Nadia Chirkova

Bayesian linear regression

2021

Slides are partially based on lectures of Dmitry Vetrov, Dmitry Kropotov and Kirill Struminsky, deepbayes.ru/2018

Plan

- Linear regression: reminder
- Bayesian linear regression:
    - model definition
    - training
    - prediction

Plan

- Linear regression: reminder
- Bayesian linear regression:
    - model definition
    - training
    - prediction

Linear regression: reminder

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

N — number of objects

d — number of features

Linear regression: reminder

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

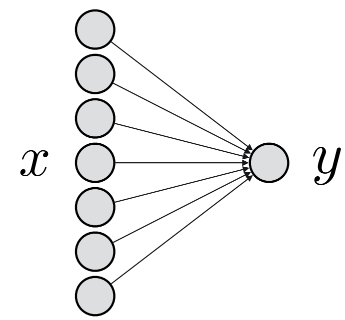
N — number of objects

d — number of features

Model:

$$Xw \approx Y$$

$$x_i^T w \approx y_i$$



linear model
with weights w

Linear regression: reminder

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

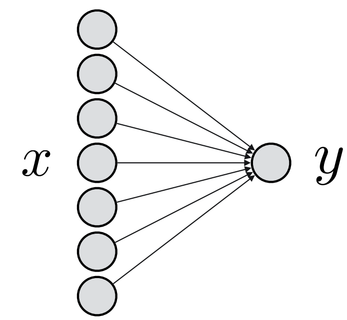
N — number of objects

d — number of features

Model:

$$Xw \approx Y$$

$$x_i^T w \approx y_i$$



linear model
with weights w

Applications:

- bioinformatics
- physics
- economics
- text processing
- search engines …

…

Linear regression: reminder

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

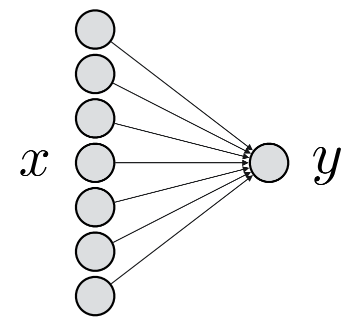
N — number of objects

d — number of features

Model:

$Xw \approx Y$

$x_i^T w \approx y_i$



linear model
with weights w

Training:

$$\frac{1}{N} \sum_{i=1}^{N} (x_i^T w - y_i)^2 \to \min_{w \in \mathbb{R}^d}$$

Prediction on a new object x_* :

$$a(x_*) = x_*^T w$$

Linear regression: reminder

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

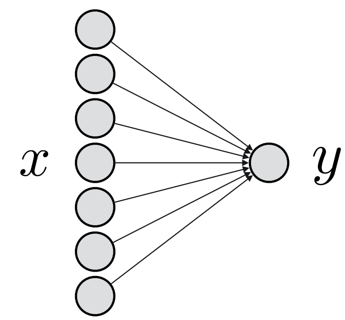
N — number of objects

d — number of features

Model:

$$Xw \approx Y$$

$$x_i^T w \approx y_i$$



linear model
with weights w

Training:

$$\frac{1}{N}\|Xw - Y\|^2 \to \min_{w \in \mathbb{R}^d}$$

Prediction on a new object x_* :

$$a(x_*) = x_*^T w$$

Linear regression: training

$$L(w) = \frac{1}{N}\|Xw - Y\|^2 \to \min_{w \in \mathbb{R}^d}$$

Convex function:

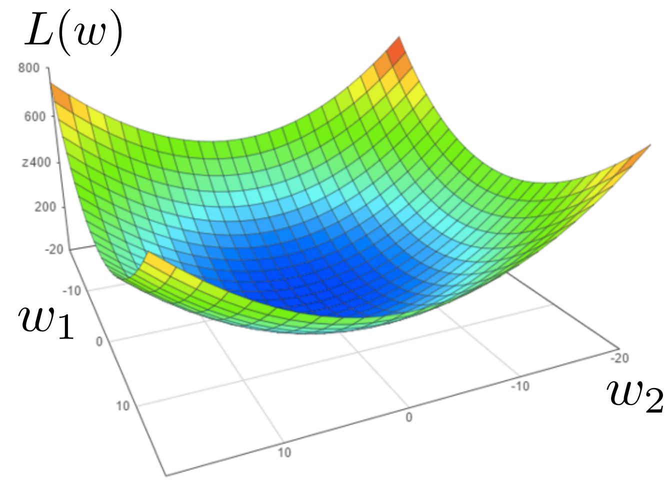


Image from https://www.globalsoftwaresupport.com/linear-regression/

Linear regression: training

$$L(w) = \frac{1}{N}\|Xw - Y\|^2 \to \min_{w \in \mathbb{R}^d}$$

Optimal weights:

$$w_{ML} = (X^T X)^{-1} X^T Y$$

— if $rank(X^T X) = d$,
otherwise infinite number of solutions

Convex function:

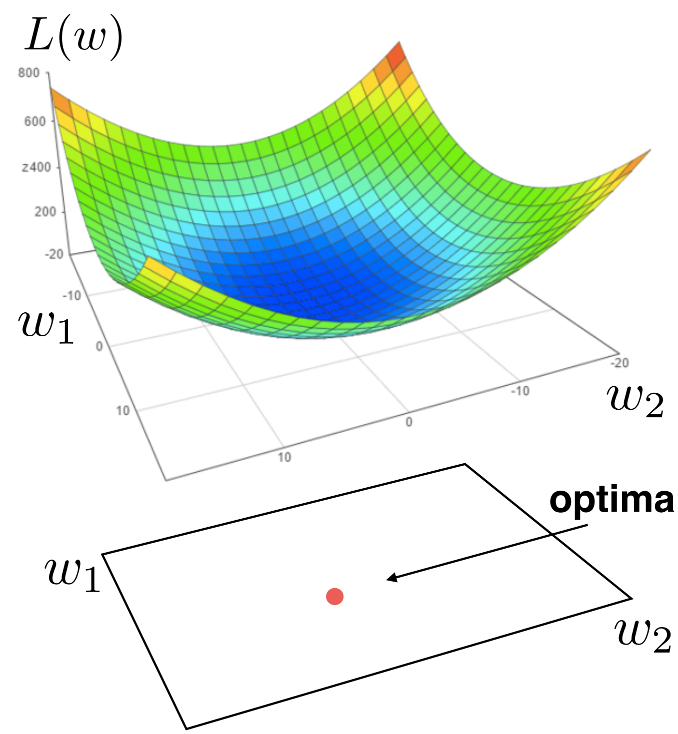


Image from https://www.globalsoftwaresupport.com/linear-regression/

Linear regression: regularization

$$L(w) = \frac{1}{N}\|Xw - Y\|^2 + \lambda\|w\|^2 \to \min_{w \in \mathbb{R}^d}$$

$$\lambda > 0$$

Optimal weights:

$$w_{MP} = (X^T X + \lambda I)^{-1} X^T Y$$

- Always unique solution
- Preventing overfitting

Strongly convex function:

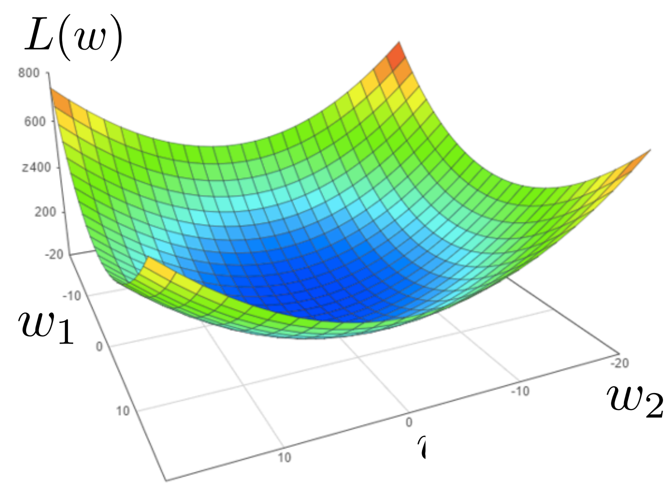


Image from https://www.globalsoftwaresupport.com/linear-regression/

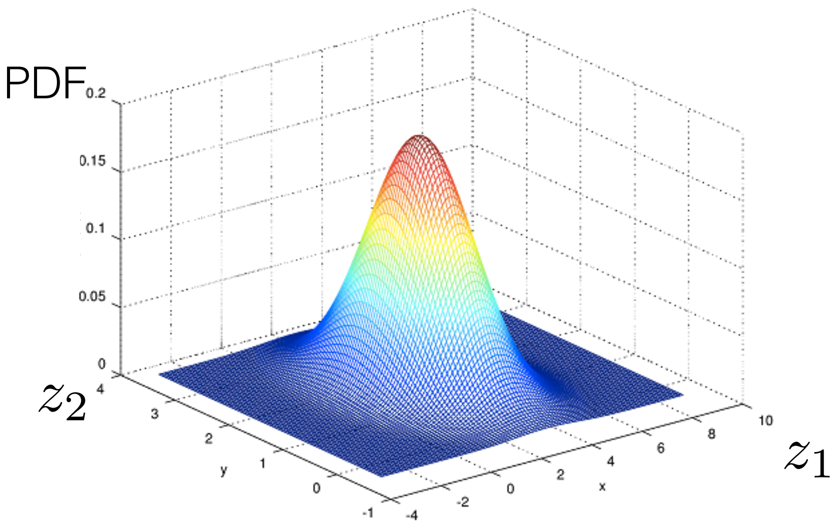
Plan

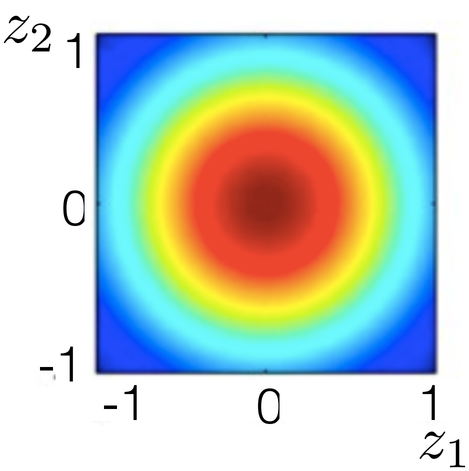
- Linear regression: reminder
- $\boxed{\text{Bayesian linear regression:}}$
    - model definition
    - training
    - prediction

Multivariate normal (Gaussian) distribution

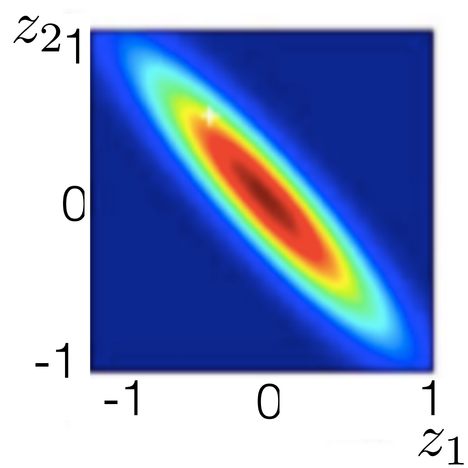
$$\mathcal{N}(z|\mu,\Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp(-\tfrac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)),$$

$z \in \mathbb{R}^d$
$\mu \in \mathbb{R}^d$
$\Sigma \in \mathbb{R}^{d\times d}$





diagonal Σ



non-diagonal Σ

Images from https://medium.com/ming-learns-thing/machine-learning-bayesian-linear-regression-f160c4eaef99,
C. Bishop. Pattern Recognition and Machine Learning

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

N — number of objects

d — number of features

Model:

$$p(Y, w|X) = p(Y|X, w)p(w)$$

how does target Y depend on input X?

what weights w do we expect?

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^N$ — target values

N — number of objects

d — number of features

Model:

$$p(Y, w|X) = p(Y|X, w)p(w)$$

- likelihood:

$$p(Y|X, w) = \prod_{i=1}^{N} \mathcal{N}(y_i|x_i^T w, 1) =$$

$$= \mathcal{N}(Y|Xw, I)$$

- prior ?

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

N — number of objects

d — number of features

Model:

$$p(Y, w|X) = p(Y|X, w)p(w)$$

- likelihood:

$$p(Y|X, w) = \prod_{i=1}^{N} \mathcal{N}(y_i | x_i^T w, 1) =$$

$$= \mathcal{N}(Y|Xw, I)$$

- conjugate prior:

$$p(w) = \mathcal{N}(w|0, \alpha I), \ \alpha > 0$$

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

N — number of objects

d — number of features

Training? Prediction?

