log-prob to 2nd order about MAP est:

$$\log p^*(\mathbf{w}) \approx \log p^*(\mathbf{w}^{\text{MAP}}) + \underbrace{(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}}_{0} + \frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})$$

Hessian

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}^{\text{MAP}}) \exp\left(\frac{1}{2}(\mathbf{w}-\mathbf{w}^{\text{MAP}})^\top \frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}} (\mathbf{w}-\mathbf{w}^{\text{MAP}})\right)$$

$$p^*(\mathbf{w}) \approx \underbrace{p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}}_{\approx p(\mathcal{Y}|\mathcal{X})} \underbrace{\mathcal{G}(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma)}_{q(\mathbf{w})} \qquad \Sigma^{-1} = -\frac{\mathrm{d}^2\log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

simple Monte Carlo

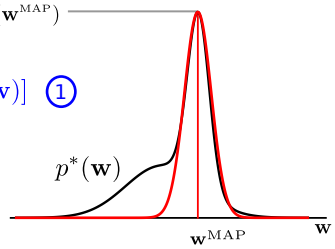
$$\mathbf{w}^{(m)} \sim q(\mathbf{w})$$

Prediction requires additional approx.: $p(y^*|\mathbf{x}^*,\mathcal{X},\mathcal{Y}) = \int p(y^*|\mathbf{w},\mathbf{x}^*)q(\mathbf{w})\mathrm{d}\mathbf{w} \approx \frac{1}{M}\sum_{m=1}^{M} p(y^*|\mathbf{w}^{(m)},\mathbf{x}^*)$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right] \quad (1)$$



1. optimise to find MAP solution

2. compute Hessian at optimum to form approximate posterior and marginal likelihood

3. predict using samples from approximate posterior and averaging resulting predictions

$$p^*(\mathbf{w}) \approx \underbrace{p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}}_{\approx p(\mathcal{Y}|\mathcal{X})} \underbrace{\mathcal{G}\left(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma\right)}_{q(\mathbf{w})}$$

(2)

$$\Sigma^{-1} = -\frac{\mathrm{d}^2 \log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

simple Monte Carlo

(3)

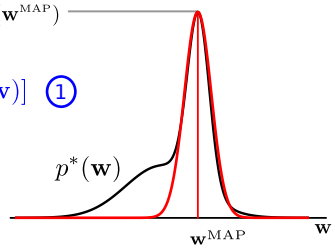
$$\mathbf{w}^{(m)} \sim q(\mathbf{w})$$

Prediction requires additional approx.: $p(y^*|\mathbf{x}^*,\mathcal{X},\mathcal{Y}) = \int p(y^*|\mathbf{w},\mathbf{x}^*)q(\mathbf{w})\mathrm{d}\mathbf{w} \approx \frac{1}{M}\sum_{m=1} p(y^*|\mathbf{w}^{(m)},\mathbf{x}^*)$

Laplace's approximation: MacKay 1991 (Saddle point approximation)

$$p(\mathbf{w}|\mathcal{Y},\mathcal{X}) \propto p(\mathcal{Y},\mathbf{w}|\mathcal{X}) = p^*(\mathbf{w})$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y},\mathcal{X}) = \arg\max_{\mathbf{w}} \left[\log p(\mathcal{Y}|\mathbf{w},\mathcal{X}) + \log p(\mathbf{w})\right] \quad (1)$$



1. optimise to find MAP solution

   might not converge → Hessian not positive definite

2. compute Hessian at optimum to form approximate posterior and marginal likelihood

   cubic cost in the number of parameters

3. predict using samples from approximate posterior and averaging resulting predictions

$$p^*(\mathbf{w}) \approx \underbrace{p^*(\mathbf{w}^{\text{MAP}})\det(2\pi\Sigma)^{1/2}}_{\approx p(\mathcal{Y}|\mathcal{X})} \underbrace{\mathcal{G}\left(\mathbf{w};\mathbf{w}^{\text{MAP}},\Sigma\right)}_{q(\mathbf{w})}$$

(2)

$$\Sigma^{-1} = -\frac{\mathrm{d}^2 \log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\text{MAP}}}$$

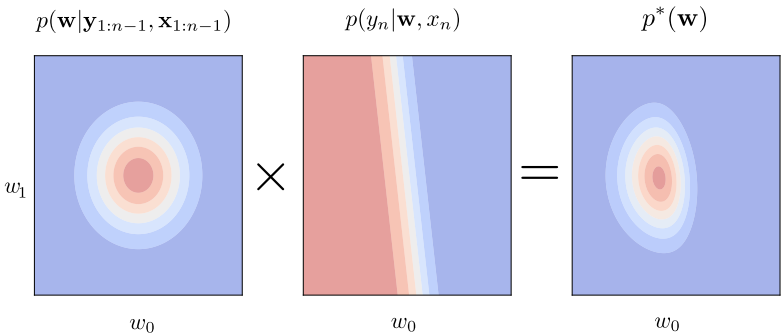
simple Monte Carlo

$$\mathbf{w}^{(m)} \sim q(\mathbf{w})$$

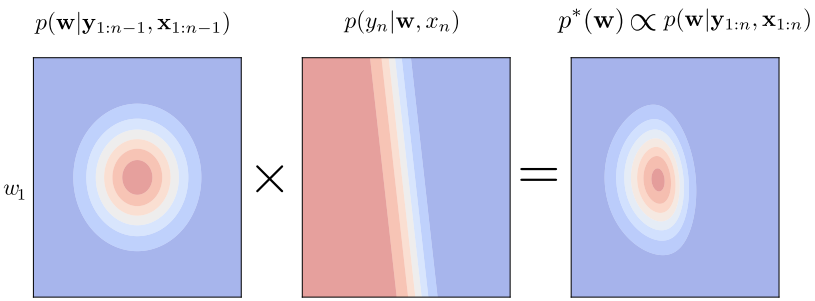
(3)

Prediction requires additional approx.: $p(y^*|\mathbf{x}^*,\mathcal{X},\mathcal{Y}) = \int p(y^*|\mathbf{w},\mathbf{x}^*)q(\mathbf{w})\mathrm{d}\mathbf{w} \approx \frac{1}{M}\sum_{m=1} p(y^*|\mathbf{w}^{(m)},\mathbf{x}^*)$

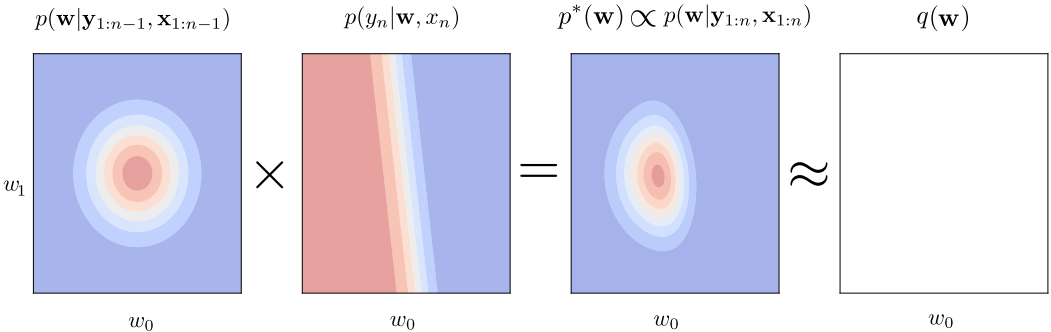
Laplace's approximation: Classification Example



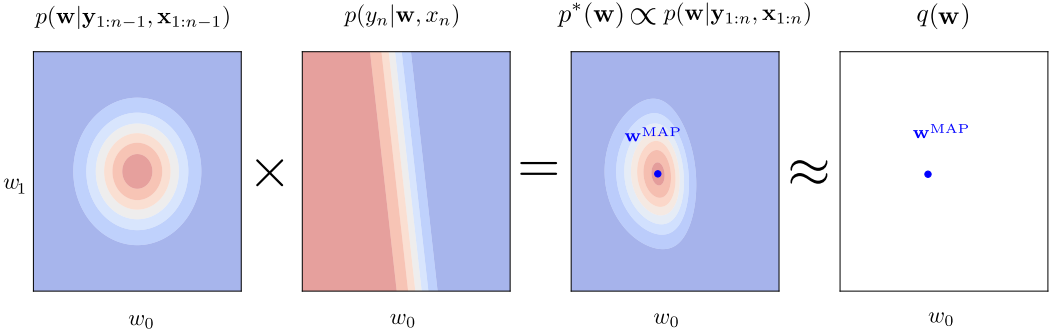
Laplace's approximation: Classification Example



Laplace's approximation: Classification Example



Laplace's approximation: Classification Example



prior
what we knew before seeing data

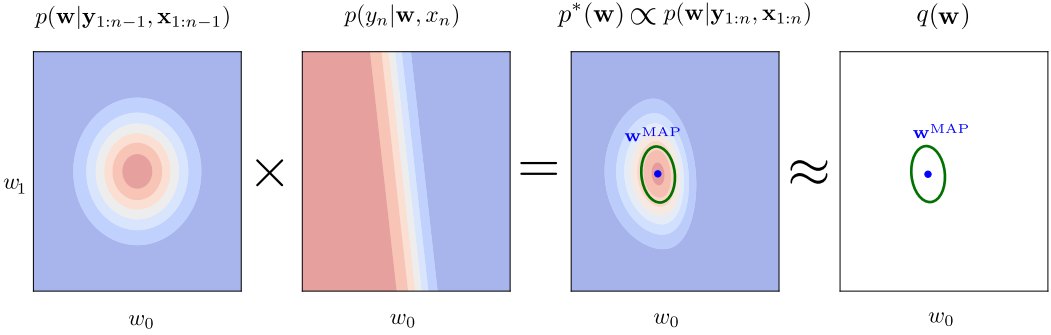
likelihood of $\mathbf{w}$
what the data told us

$\propto$ posterior
what we know after seeing data

Laplace approx.

1. $\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

Laplace's approximation: Classification Example



prior
what we knew before seeing data

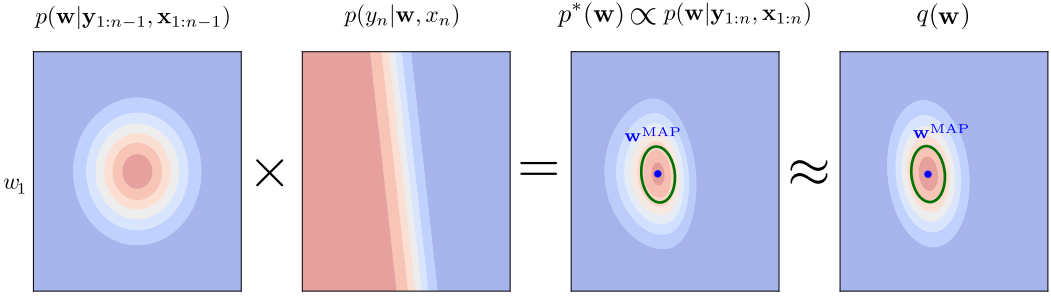
likelihood of $\mathbf{w}$
what the data told us

$\propto$ posterior
what we know after seeing data

Laplace approx.

1. $\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$
2. $\Sigma^{-1} = -\frac{\mathrm{d}^2 \log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\mathrm{MAP}}}$

Laplace's approximation: Classification Example



prior
what we knew
before seeing data

likelihood of $\mathbf{w}$
what the
data told us

$\propto$ posterior
what we know
after seeing data

Laplace approx.
$\mathcal{G}(\mathbf{w}; \mathbf{w}^{\mathrm{MAP}}, \Sigma)$

1. $\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$
2. $\Sigma^{-1} = -\frac{\mathrm{d}^2 \log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\top}\Big|_{\mathbf{w}=\mathbf{w}^{\mathrm{MAP}}}$

References: Laplace's method for neural networks

D. J. C. MacKay A practical Bayesian framework for backpropagation networks, Neural Computation 4(3) 448-472, 1992

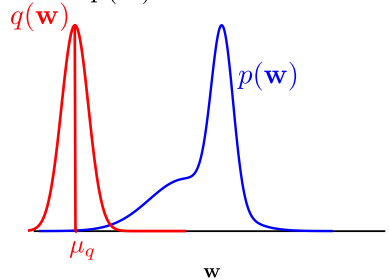
D. J. C. MacKay Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks , Network: Computation in Neural Systems 6(3), 469-505, 1995

H. Ritter et al. A Scalable Laplace Approximation for Neural Networks, ICLR, 2018

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence

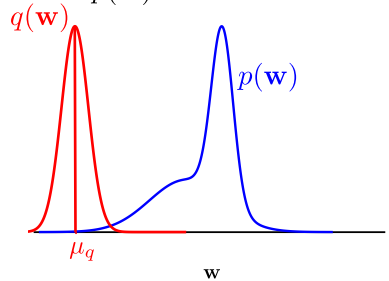
$$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$



Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative



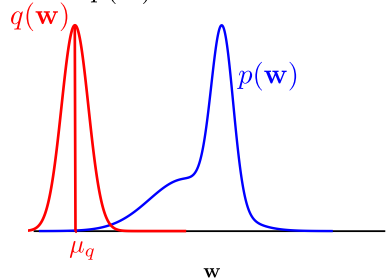
Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence

$$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}))$$



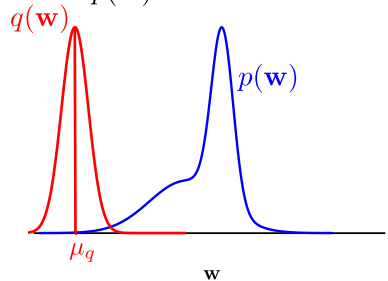
Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence

$$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})}$$

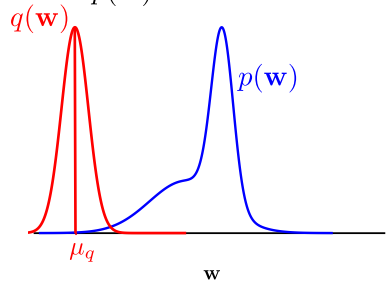


Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$



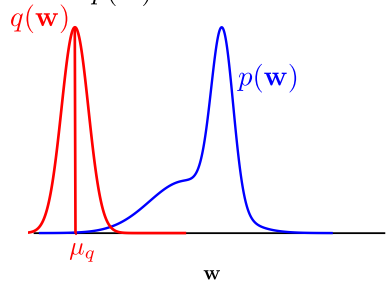
Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$



Variational Inference: the KL Divergence

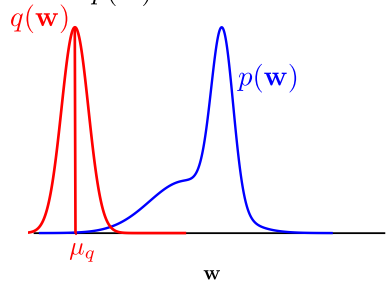
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w}))$$



Variational Inference: the KL Divergence

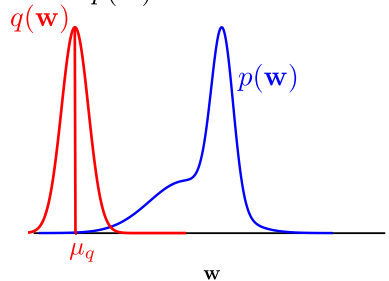
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$



Variational Inference: the KL Divergence

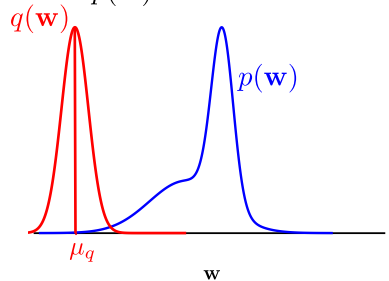
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$



Variational Inference: the KL Divergence

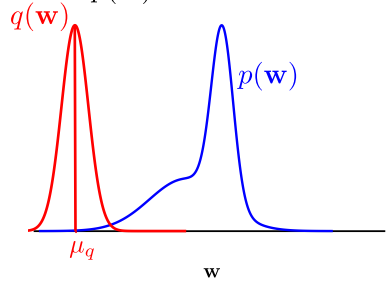
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$



Variational Inference: the KL Divergence

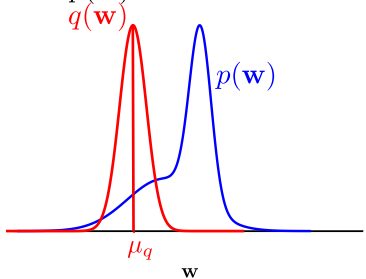
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

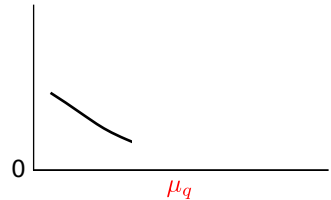
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

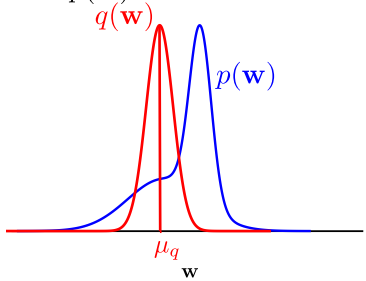
Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

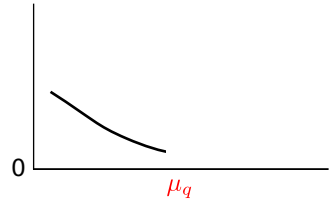
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

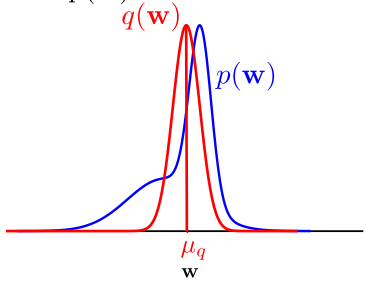
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$



Variational Inference: the KL Divergence

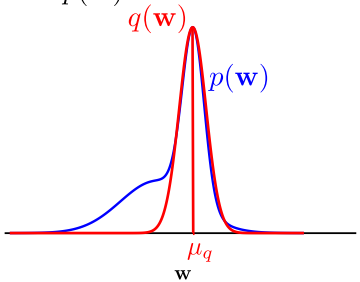
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

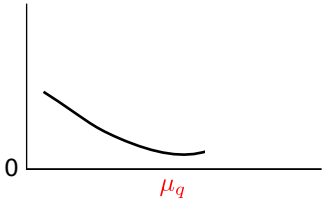
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

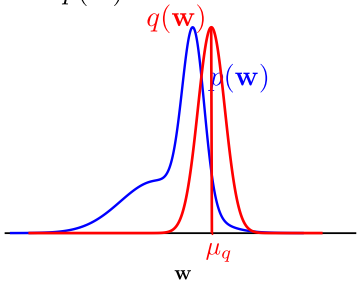
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