Model:

$$p(Y, w|X) = p(Y|X, w)p(w)$$

- likelihood:

$$p(Y|X, w) = \prod_{i=1}^{N} \mathcal{N}(y_i|x_i^T w, 1) =$$

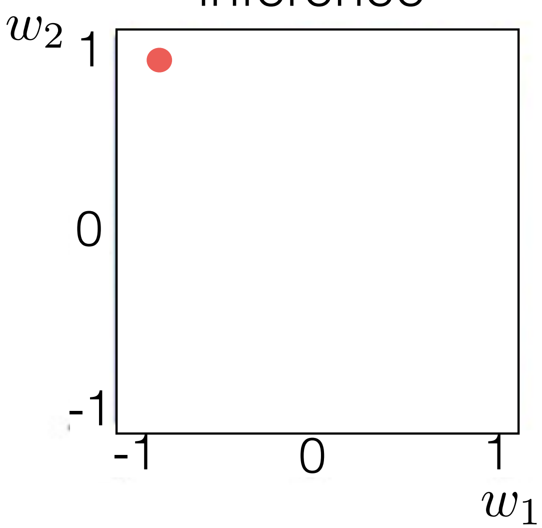
$$= \mathcal{N}(Y|Xw, I)$$

- conjugate prior:

$$p(w) = \mathcal{N}(w|0, \alpha I), \ \alpha > 0$$

Options of training Bayesian linear regression

Maximum likelihood inference

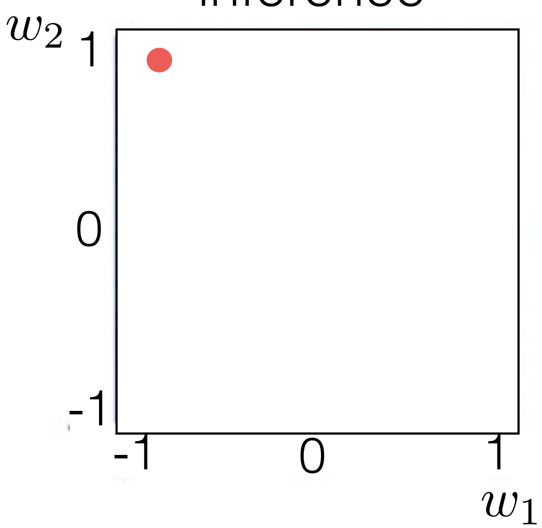


$$p(Y|X,w) \to \max_w$$

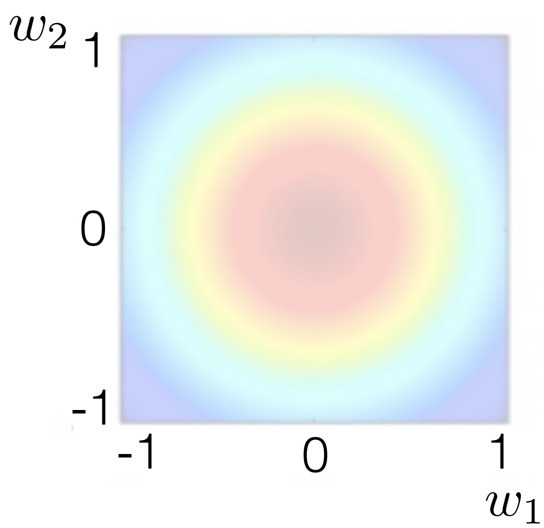
Image draft from C. Bishop. Pattern Recognition and Machine Learning