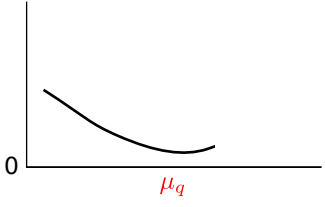
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

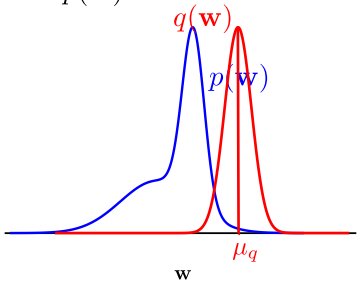
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

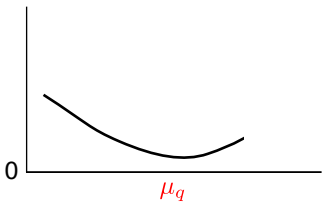
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

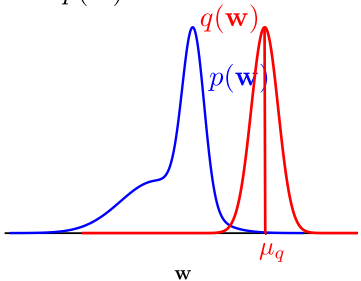
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$



Variational Inference: the KL Divergence

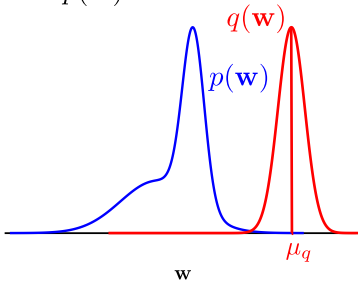
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

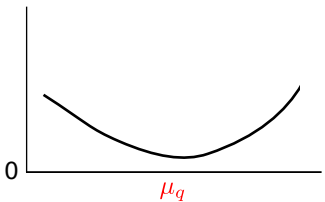
1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$





Variational Inference: the KL Divergence

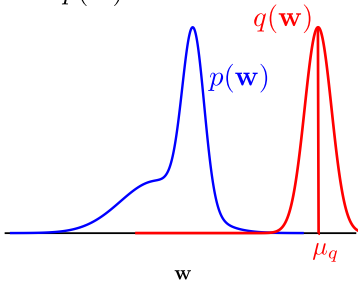
Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$



Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right)$$

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w}$$

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

Variational Inference: the KL Divergence

Kullback–Leibler (KL) divergence $$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$$

1. non-negative

$$\frac{\delta^2}{\delta q^2} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

divergence measures 'distance' between distributions

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} \overset{1\&2}{\geq} 0$ equality when $q(\mathbf{w}) = p(\mathbf{w})$

Use KL to optimise an approximation $q(\mathbf{w})$ to the true posterior

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} \overset{1\&2}{\geq} 0$ equality when $q(\mathbf{w}) = p(\mathbf{w})$

Use KL to optimise an approximation $q(\mathbf{w})$ to the true posterior

$$q^{\mathrm{VI}}(\mathbf{w}) \overset{1\&2}{=} \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} \overset{1\&2}{\geq} 0$ equality when $q(\mathbf{w}) = p(\mathbf{w})$

Use KL to optimise an approximation $q(\mathbf{w})$ to the true posterior

$$q^{\mathrm{VI}}(\mathbf{w}) \overset{1\&2}{=} \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathcal{Y}, \mathcal{X})) \overset{3}{=} \arg\min_{q \in \mathcal{Q}} \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w}) p(\mathcal{Y}|\mathbf{w}, \mathcal{X})} \mathrm{d}\mathbf{w}$$

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z}p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z}p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} \overset{1\&2}{\geq} 0$ equality when $q(\mathbf{w}) = p(\mathbf{w})$

Use KL to optimise an approximation $q(\mathbf{w})$ to the true posterior

$$q^{\mathrm{VI}}(\mathbf{w}) \overset{1\&2}{=} \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathcal{Y}, \mathcal{X})) \overset{3}{=} \arg\min_{q \in \mathcal{Q}} \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w}) p(\mathcal{Y}|\mathbf{w}, \mathcal{X})} \mathrm{d}\mathbf{w}$$

$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})} \qquad \mathbf{w}^{(m)} \sim q(\mathbf{w})$$

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z} p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z} p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w}) Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} \overset{1\&2}{\geq} 0$ equality when $q(\mathbf{w}) = p(\mathbf{w})$

Use KL to optimise an approximation $q(\mathbf{w})$ to the true posterior

$$q^{\mathrm{VI}}(\mathbf{w}) \overset{1\&2}{=} \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathcal{Y}, \mathcal{X})) \overset{3}{=} \arg\min_{q \in \mathcal{Q}} \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w}) p(\mathcal{Y}|\mathbf{w}, \mathcal{X})} \mathrm{d}\mathbf{w}$$

$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})} \qquad \mathbf{w}^{(m)} \sim q(\mathbf{w})$$

requires reparameterisation trick

3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

$$\mathrm{KL}\left(q(\mathbf{w})||\frac{1}{Z} p^*(\mathbf{w})\right) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{\frac{1}{Z} p^*(\mathbf{w})} \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w}) Z}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w}$$

$$= \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \int q(\mathbf{w}) \log Z \mathrm{d}\mathbf{w} = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p^*(\mathbf{w})} \mathrm{d}\mathbf{w} + \log Z$$

→ suitable for approximate inference

Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

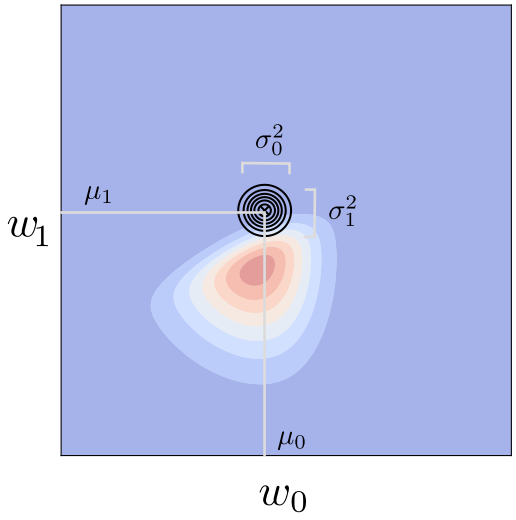
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

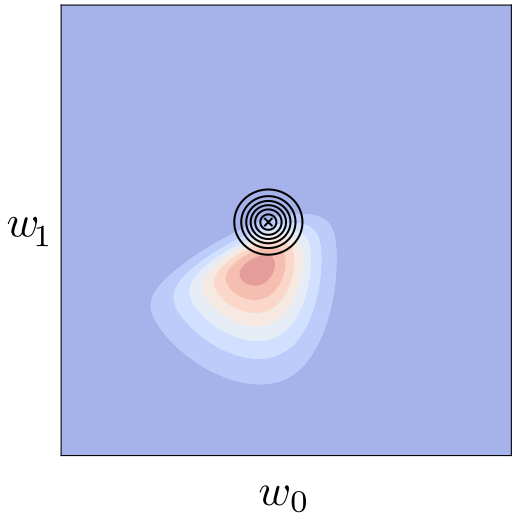
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

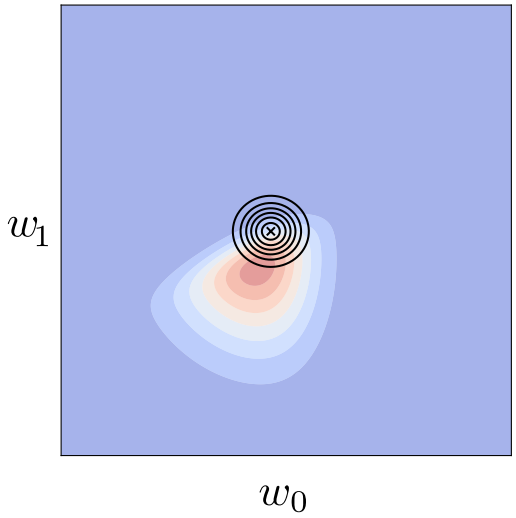
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

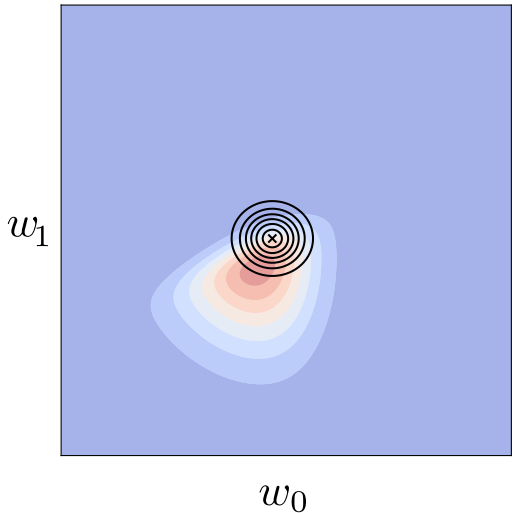
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

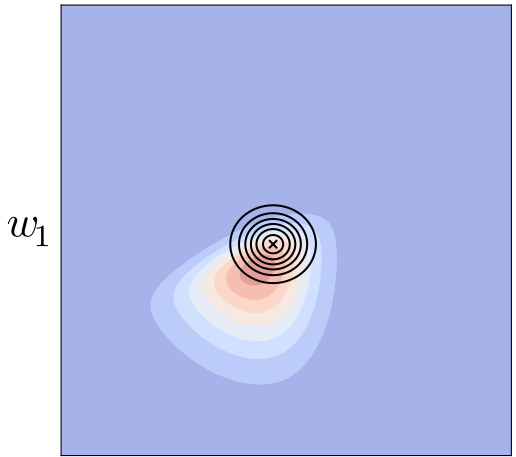
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

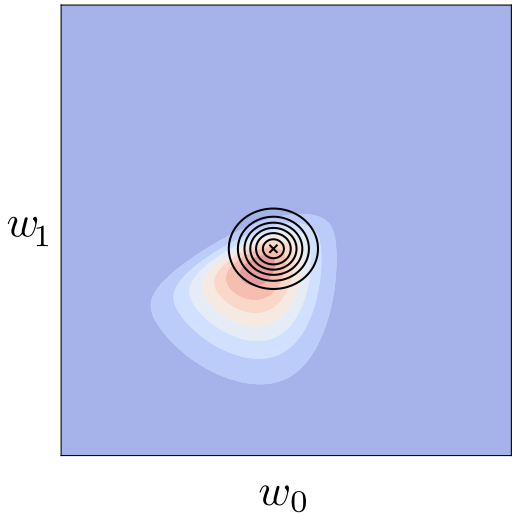
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

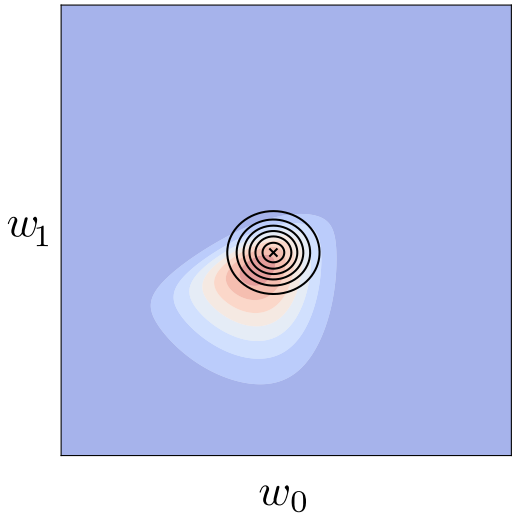
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

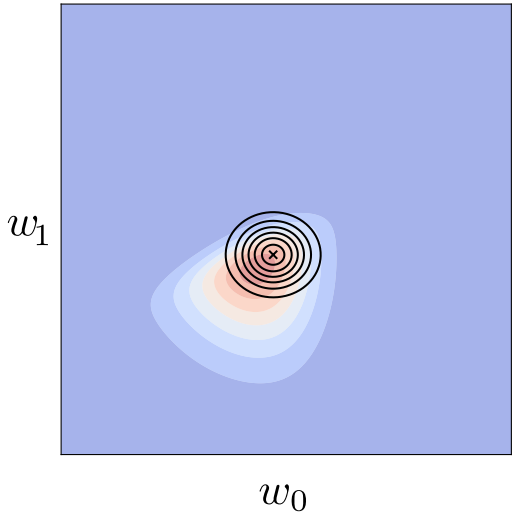
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

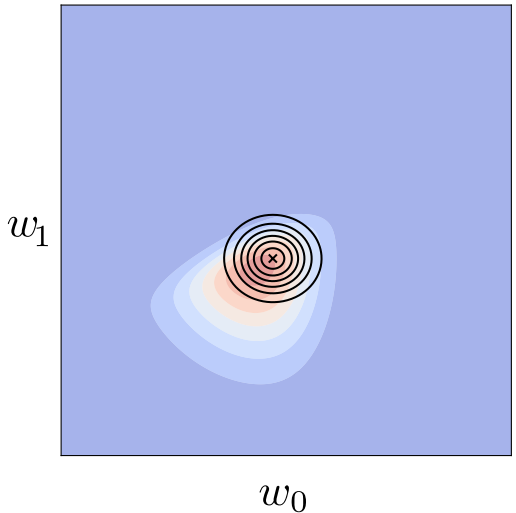
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

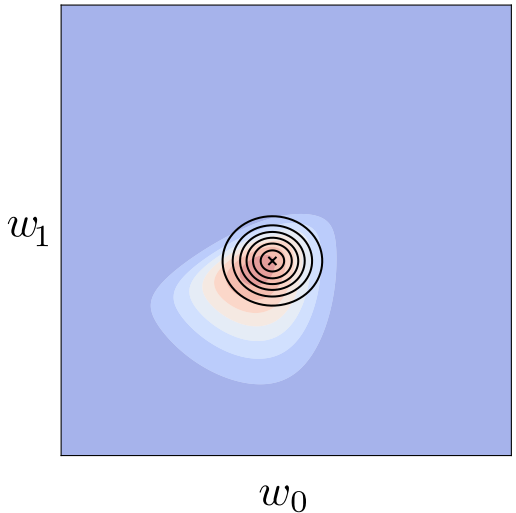
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

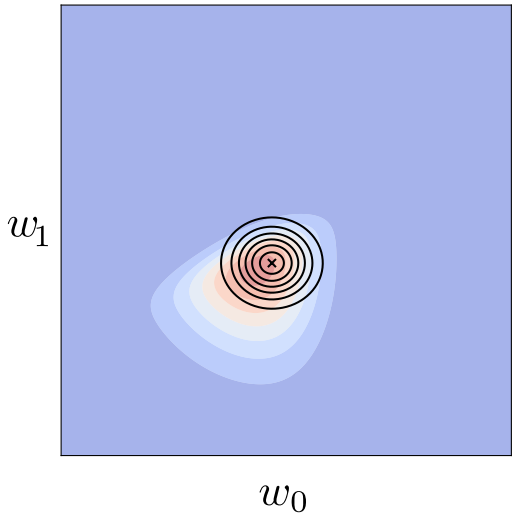
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

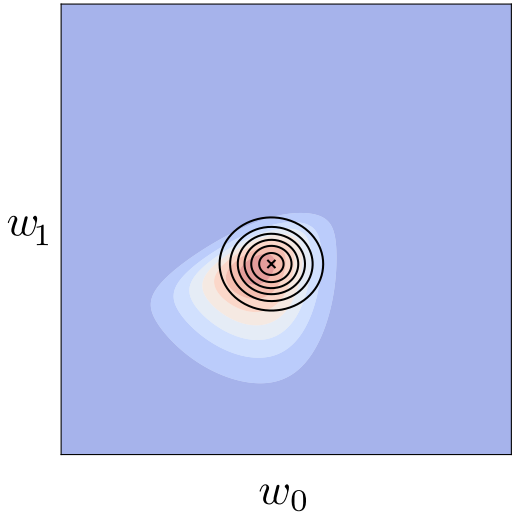
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

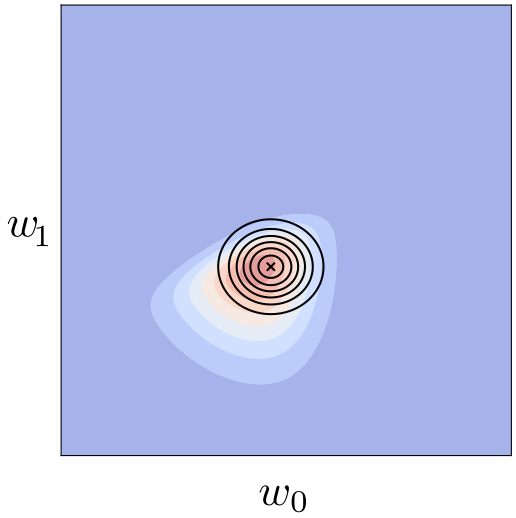
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

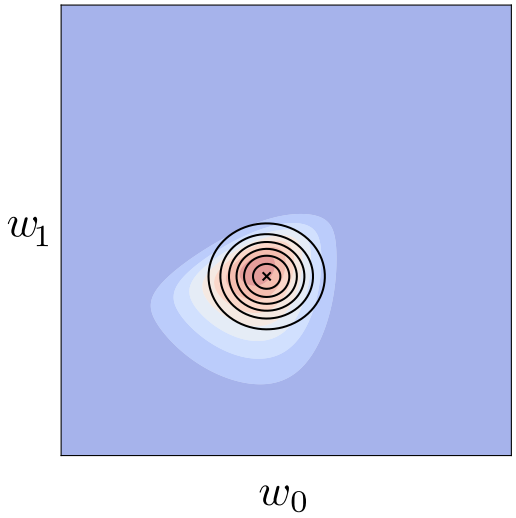
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

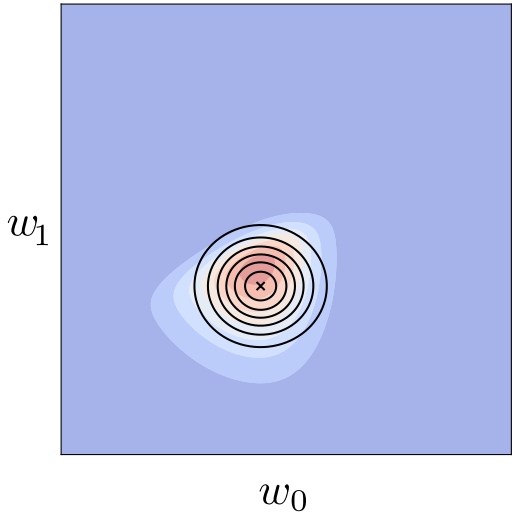
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

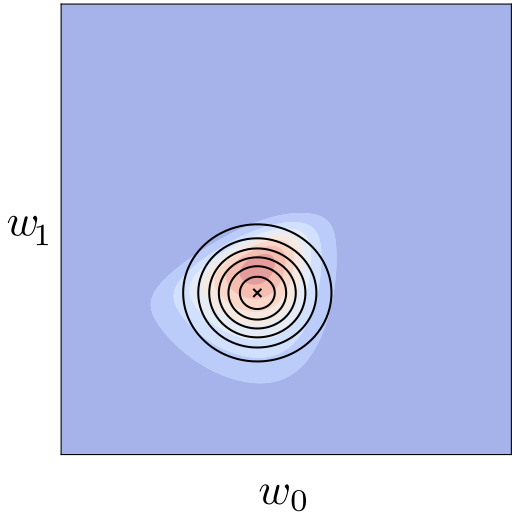
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization



Variational Inference: Classification Example

$$q^{\mathrm{VI}}(\mathbf{w}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X}))$$

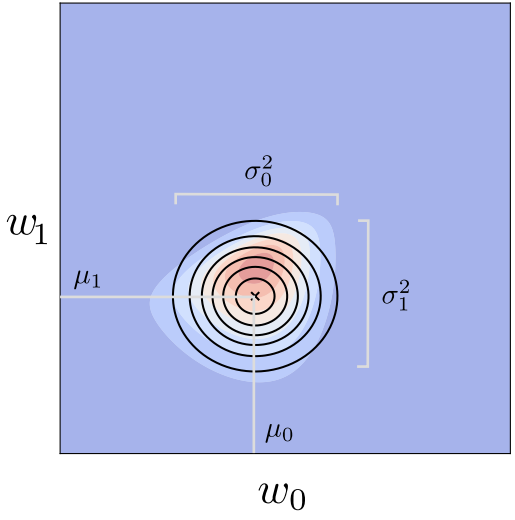
$$\approx \arg\min_{q \in \mathcal{Q}} \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(\mathbf{w}^{(m)})}{p(\mathbf{w}^{(m)}) p(\mathcal{Y}|\mathbf{w}^{(m)}, \mathcal{X})}$$

approximate family:
factorised Gaussian

$$q(\mathbf{w}) = \mathcal{G}(w_0; \mu_0, \sigma_0^2) \times \mathcal{G}(w_1; \mu_1, \sigma_1^2)$$

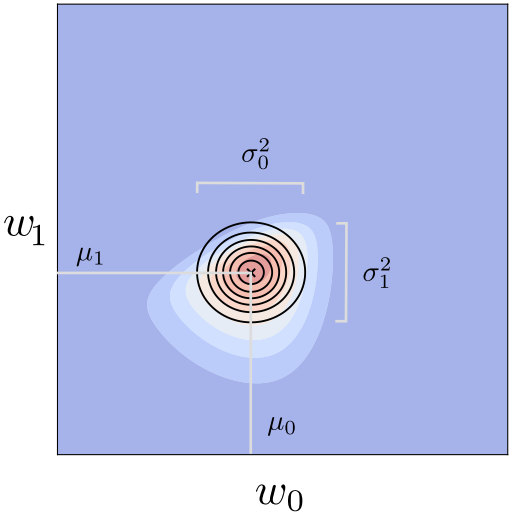
optimise w.r.t. $\mu_0, \sigma_0^2, \mu_1, \sigma_1^2$

VI converts inference into optimization

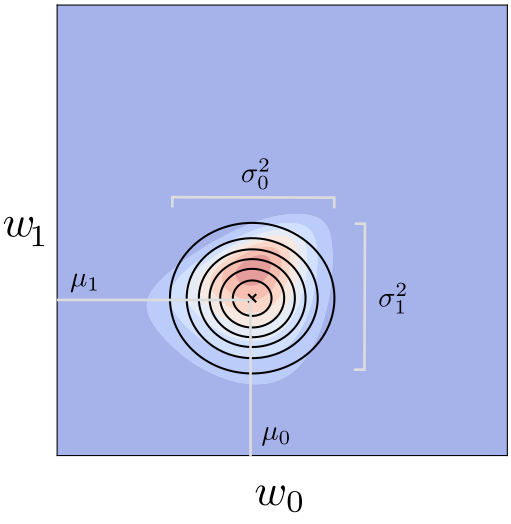


Variational Inference vs Laplace: Classification Example

Laplace Approximation (diagonal Hessian)



Variational Inference (factorised)



References: Variational inference for neural networks

G. Hinton and D. Van Camp Keeping the neural networks simple by minimizing the description length of the weights, COLT 5-13, ACM, 1993

D. Barber and C. Bishop Ensemble Learning in Bayesian Neural Networks, Neural Networks and Machine Learning 1998

A. Graves Practical Variational Inference for Neural Networks, NIPS 2011

C. Blundell et al. Weight Uncertainty in Neural Networks, ICML, 2015

MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$

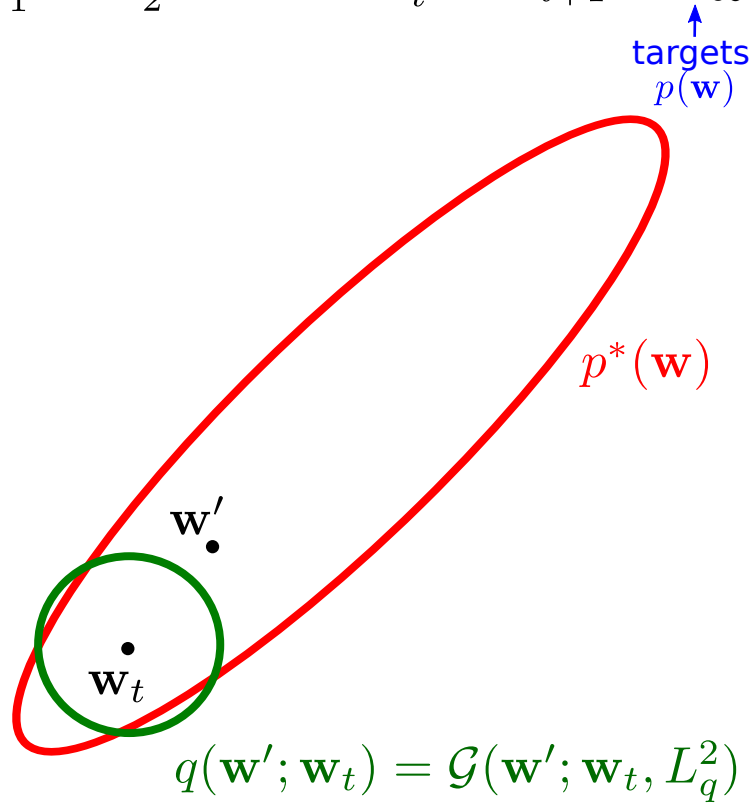
targets $p(\mathbf{w})$

MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$ targets $p(\mathbf{w})$

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

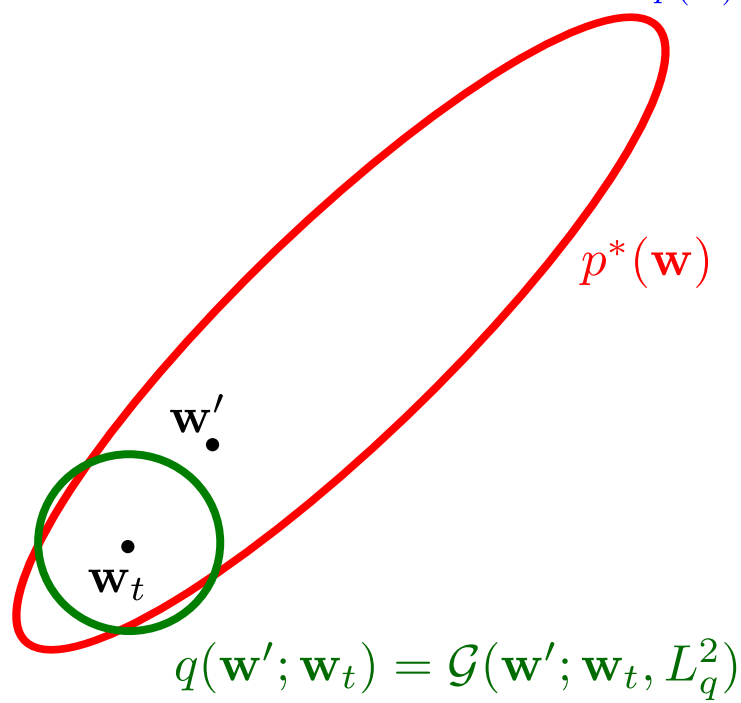
accept if $a \geq 1 \quad \mathbf{w}_{t+1} = \mathbf{w}'$

else

accept with prob $a \quad \mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

endif



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$ targets $p(\mathbf{w})$

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

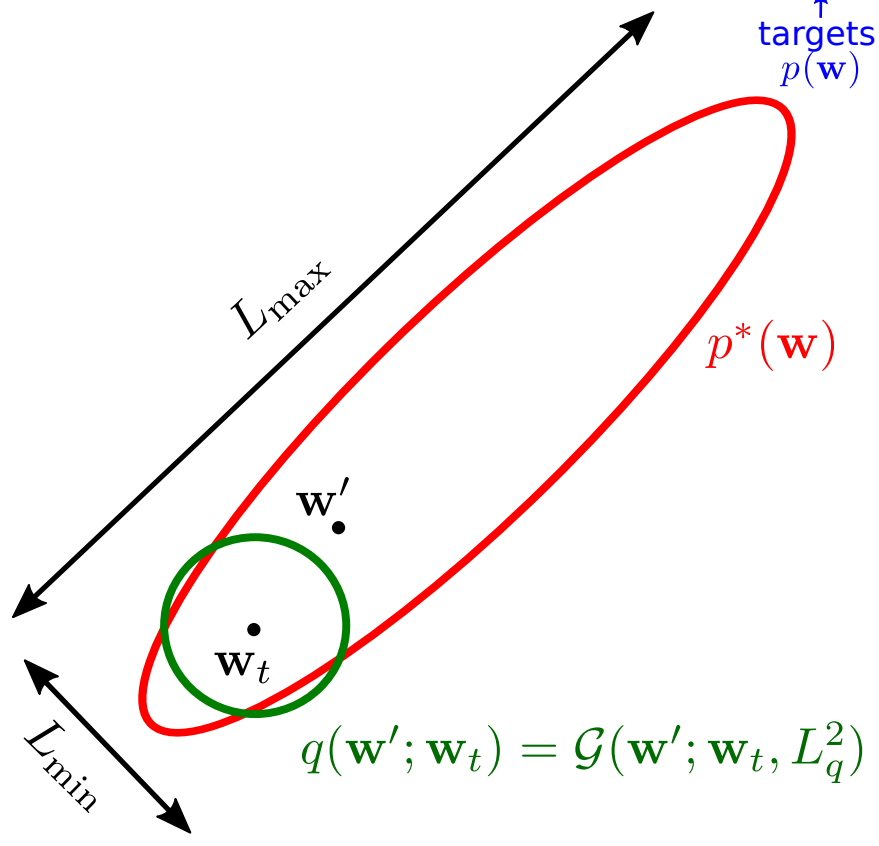
accept if $a \geq 1 \quad \mathbf{w}_{t+1} = \mathbf{w}'$

else

accept with prob $a \quad \mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

endif



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$

reasonable acceptance: $L_q \approx L_{\min}$

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

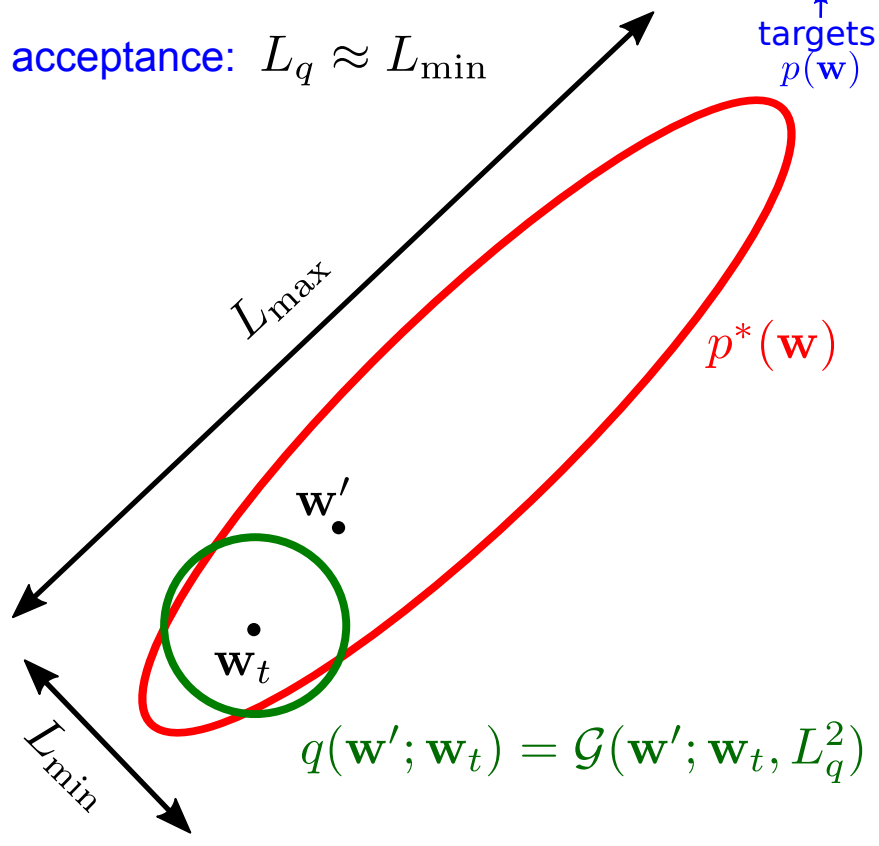
accept if $a \geq 1$ $\mathbf{w}_{t+1} = \mathbf{w}'$

else

accept with prob a $\mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

endif



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$ (targets $p(\mathbf{w})$)

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

accept if $a \geq 1$ $\mathbf{w}_{t+1} = \mathbf{w}'$

else

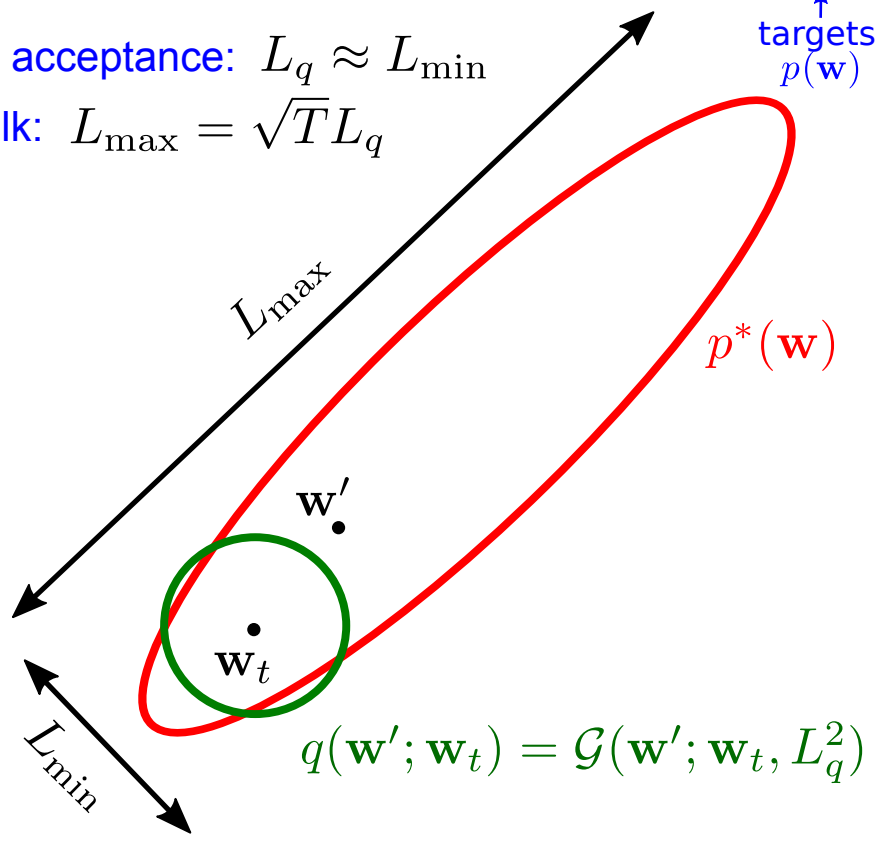
accept with prob a $\mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

endif

reasonable acceptance: $L_q \approx L_{\min}$

random walk: $L_{\max} = \sqrt{T} L_q$



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$ (targets $p(\mathbf{w})$)

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

accept if $a \geq 1$ $\mathbf{w}_{t+1} = \mathbf{w}'$

else

accept with prob a $\mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

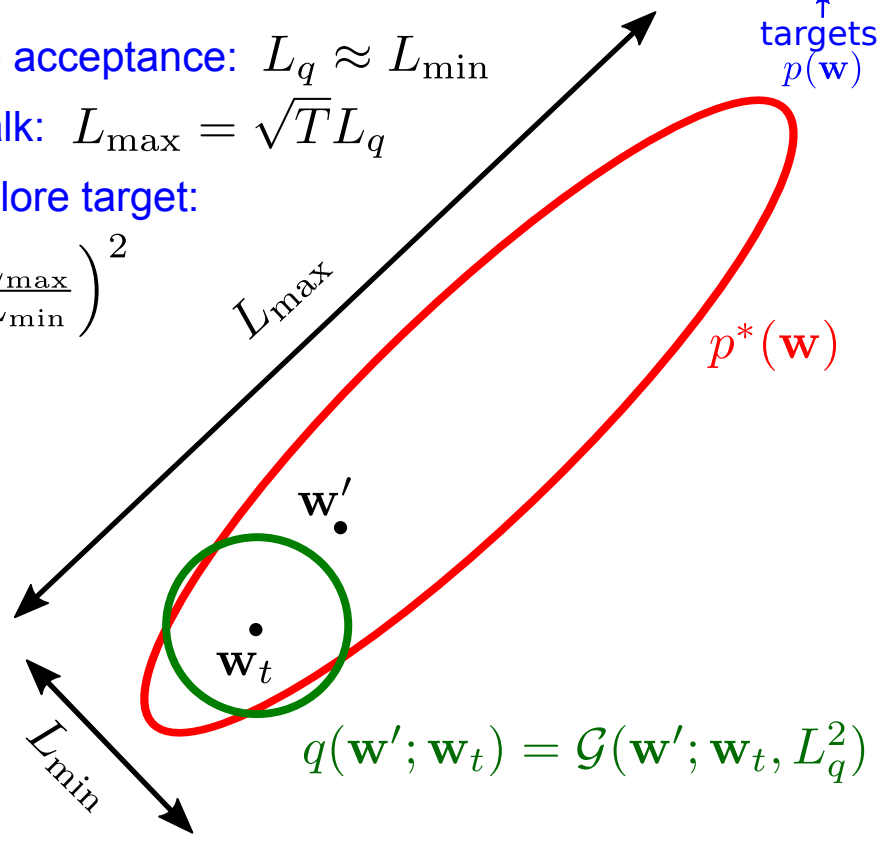
endif

reasonable acceptance: $L_q \approx L_{\min}$

random walk: $L_{\max} = \sqrt{T} L_q$

time to explore target:

$$T = \left(\frac{L_{\max}}{L_{\min}} \right)^2$$



MCMC for neural networks: Would vanilla Metropolis Hastings work?