Options of training Bayesian linear regression

Maximum likelihood inference



Prior distribution

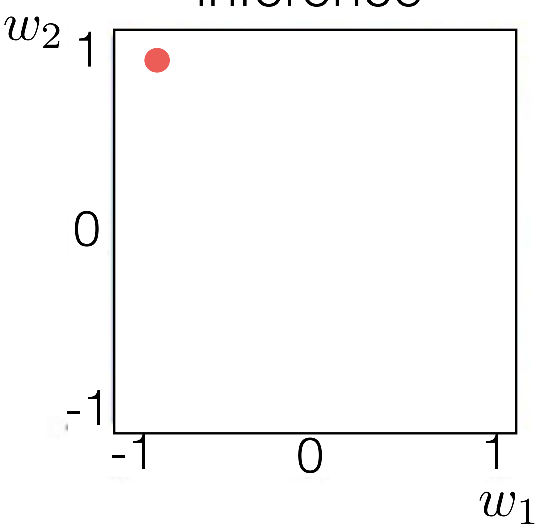


$$p(Y|X,w) \to \max_w$$

Image draft from C. Bishop. Pattern Recognition and Machine Learning

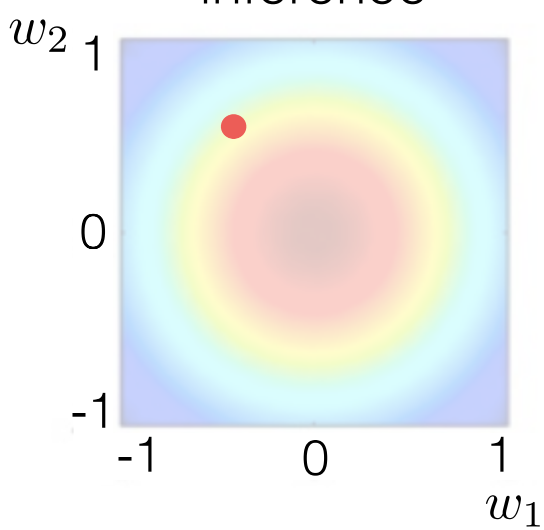
Options of training Bayesian linear regression

Maximum likelihood inference



$$p(Y|X, w) \to \max_w$$

Maximum posterior inference

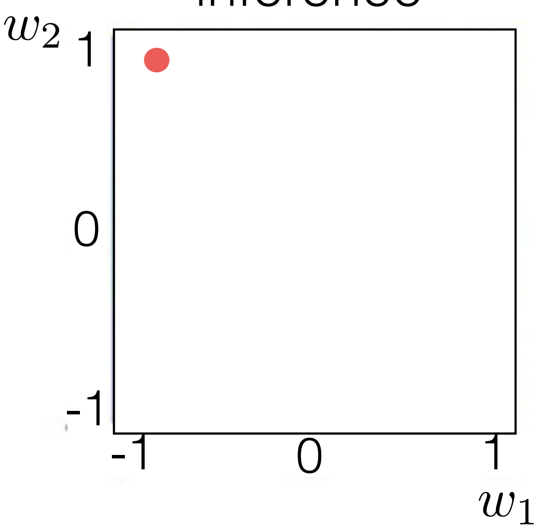


$$p(Y|X, w)p(w) \to \max_w$$

Image draft from C. Bishop. Pattern Recognition and Machine Learning

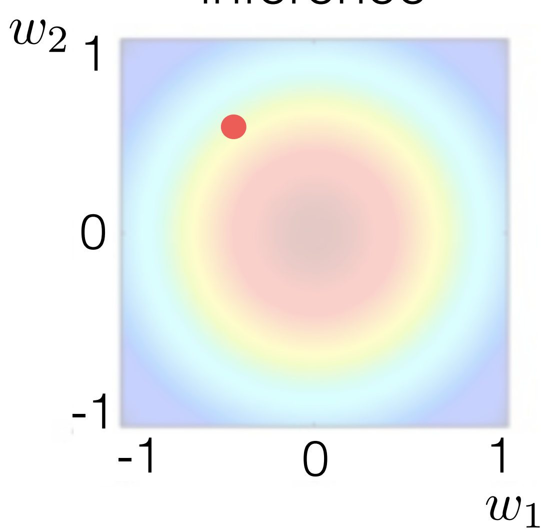
Options of training Bayesian linear regression