samples from posterior can be used for prediction etc. via Monte Carlo

$$\mathbf{w}_t \sim \frac{1}{Z} p^*(\mathbf{w}) \qquad \int f(\mathbf{w}) \frac{1}{Z} p^*(\mathbf{w}) \approx \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{w}_t) \qquad f(\mathbf{w}_t) = p(y|\mathbf{z}, \mathbf{w}_t)$$

idea: produce samples from Markov Chain: $\mathbf{w}_1 \xrightarrow{\mathsf{T}} \mathbf{w}_2 \xrightarrow{\mathsf{T}} \cdots \xrightarrow{\mathsf{T}} \mathbf{w}_t \xrightarrow{\mathsf{T}} \mathbf{w}_{t+1} \cdots \mathbf{w}_\infty$ targets $p(\mathbf{w})$

Metropolis-Hastings

propose $\mathbf{w}' \sim q(\mathbf{w}'; \mathbf{w}_t)$

compute acceptance ratio $a = \dfrac{p^*(\mathbf{w}')q(\mathbf{w}_t; \mathbf{w}')}{p^*(\mathbf{w}_t)q(\mathbf{w}'; \mathbf{w}_t)}$

accept if $a \geq 1$ $\mathbf{w}_{t+1} = \mathbf{w}'$

else

accept with prob a $\mathbf{w}_{t+1} = \mathbf{w}'$

reject otherwise $\mathbf{w}_{t+1} = \mathbf{w}_t$

endif

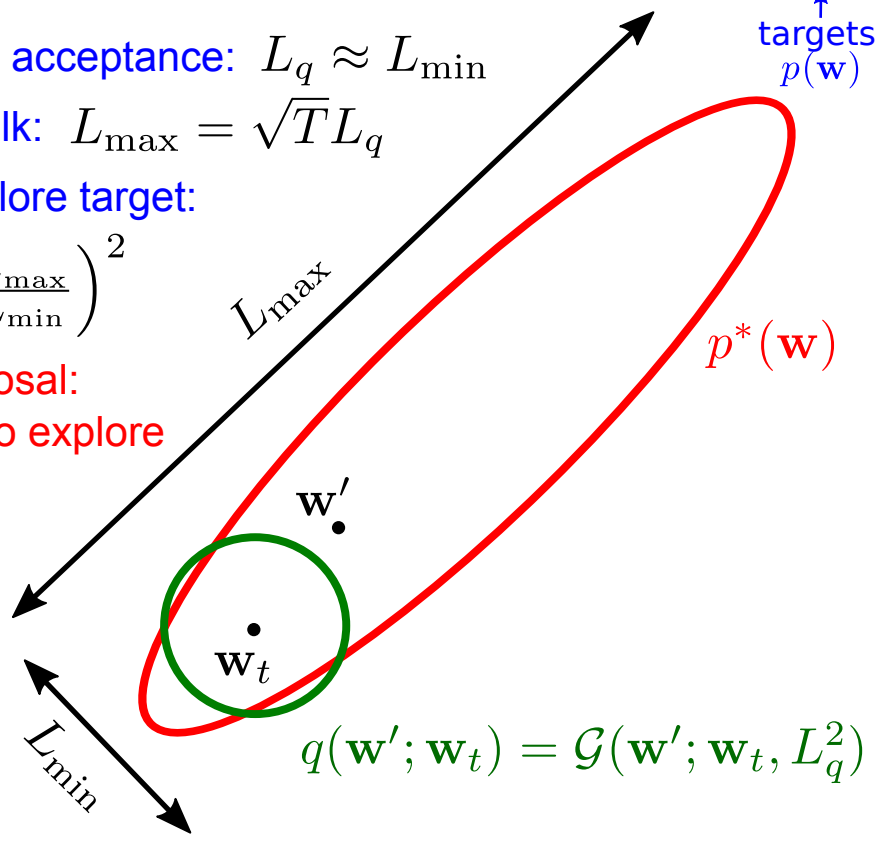
reasonable acceptance: $L_q \approx L_{\min}$

random walk: $L_{\max} = \sqrt{T} L_q$

time to explore target:

$$T = \left(\frac{L_{\max}}{L_{\min}} \right)^2$$

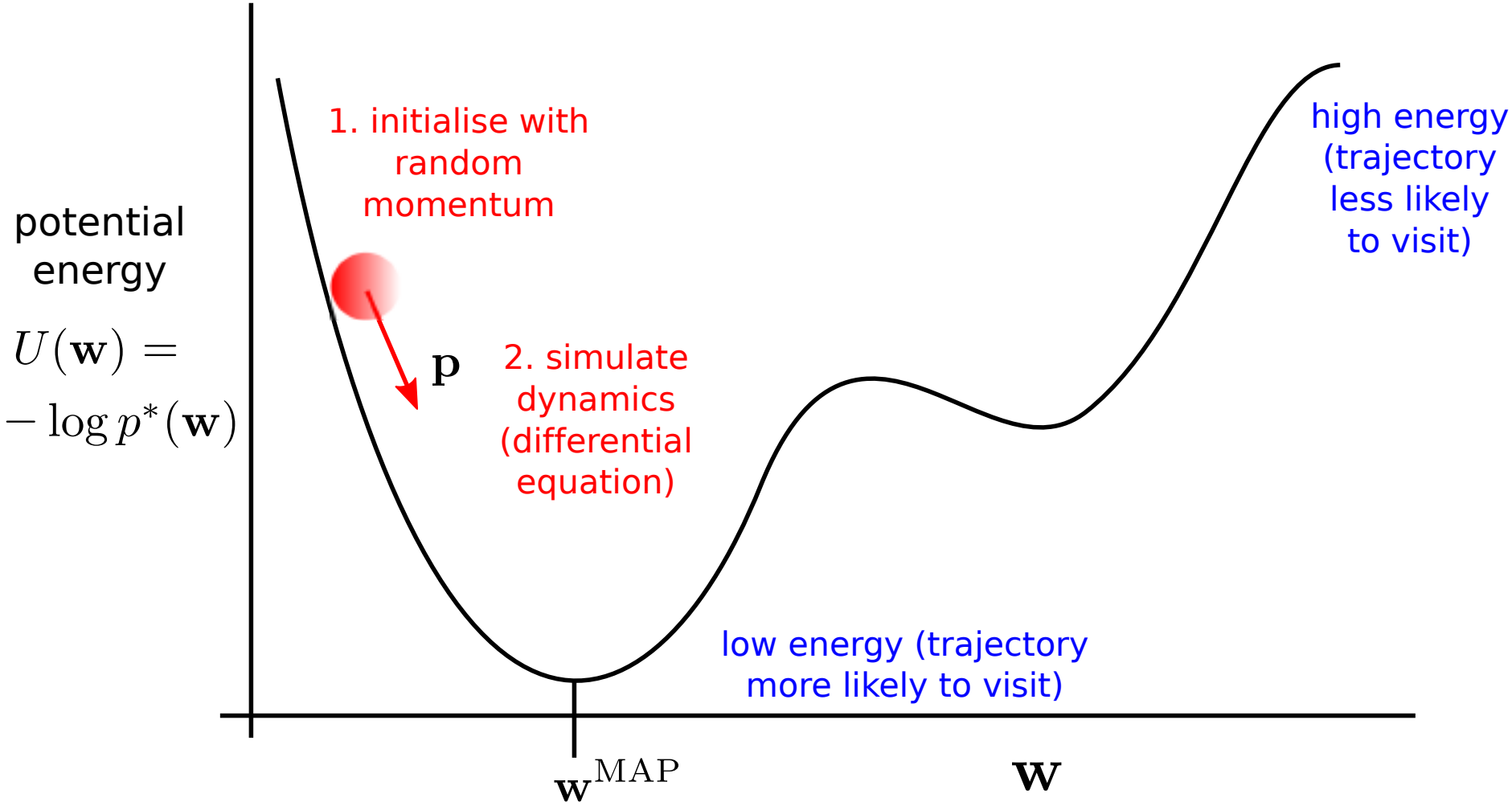
naive proposal:
very slow to explore



Hamiltonian Monte Carlo: Intuition

idea: use gradient information of log-target to shape MH proposal

construct physical system whose dynamics expore weight-space



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$- \log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$- \log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$

Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

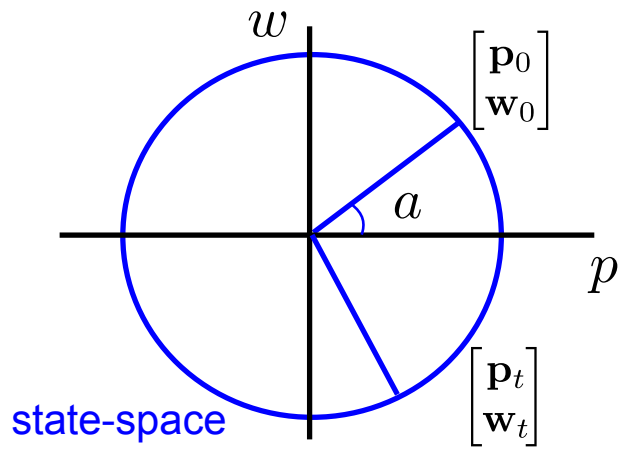
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$- \log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

$$\begin{bmatrix} \mathbf{p}_t \\ \mathbf{w}_t \end{bmatrix} = \mathrm{T}_t\left(\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{w}_0 \end{bmatrix}\right)$$

final state

dynamical mapping

initial state

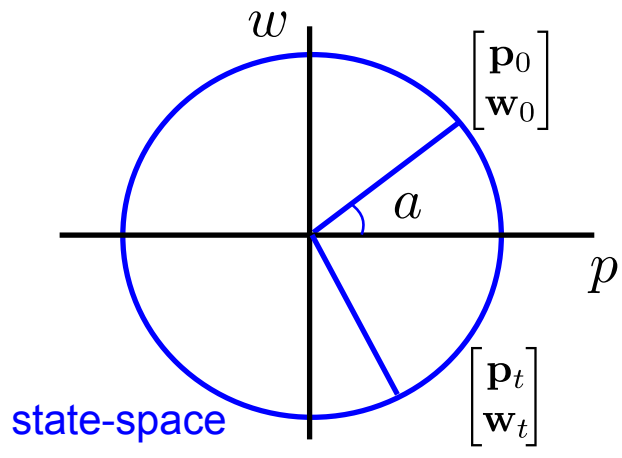
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$- \log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

$$\begin{bmatrix} \mathbf{p}_t \\ \mathbf{w}_t \end{bmatrix} = \mathrm{T}_t \left(\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{w}_0 \end{bmatrix} \right)$$

Reversibility (mapping has an inverse):

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{w} \end{bmatrix} = \mathrm{T}_{-t} \left(\mathrm{T}_t \left(\begin{bmatrix} \mathbf{p} \\ \mathbf{w} \end{bmatrix} \right) \right)$$

used to show detailed balance holds

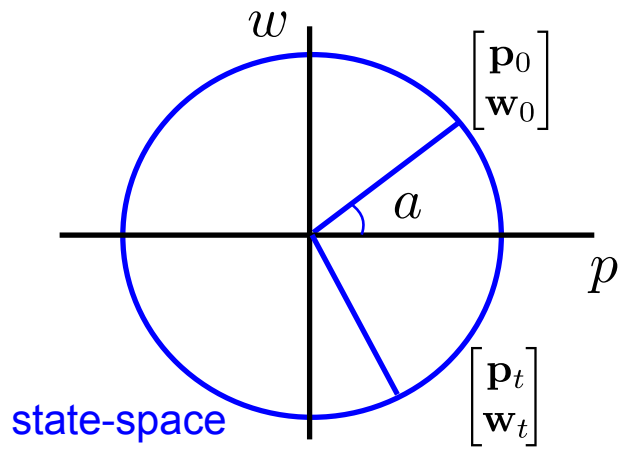
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Energy conservation (no dissipation)

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_i \frac{\partial H}{\partial w_i}\frac{\mathrm{d}w_i}{\mathrm{d}t} + \sum_i \frac{\partial H}{\partial p_i}\frac{\mathrm{d}p_i}{\mathrm{d}t}$$

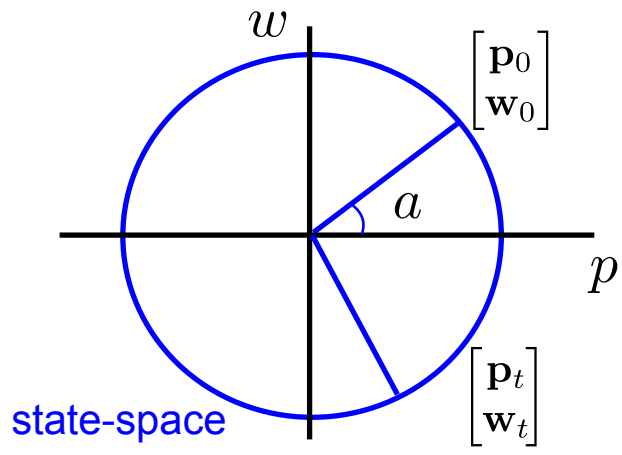
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Energy conservation (no dissipation)

$$\begin{aligned}\frac{\mathrm{d}H}{\mathrm{d}t} &= \sum_i \frac{\partial H}{\partial w_i}\frac{\mathrm{d}w_i}{\mathrm{d}t} + \sum_i \frac{\partial H}{\partial p_i}\frac{\mathrm{d}p_i}{\mathrm{d}t} \\ &= \sum_i \frac{\partial H}{\partial w_i}\frac{\partial H}{\partial p_i} - \sum_i \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial w_i} = 0\end{aligned}$$

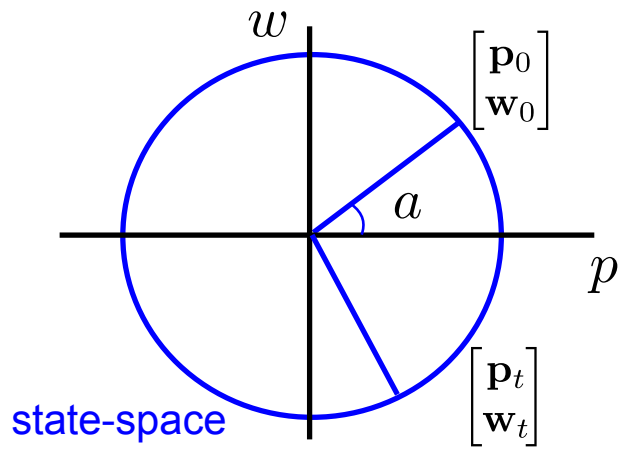
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$-\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Energy conservation (no dissipation)

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_i \frac{\partial H}{\partial w_i}\frac{\mathrm{d}w_i}{\mathrm{d}t} + \sum_i \frac{\partial H}{\partial p_i}\frac{\mathrm{d}p_i}{\mathrm{d}t}$$

$$= \sum_i \frac{\partial H}{\partial w_i}\frac{\partial H}{\partial p_i} - \sum_i \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial w_i} = 0$$

if H = constant, acceptance rate of HMC =1 (numerics)

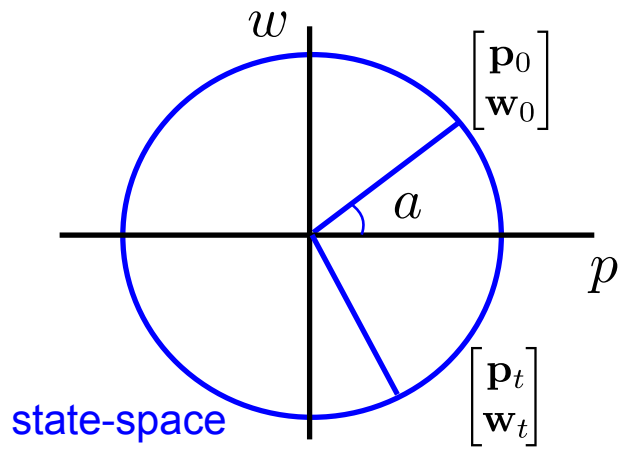
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



Hamiltonian Monte Carlo: Neal 1994

idea: use gradient information of log-target to shape MH proposal

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top$$

weights: spatial variables (potential energy)

$$U(\mathbf{w}) = -\log p^*(\mathbf{w})$$

augment with momentum variables (kinetic energy)

Hamiltonian dynamics:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

Volume preservation (Liouville's theorem)

$$\begin{bmatrix} \mathbf{p}_t \\ \mathbf{w}_t \end{bmatrix} = \mathrm{T}_t\left(\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{w}_0 \end{bmatrix}\right)$$

image of region of state-space has volume indepedent of t
shape of region will change over time
(don't need to account for Jacobians of T in MCMC)

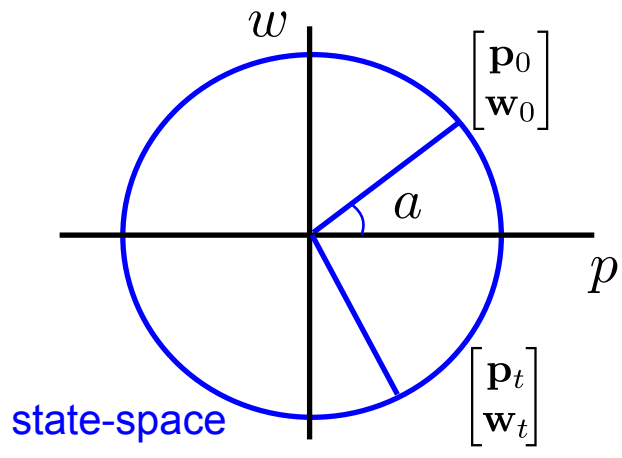
$$H(w, p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = p \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -w$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}t^2} = -w \qquad \frac{\mathrm{d}^2p}{\mathrm{d}t^2} = -p$$

$$w(t) = r\cos(a + t)$$

$$p(t) = -r\sin(a + t)$$



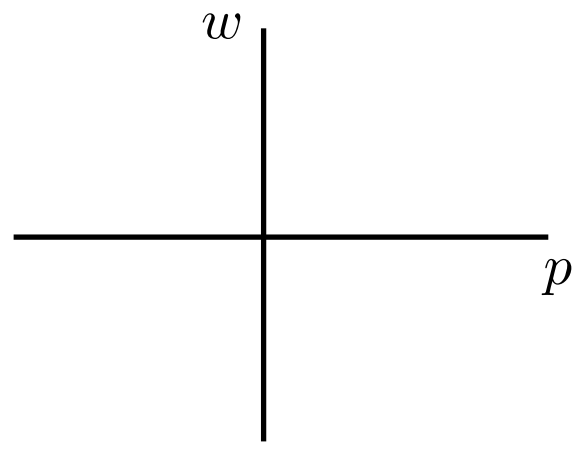
Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$
$$w(t) = r\cos(a+t)$$
$$p(t) = -r\sin(a+t)$$



Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

Move 1: resample momenta from marginal (leaves dist invariant)

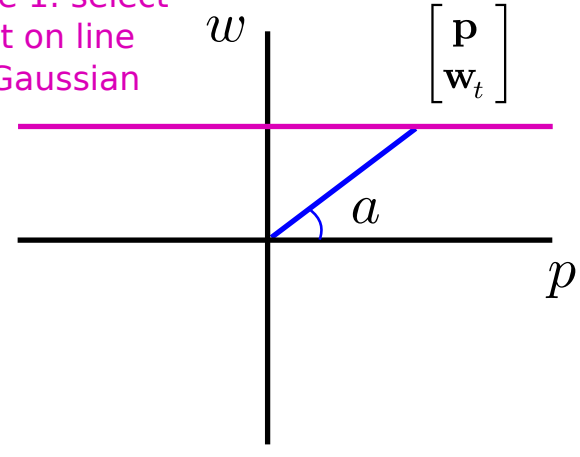
$$\mathbf{p} \sim \mathcal{G}(\mathbf{p};0,\mathbf{I}) \qquad \{\mathbf{w}_t,\mathbf{p}_t\} \longrightarrow \{\mathbf{w}_t,\mathbf{p}\}$$

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$w(t) = r\cos(a+t)$$

$$p(t) = -r\sin(a+t)$$

Move 1: select point on line via Gaussian



Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

Move 1: resample momenta from marginal (leaves dist invariant)

$$\mathbf{p} \sim \mathcal{G}(\mathbf{p};0,\mathbf{I}) \qquad \{\mathbf{w}_t,\mathbf{p}_t\} \longrightarrow \{\mathbf{w}_t,\mathbf{p}\}$$

Move 2: use Hamiltonian dynamics to make MH update:

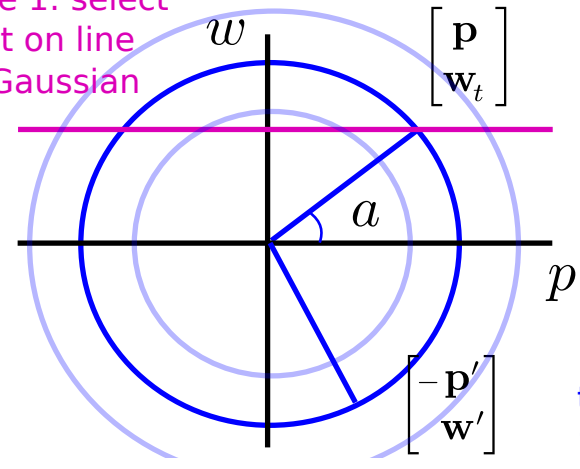
$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i} \longrightarrow \{\mathbf{p}',\mathbf{w}'\}$$

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$w(t) = r\cos(a+t)$$

$$p(t) = -r\sin(a+t)$$

Move 1: select point on line via Gaussian



Move 2: simulate trajectory for time s

Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

Move 1: resample momenta from marginal (leaves dist invariant)

$$\mathbf{p} \sim \mathcal{G}(\mathbf{p};0,\mathbf{I}) \qquad \{\mathbf{w}_t,\mathbf{p}_t\} \longrightarrow \{\mathbf{w}_t,\mathbf{p}\}$$

Move 2: use Hamiltonian dynamics to make MH update:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i} \longrightarrow \{\mathbf{p}',\mathbf{w}'\}$$

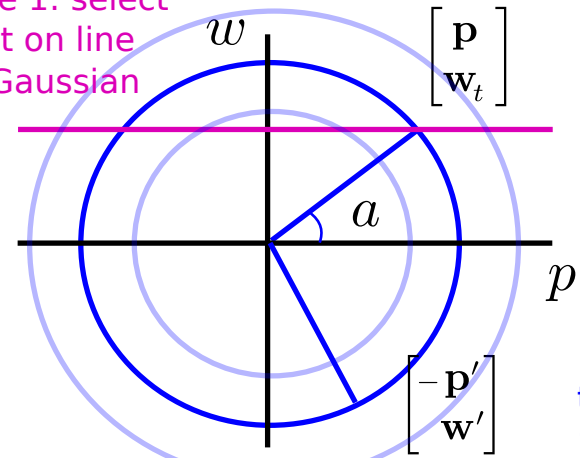
reverse momentum at end (symmetric proposal)

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$w(t) = r\cos(a+t)$$

$$p(t) = -r\sin(a+t)$$

Move 1: select point on line via Gaussian



Move 2: simulate trajectory for time s

Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

Move 1: resample momenta from marginal (leaves dist invariant)

$$\mathbf{p} \sim \mathcal{G}(\mathbf{p};0,\mathbf{I}) \qquad \{\mathbf{w}_t,\mathbf{p}_t\} \longrightarrow \{\mathbf{w}_t,\mathbf{p}\}$$

Move 2: use Hamiltonian dynamics to make MH update:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i} \longrightarrow \{\mathbf{p}',\mathbf{w}'\}$$

reverse momentum at end (symmetric proposal)

$$a = \exp(H(\mathbf{w}_t,\mathbf{p}) - H(\mathbf{w}',\mathbf{p}')) = 1$$

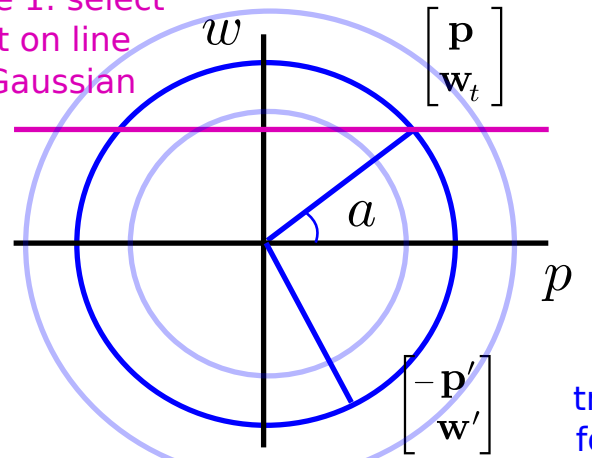
if simulation perfect

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$w(t) = r\cos(a+t)$$

$$p(t) = -r\sin(a+t)$$

Move 1: select point on line via Gaussian



Move 2: simulate trajectory for time s

Hamiltonian Monte Carlo: Neal 1994

goal: target joint using MCMC, discard momenta to get samples from $p(\mathbf{w})$

$$H(\mathbf{w},\mathbf{p}) = U(\mathbf{w}) + \tfrac{1}{2}\mathbf{p}\mathbf{p}^\top \qquad p^*(\mathbf{w},\mathbf{p}) = \exp(-H(\mathbf{w},\mathbf{p})) \propto p^*(\mathbf{w})\mathcal{G}(\mathbf{p};0,\mathbf{I})$$

variables independent

Move 1: resample momenta from marginal (leaves dist invariant)

$$\mathbf{p} \sim \mathcal{G}(\mathbf{p};0,\mathbf{I}) \qquad \{\mathbf{w}_t,\mathbf{p}_t\} \longrightarrow \{\mathbf{w}_t,\mathbf{p}\}$$

Move 2: use Hamiltonian dynamics to make MH update:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i} \longrightarrow \{\mathbf{p}',\mathbf{w}'\}$$

reverse momentum at end (symmetric proposal)

$$a = \exp(H(\mathbf{w}_t,\mathbf{p}) - H(\mathbf{w}',\mathbf{p}')) = 1$$

if simulation perfect

accept if $a \geq 1$ $\{\mathbf{w}_{t+1},\mathbf{p}_{t+1}\} = \{\mathbf{p}',\mathbf{w}'\}$

else

accept with prob a $\{\mathbf{w}_{t+1},\mathbf{p}_{t+1}\} = \{\mathbf{p}',\mathbf{w}'\}$

reject otherwise $\{\mathbf{w}_{t+1},\mathbf{p}_{t+1}\} = \{\mathbf{w}_t,\mathbf{p}\}$

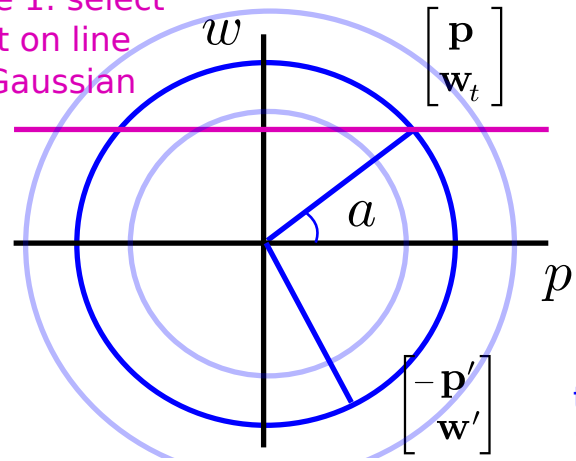
endif

$$H(w,p) = \tfrac{1}{2}w^2 + \tfrac{1}{2}p^2$$

$$w(t) = r\cos(a+t)$$

$$p(t) = -r\sin(a+t)$$

Move 1: select point on line via Gaussian



Move 2: simulate trajectory for time s

Hamiltonian Monte Carlo: Neal 1994

```
g = gradE ( x ) ;                  # set gradient using initial x
E = findE ( x ) ;                  # set objective function too

for l = 1:L                        # loop L times
   p = randn ( size(x) ) ;         # initial momentum is Normal(0,1)
   H = p' * p / 2 + E ;            # evaluate H(x,p)

   xnew = x ;  gnew = g ;
   for tau = 1:Tau                 # make Tau 'leapfrog' steps

      p = p - epsilon * gnew / 2 ; # make half-step in p
      xnew = xnew + epsilon * p ;  # make step in x
      gnew = gradE ( xnew ) ;      # find new gradient
      p = p - epsilon * gnew / 2 ; # make half-step in p

   endfor

   Enew = findE ( xnew ) ;         # find new value of H
   Hnew = p' * p / 2 + Enew ;
   dH = Hnew - H ;                 # Decide whether to accept

   if ( dH < 0 )                   accept = 1 ;
   elseif ( rand() < exp(-dH) )    accept = 1 ;
   else                            accept = 0 ;
   endif

   if ( accept )
      g = gnew ;   x = xnew ;    E = Enew ;
   endif
endfor
```

Move 1: sample momentum:

$$\mathbf{p} \sim \mathcal{G}(\mathbf{p}; 0, \mathbf{I})$$

Move 2: Hamiltonian dynamics for MH

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial w_i}$$

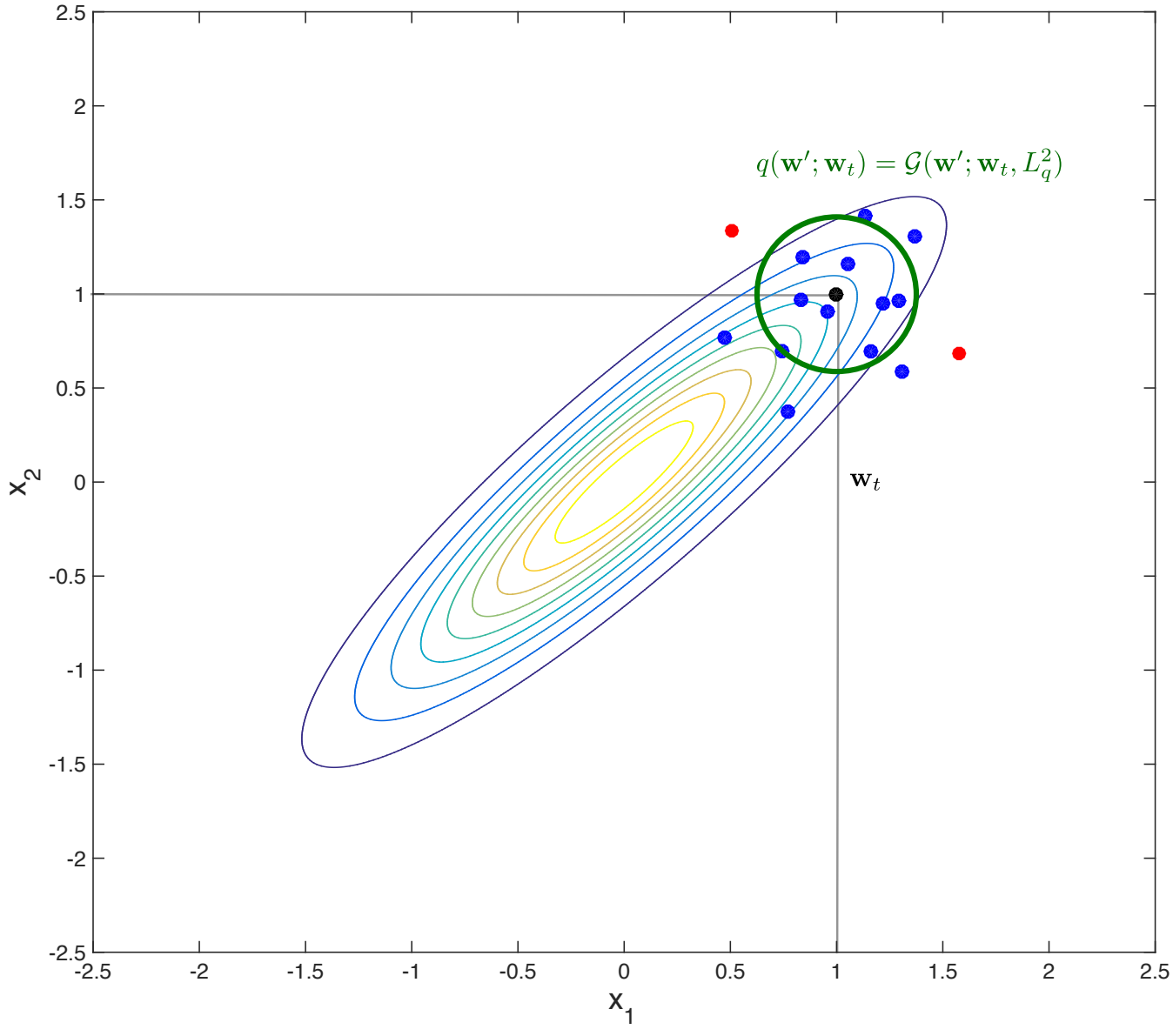
(leap-frog steps preserve volume exactly, but do not exactly preserve energy)

Accept/reject

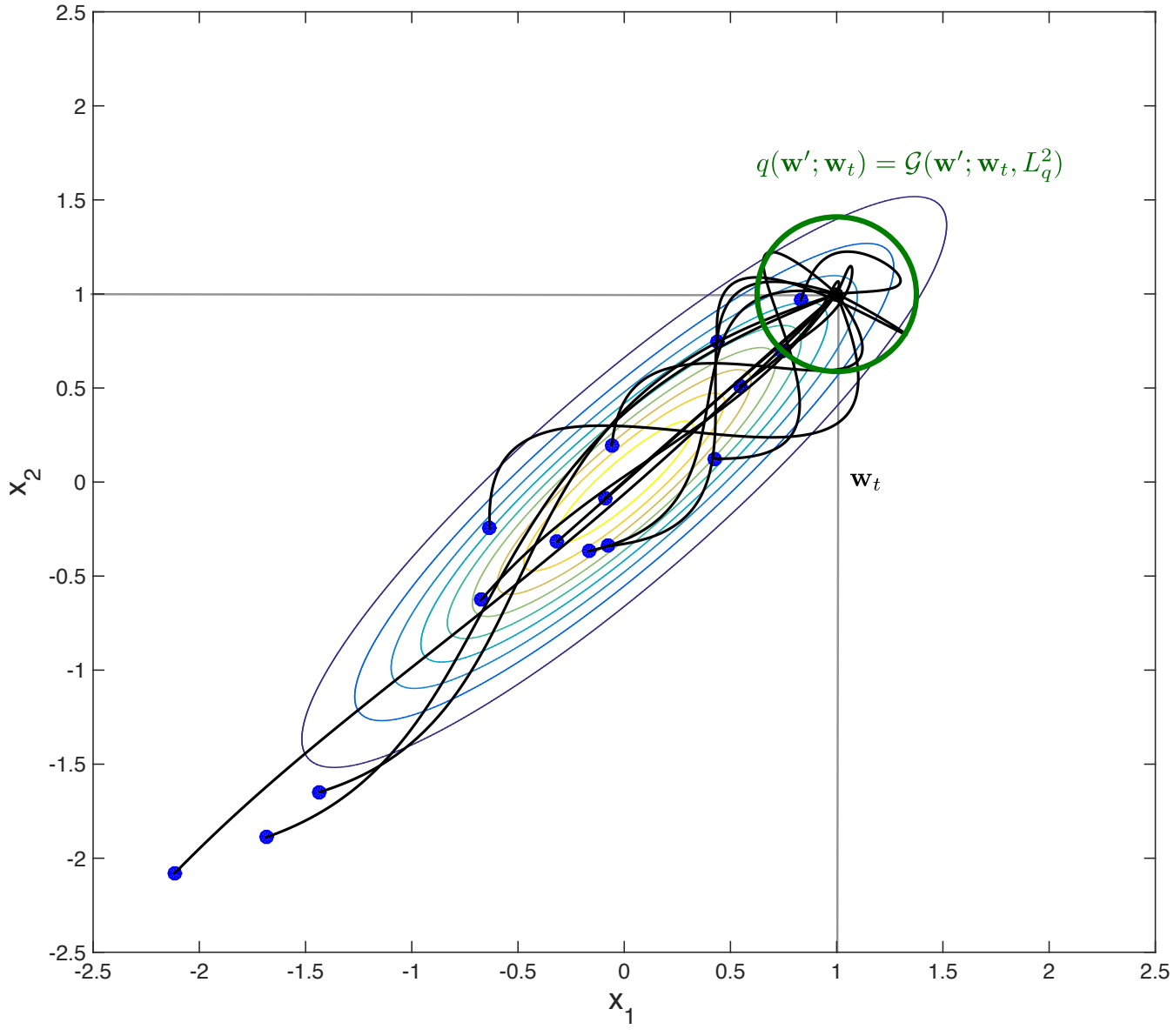
$$a = \exp(H(\mathbf{w}_t, \mathbf{p}) - H(\mathbf{w}', \mathbf{p}'))$$

pseudo-code from Mackay ITILA pg. 388

Isotropic MH proposal: Demo 2D Gaussian 15 proposals from [1,1]



Hamiltonian Monte Carlo: Demo 2D Gaussian 15 proposals from [1,1]



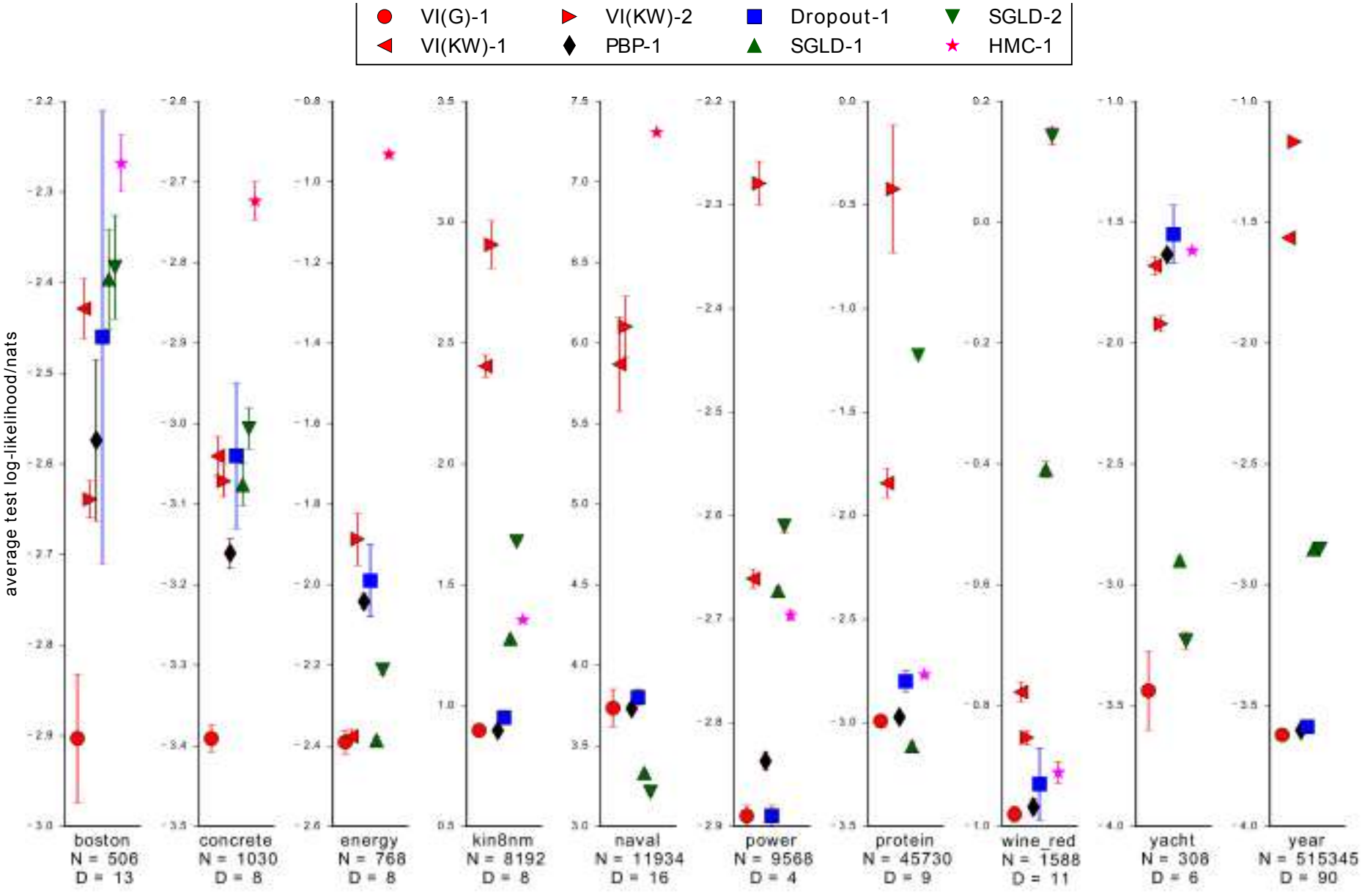
Recent developments

- HMC with one leapfrog step results in Langevin sampling: $q(\mathbf{w}'|\mathbf{w}_t) = \mathcal{G}\left(\mathbf{w}'; \mathbf{w}_t + \epsilon \frac{\mathrm{d} \log p^*(\mathbf{w})}{\mathrm{d}\mathbf{w}}, 2\epsilon\right)$
- Langevin proposal = gradient ascent + Gaussian noise
- Welling and Teh: **stochastic gradient Langevin dynamics** (SGLD):
  - stochastic gradient ascent + Gaussian noise = valid posterior sampler
  - SGA: Robbins-Munro conditions guarantee convergence
  - SGLD: Robbins-Munro conditions applied to ϵ also allow MH step to be dispensed with (nice coincidence)
- SGLD is very slow to mix
- Generalisations to HMC are in development

Bayesian Learning via Stochastic Gradient Langevin Dynamics, Welling and Teh, ICML, 2011

Stochastic Gradient Hamiltonian Monte Carlo, Chen, Fox, Guestrin, NIPS, 2015

Comparison of methods (held out log-likelihood)



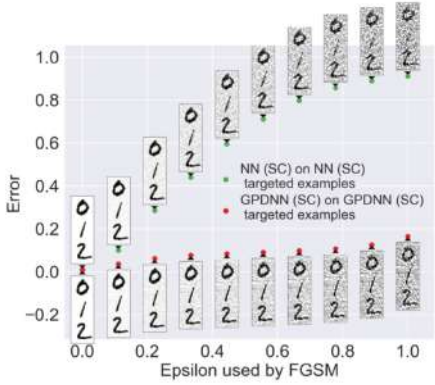
Case study 1. Robust Deep Learning
Uncertainty calibration & adversarial examples

fragile (adversarial examples)

correct diff ostrich

fragile (adversarial examples)

correct diff ostrich

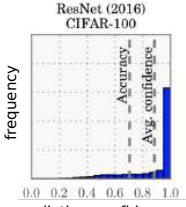


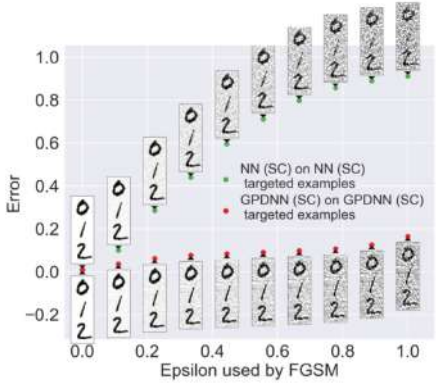
Towards Robust Deep Learning

fragile (adversarial examples)

correct diff ostrich

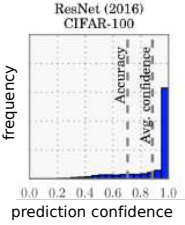
well calibrated uncertainty estimates: deep learning is often confidently wrong

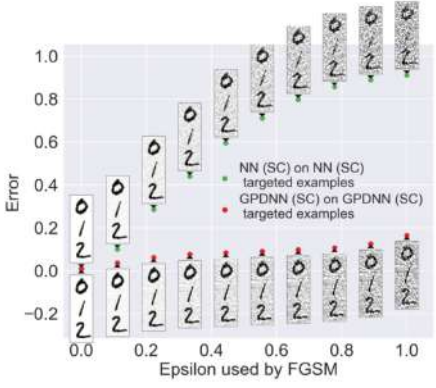


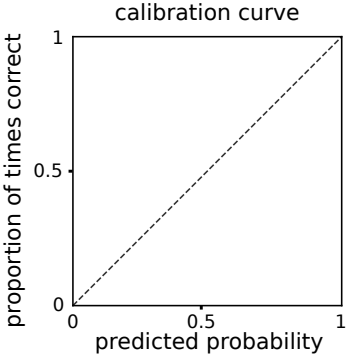


fragile (adversarial examples)

well calibrated uncertainty estimates: deep learning is often confidently wrong

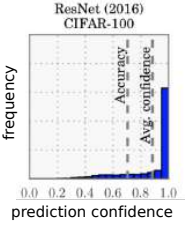


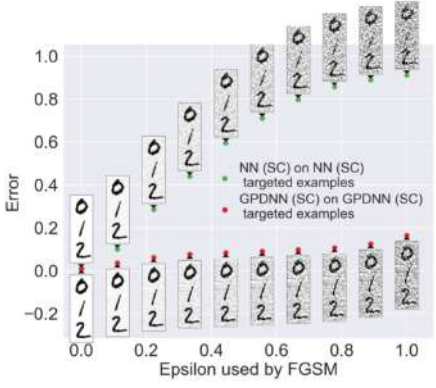


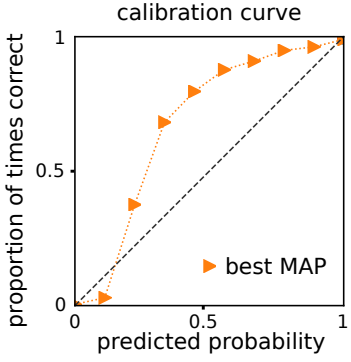


fragile (adversarial examples)

well calibrated uncertainty estimates: deep learning is often confidently wrong



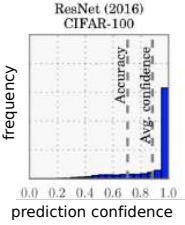


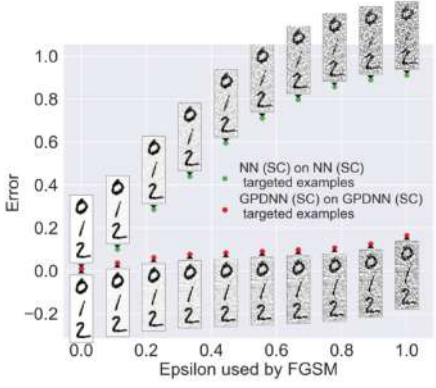


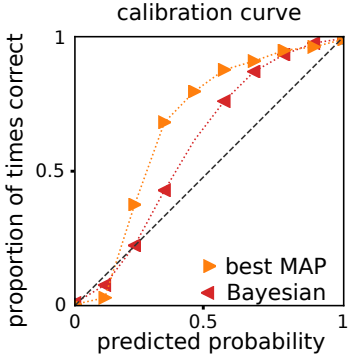
Towards Robust Deep Learning

fragile (adversarial examples)

well calibrated uncertainty estimates: deep learning is often confidently wrong







References: Case Study 1

Bayesian methods yield neural networks that have **better calibrated predictions** and which are **more robust to adversarial examples**

J. Bradshaw et al. Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks, arXiv, 2017

M. Bauer et al. Discriminative k-shot learning using probabilistic models, arXiv, 2017

C. Guo et al. On Calibration of Modern Neural Networks, ICML 2017

See also

Y. Li and Y. Gal Dropout Inference in Bayesian Neural Networks with Alpha-divergences, ICML 2017

Case study 2. Flexible Deep Learning
Continual multi-task learning

What is Continual Learning?

"corner cutter" or "line pin"

 ?

 ?

What is Continual Learning?

corner cutter

line pins

"corner cutter" or "line pin"

?

?

collect data set online

predictions

What is Continual Learning?

corner cutter

line pins

"corner cutter" or "line pin"

?

?

collect data set online

predictions

What is Continual Learning?

corner cutter

line pins

"corner cutter" or "line pin"

?

?

collect data set online

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

"corner cutter" or "line pin"

?

?

collect data set online

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

?

?

predictions

What is Continual Learning?

corner cutter

line pins

"corner cutter" or "line pin"

when new examples seen
incrementally update:

new model = f(old model,new data)

collect data set online

predictions

What is Continual Learning?

corner cutter

line pins

collect data set online

"corner cutter" or "line pin"

predictions

What is Continual Learning?

corner cutter

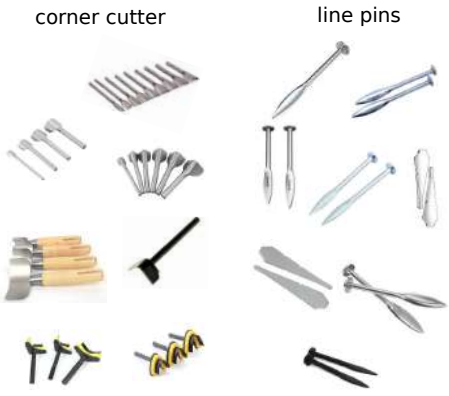
line pins

collect data set online

"corner cutter" or "line pin"

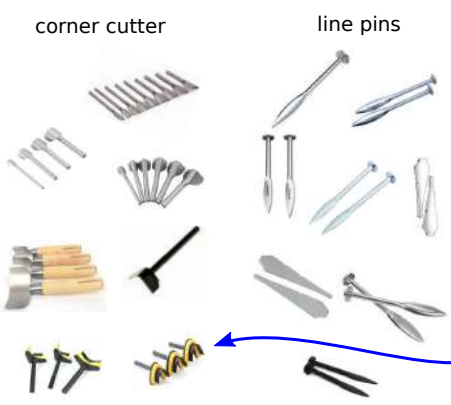
predictions

What is Continual Learning?



"corner cutter" or "line pin"

What is Continual Learning?



statistics of data may change over time: continuously adapt without revisiting all old data

"corner cutter" or "line pin"

What is Continual Learning?

corner cutter

line pins

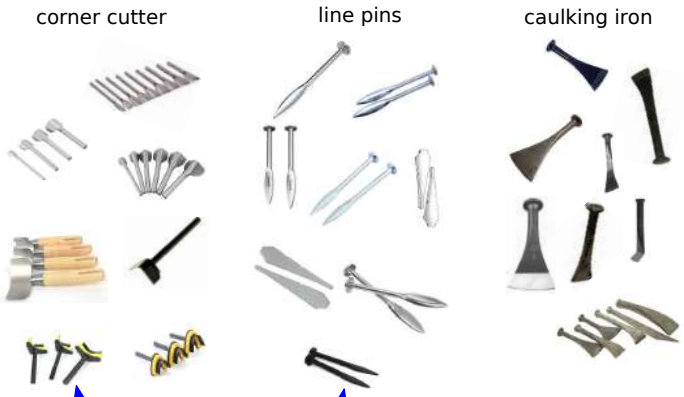
What is Continual Learning?

corner cutter

line pins

caulking iron

What is Continual Learning?



What is Continual Learning?

?

corner cutter

line pins

caulking iron

What is Continual Learning?

corner cutter

line pins

caulking iron

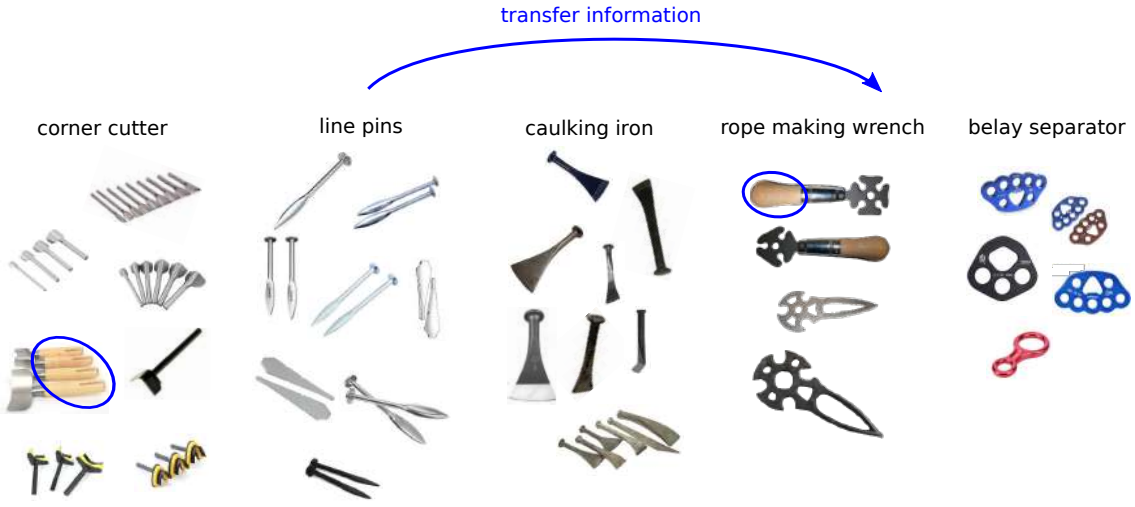
rope making wrench

belay separator

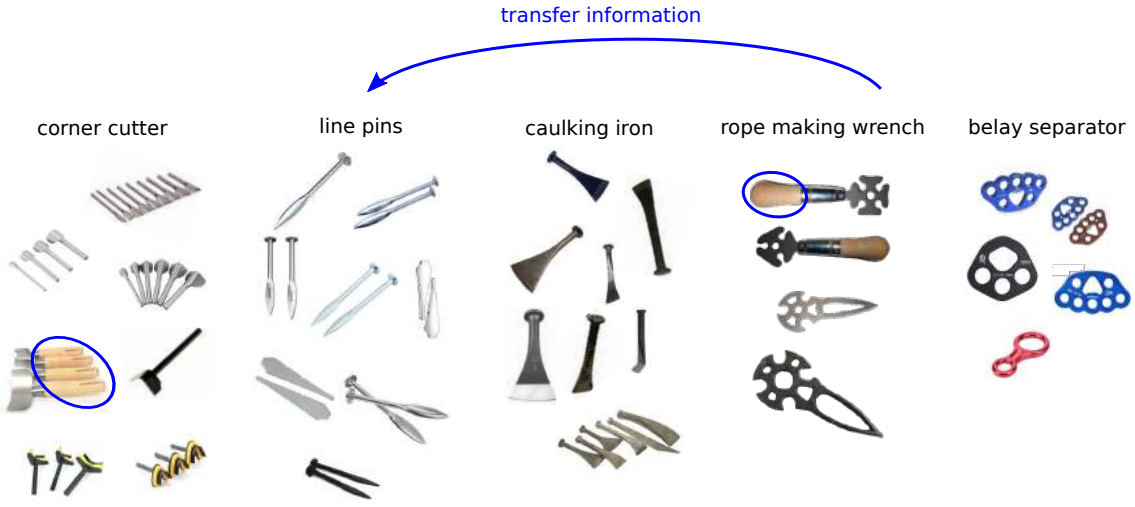
TASK 1

TASK 2

What is Continual Learning?



What is Continual Learning?



What is Continual Learning?

?

corner cutter

line pins

caulking iron

rope making wrench

belay separator

TASK 1

TASK 2

What is Continual Learning?

?

corner cutter

line pins

caulking iron

rope making wrench

belay separator

TASK 1

TASK 2

What is Continual Learning?

avoid forgetting old tasks

?

corner cutter

line pins

caulking iron

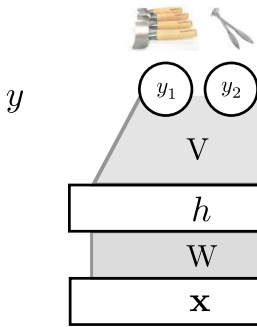
rope making wrench

belay separator

TASK 1

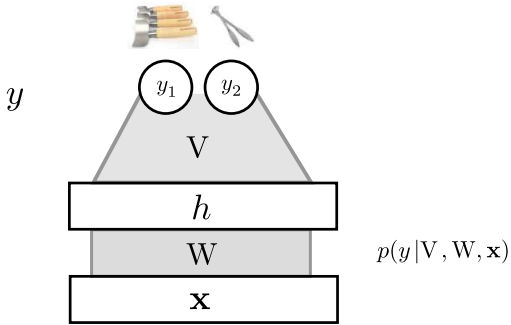
TASK 2

A zoo of discriminative continual learning tasks



$p(y|\mathrm{V}, \mathrm{W}, \mathbf{x})$

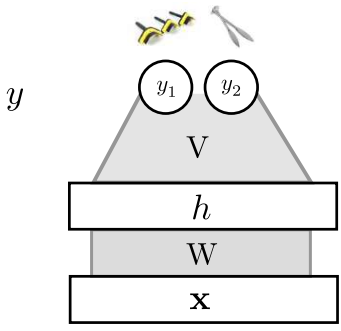
A zoo of discriminative continual learning tasks



1. online iid data (online learning)

$$p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p(\mathbf{x}_n)$$

A zoo of discriminative continual learning tasks



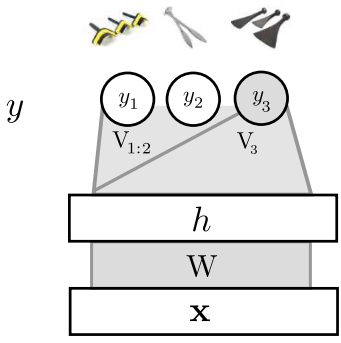
$p(y|\mathrm{V}, \mathrm{W}, \mathbf{x})$

1. online iid data (online learning) $p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p(\mathbf{x}_n)$

2. online non-iid inputs (covariate shift) $p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p_n(\mathbf{x}_n|\mathbf{x}_{1:n-1})$

A zoo of discriminative continual learning tasks

3. new classes
(k-shot learning)



$p(y|\mathrm{V}_{1:2}, \mathrm{V}_3, \mathrm{W}, \mathbf{x})$

1. online iid data
(online learning)

$p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p(\mathbf{x}_n)$

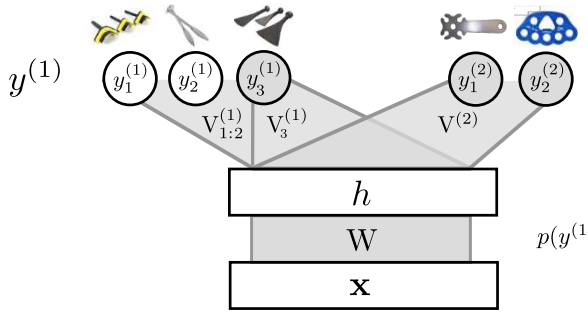
2. online non-iid inputs
(covariate shift)

$p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p_n(\mathbf{x}_n|\mathbf{x}_{1:n-1})$

A zoo of discriminative continual learning tasks

3. new classes
(k-shot learning)

4. new tasks
(online multi-task learning
transfer learning)



$p(y^{(1)}, y^{(2)} | \mathrm{V}^{(1)}, \mathrm{V}^{(2)}, \mathrm{W}, \mathbf{x})$

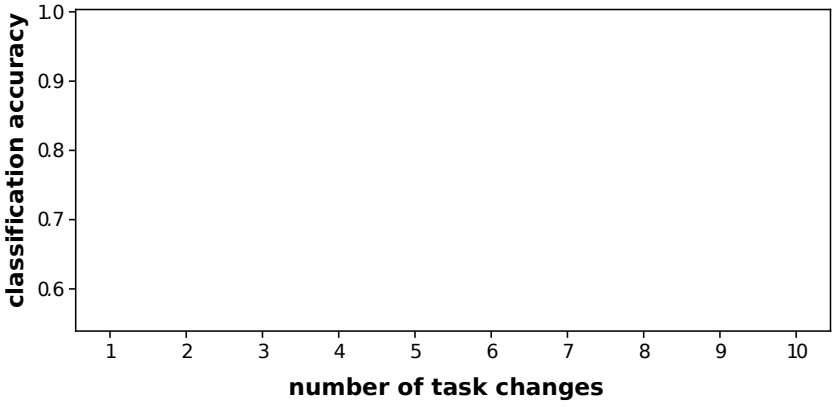
1. online iid data
(online learning)

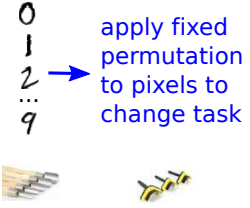
$$p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p(\mathbf{x}_n)$$

2. online non-iid inputs
(covariate shift)

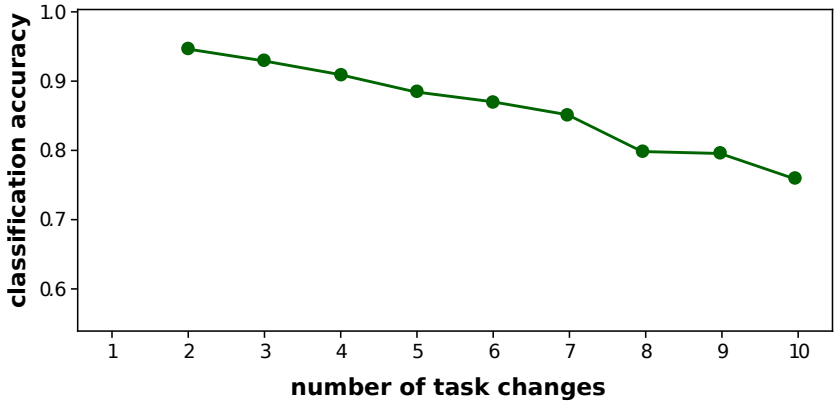
$$p(\mathbf{x}_{1:N}) = \prod_{n=1}^{N} p_n(\mathbf{x}_n | \mathbf{x}_{1:n-1})$$

Continual Learning Test 1: Permuted MNIST (online non-iid inputs, single head)

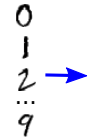




Continual Learning Test 1: Permuted MNIST (online non-iid inputs, single head)

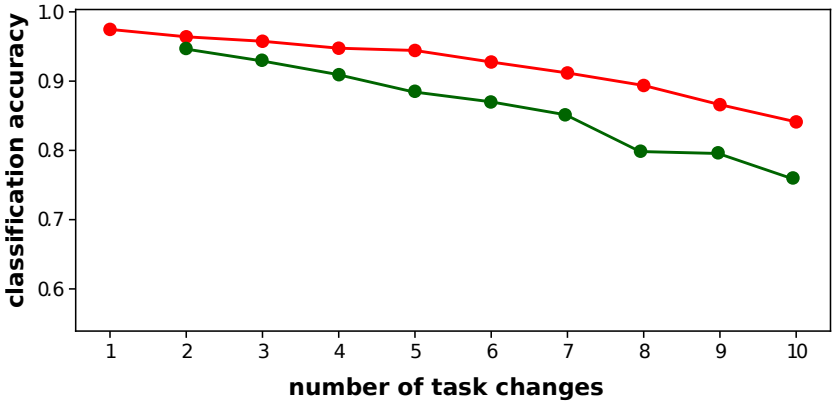


Deep baseline

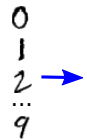


apply fixed permutation to pixels to change task

Continual Learning Test 1: Permuted MNIST (online non-iid inputs, single head)

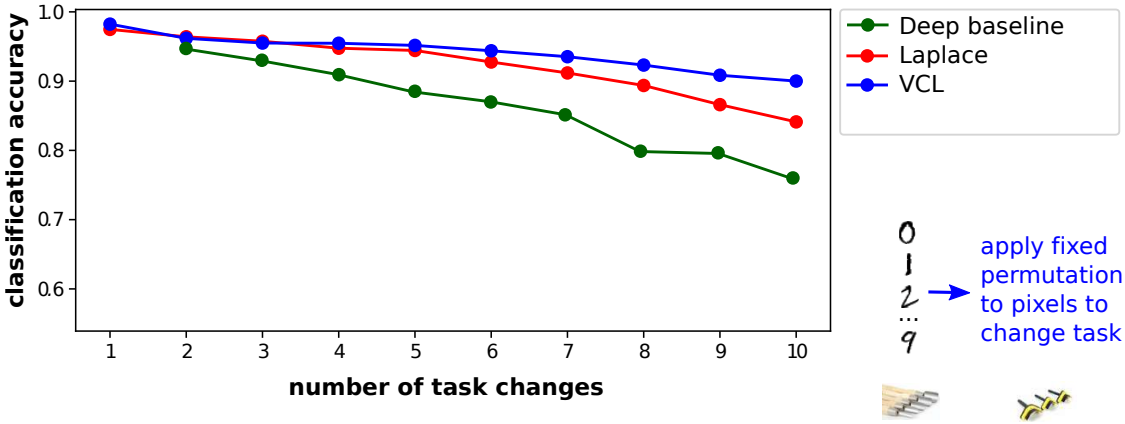


Deep baseline
Laplace

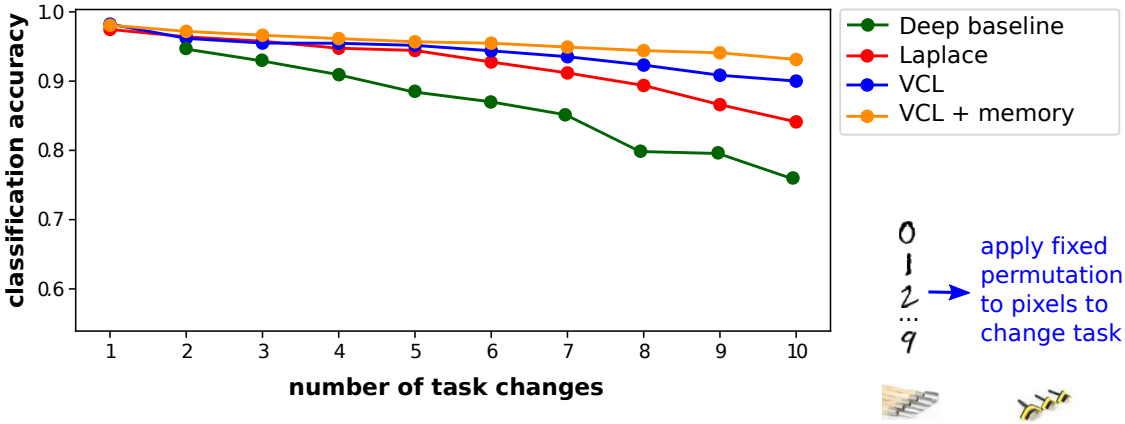


apply fixed permutation to pixels to change task

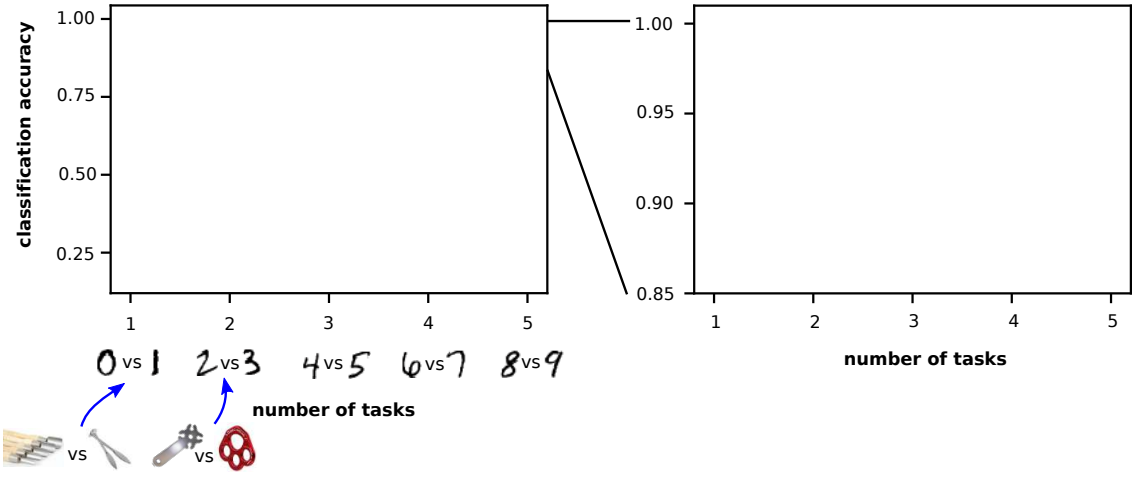
Continual Learning Test 1: Permuted MNIST (online non-iid inputs, single head)



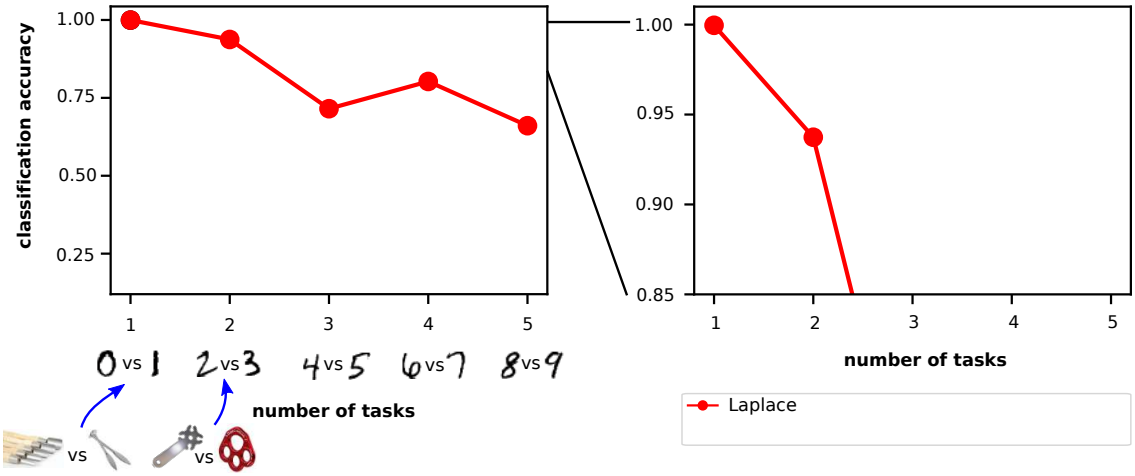
Continual Learning Test 1: Permuted MNIST (online non-iid inputs, single head)



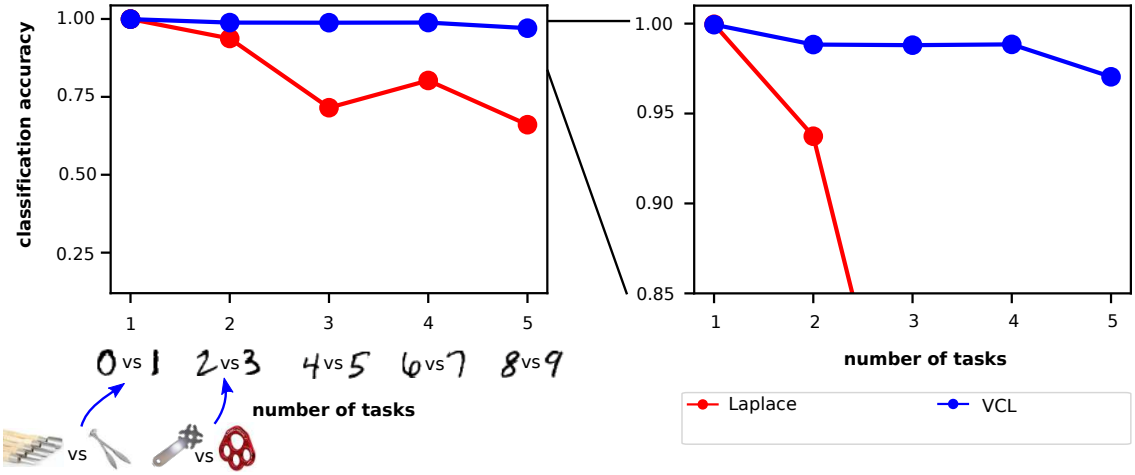
Continual Learning Test 2: Split MNIST (new tasks, multi-head)



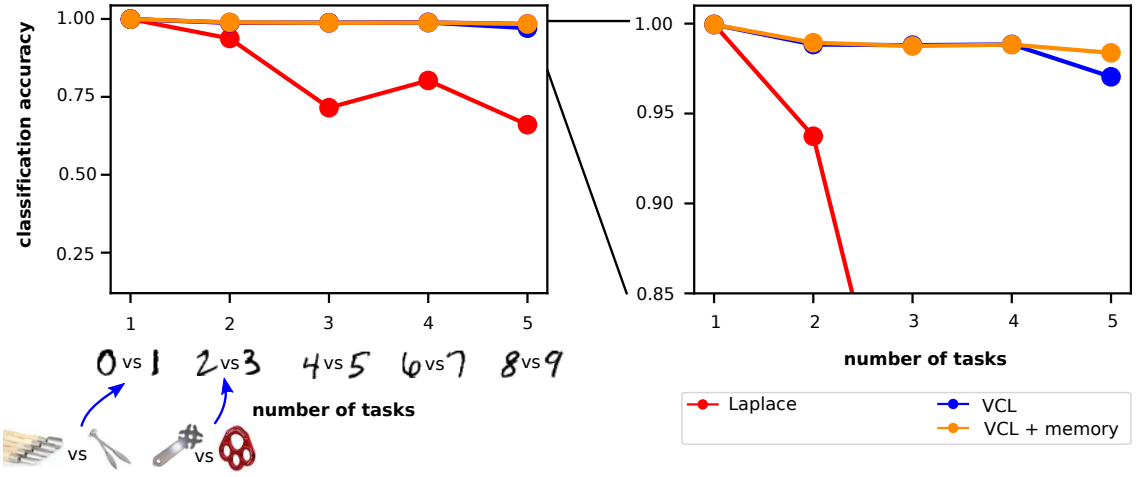
Continual Learning Test 2: Split MNIST (new tasks, multi-head)



Continual Learning Test 2: Split MNIST (new tasks, multi-head)



Continual Learning Test 2: Split MNIST (new tasks, multi-head)



References: Case Study 2

- **Continual learning** is naturally handled by **Bayesian inference**: allows multi-task transfer and avoids catastrophic forgetting
- **Variational Continual Learning** is a **state-of-the-art continual learning method**
- Orthogonal research directions: **complex models** (adapting more than just the head of the network) and **online automatic model building**

J. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks, PNAS, 2017

C. Nguyen et al. Variational Continual Learning, ICLR 2018

- See also:

F. Huszar On Quadratic Penalties in Elastic Weight Consolidation, PNAS, 2018

F. Zenke et al. Continual Learning Through Synaptic Intelligence, ICML 2017

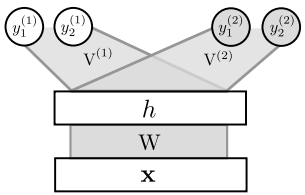
Case study 3: Data Efficient Deep Learning

One-shot learning

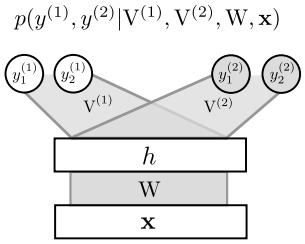
One shot learning using Approximate Bayesian inference

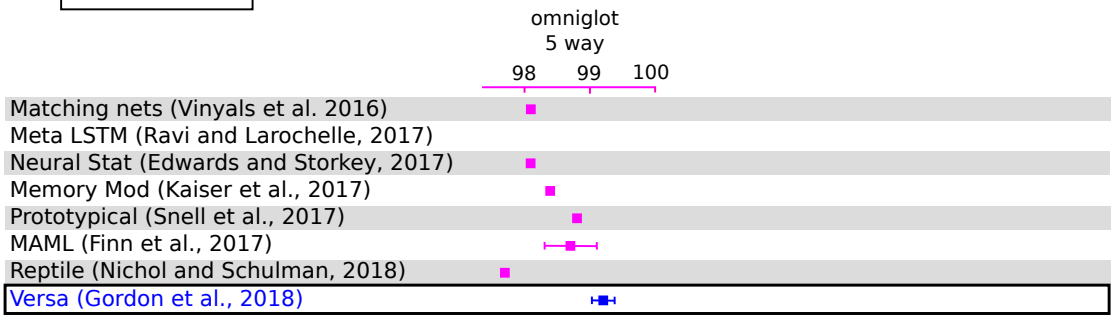
One shot learning using Approximate Bayesian inference

$$p(y^{(1)}, y^{(2)} | \mathrm{V}^{(1)}, \mathrm{V}^{(2)}, \mathrm{W}, \mathbf{x})$$

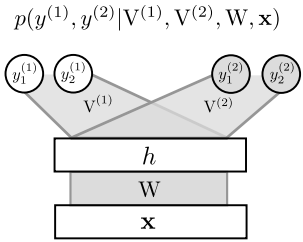


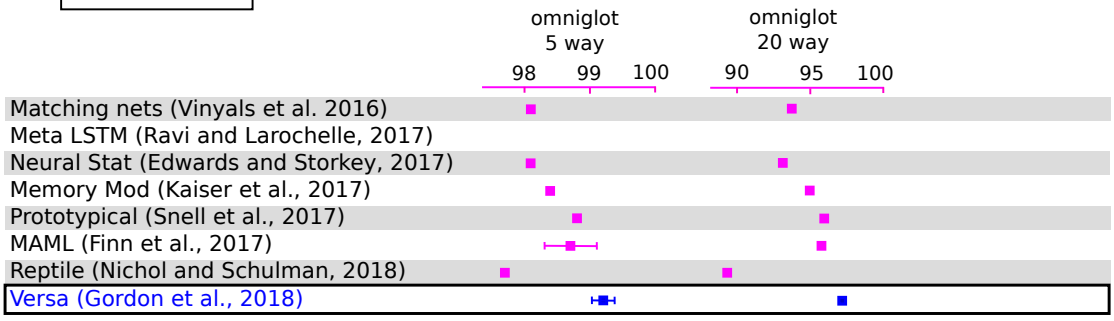
One shot learning using Approximate Bayesian inference



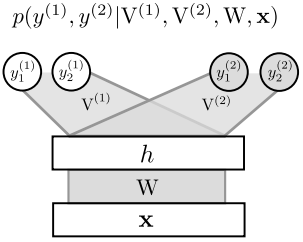


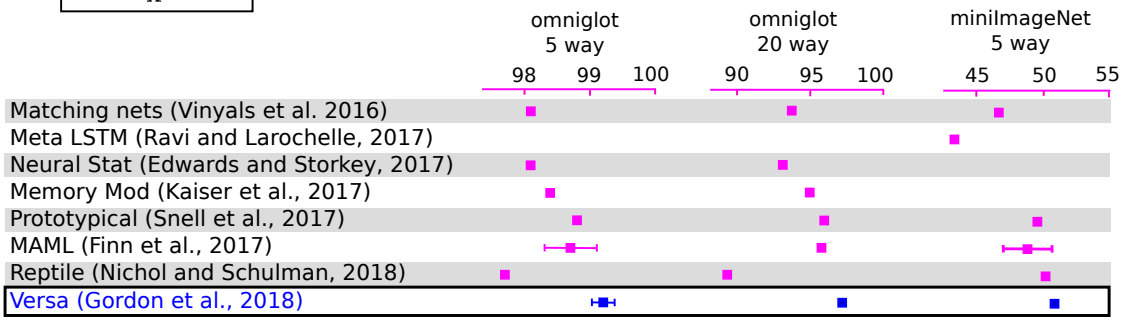
One shot learning using Approximate Bayesian inference





One shot learning using Approximate Bayesian inference





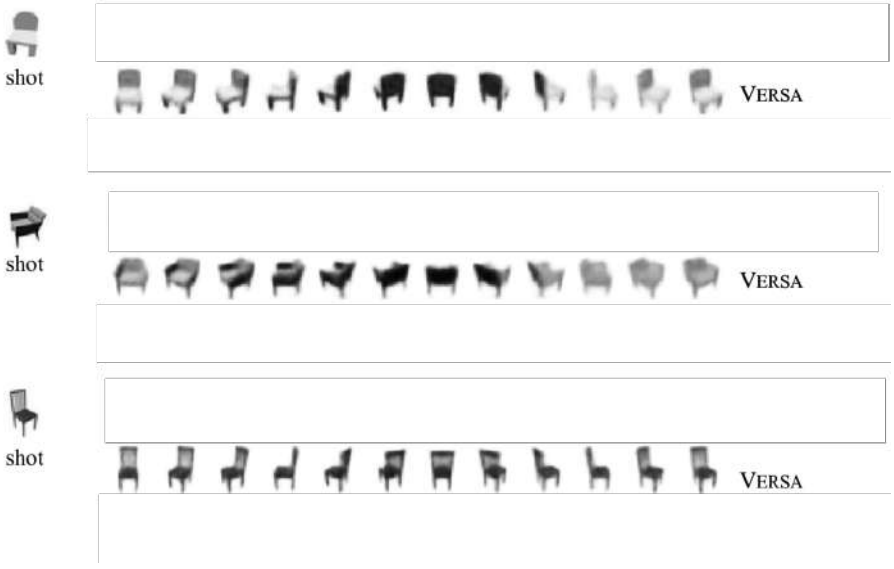
One shot learning using Approximate Bayesian inference


shot

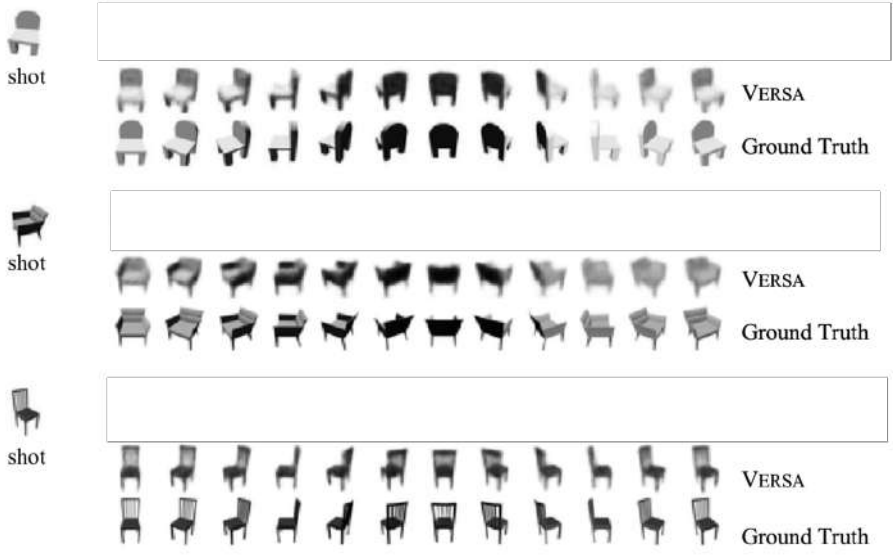

shot


shot

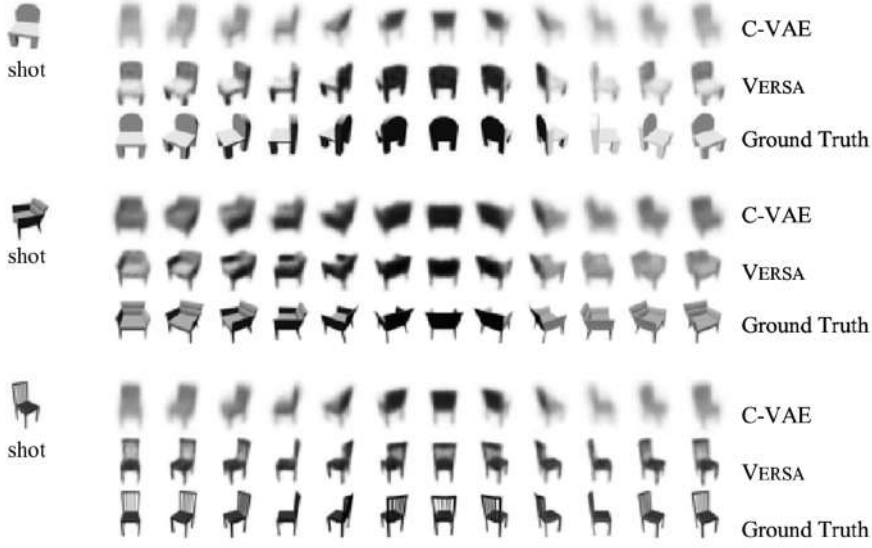
One shot learning using Approximate Bayesian inference



One shot learning using Approximate Bayesian inference



One shot learning using Approximate Bayesian inference



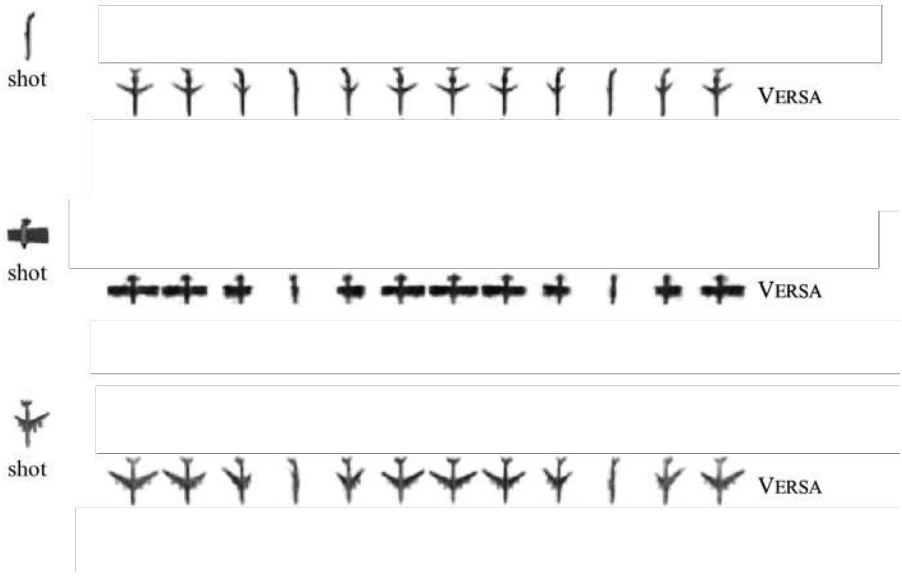
One shot learning using Approximate Bayesian inference


shot

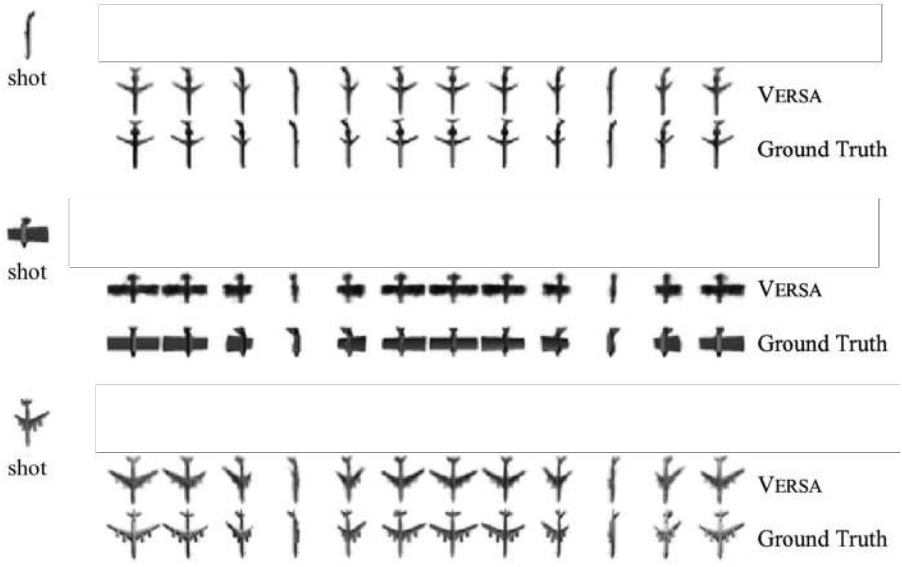

shot


shot

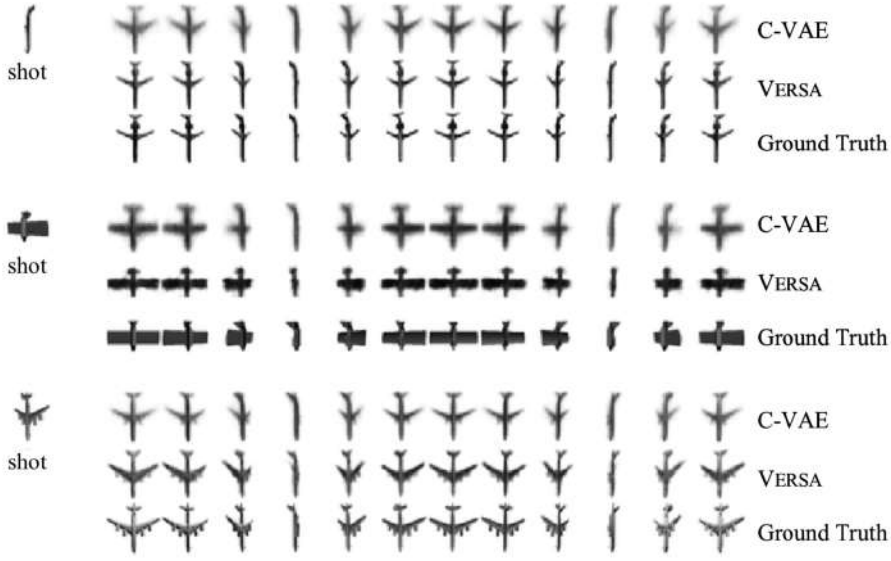
One shot learning using Approximate Bayesian inference



One shot learning using Approximate Bayesian inference



One shot learning using Approximate Bayesian inference



References: Case Study 3

- Bayesian methods enable **neural networks to be accurately trained on small data** and probabilistic modelling allows **statistical information to be shared between many different data sources**

J. Gordon et al. Decision-Theoretic Meta Learning: Versatile and Efficient Amortization of Few-Shot Learning, arXiv, 2018

See also

E. Grant et al. Recasting Gradient Based Meta-Learning as Hierarchical Bayes, ICLR, 2018

C. Finn et al. Probabilistic Model-Agnostic Meta-Learning, arXiv, 2018

T. Kim et al. Bayesian Model-Agnostic Meta-Learning, arXiv, 2018

Why be Bayesian (distributional estimates of weights)?

	maximum-likelihood	Bayes
learning	$\mathbf{w}^{\text{ML}} = \arg\max_{\mathbf{w}} \log P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$	$p(\mathbf{w}\|\mathcal{Y}, \mathcal{X}) = \frac{1}{P(\mathcal{Y}\|\mathcal{X})}\, p(\mathbf{w})\; P(\mathcal{Y}\|\mathbf{w}, \mathcal{X})$
prediction	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) \approx p(y^*\|\mathbf{w}^{\text{ML}}, x^*)$	$p(y^*\|x^*, \mathcal{Y}, \mathcal{X}) = \int p(\mathbf{w}\|\mathcal{Y}, \mathcal{X})\, p(y^*\|\mathbf{w}, x^*) \mathrm{d}\mathbf{w}$
	single weight setting	ensemble over weight settings
		requires approximation

Robust Deep Learning

point estimates over-confident, averaging over weight settings less so
Bayesian methods are more robust to adversarial examples (hard to fool ensemble of networks + uncertainty)

Data-efficient Deep Learning

small data, big model: build models 'the size of a house' & let data prune/learn structure
leverage heterogeneous data sources (multi-task learning) using shared parameters

Flexibile Deep Learning

continual learning: use old posterior as prior
active learning: select data that are expected to reduce uncertainty in parameter estimates the most

Acknowledgements

Thanks to **Stratis Markou** for helping to prepare the slides.

Thanks also to **members of my research group** including Matthias Bauer, John Bronksill, Thang Bui, Jonathan Gordon, Yingzhen Li, Cuong V. Nguyen, Sebastian Nowozin, and Mateo Rojas-Carulla, who contributed to some of the work covered in the Case Studies.