Maximum likelihood inference



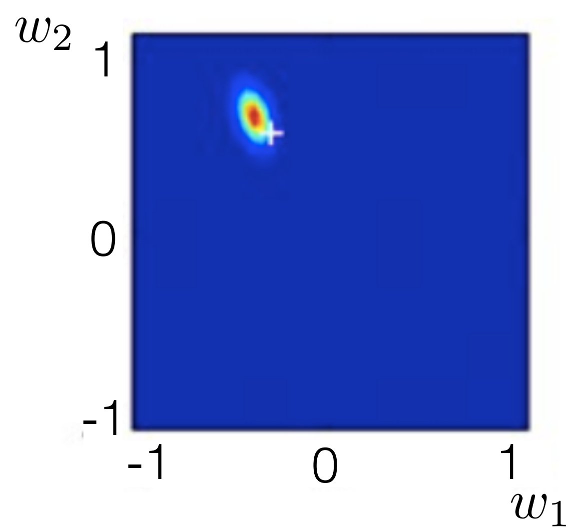
$$p(Y|X,w) \to \max_w$$

Maximum posterior inference



$$p(Y|X,w)p(w) \to \max_w$$

Full Bayesian inference: posterior distribution

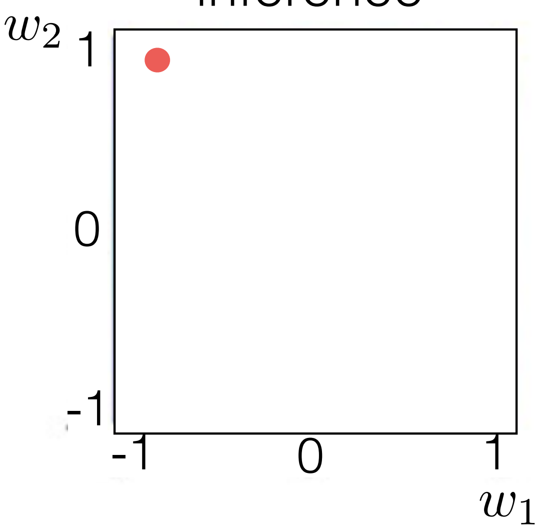


$$p(w|X,Y) \propto$$
$$\propto p(Y|X,w)p(w)$$

Image draft from C. Bishop. Pattern Recognition and Machine Learning

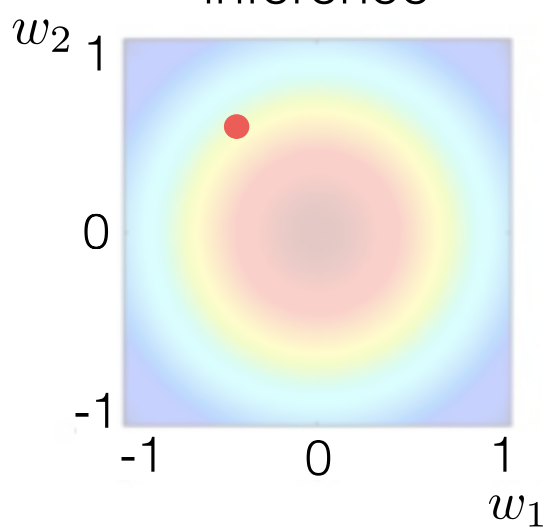
Options of training Bayesian linear regression

Maximum likelihood inference



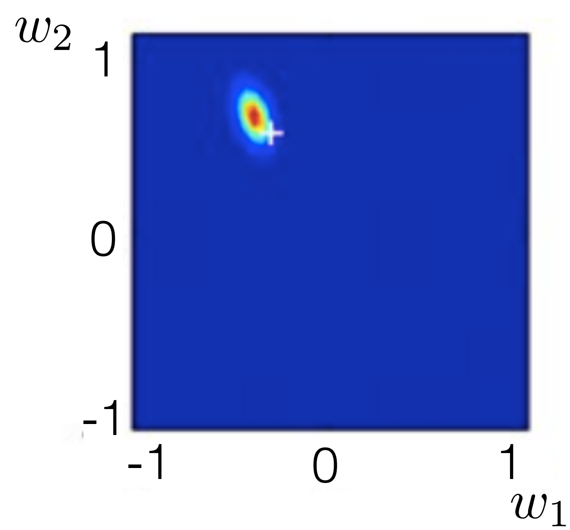
corresponds to conventional training of linear regression

Maximum posterior inference



corresponds to conventional training of **regularized** linear regression

Full Bayesian inference: posterior distribution



Bayesian linear regression: training

Full Bayesian inference: $p(w|X, Y)$

Likelihood and prior are conjugate $\longrightarrow$ posterior is normal

Bayesian linear regression: training

Full Bayesian inference: $p(w|X,Y)$

Likelihood: $p(Y|X,w) = \mathcal{N}(Y|Xw, I)$ Prior: $p(w) = \mathcal{N}(w|0, \alpha I),\ \alpha > 0$

Bayesian linear regression: training

Full Bayesian inference: $p(w|X,Y)$

Likelihood: $p(Y|X,w) = \mathcal{N}(Y|Xw, I)$ Prior: $p(w) = \mathcal{N}(w|0, \alpha I),\ \alpha > 0$

$$p(w|X,Y) \propto p(Y|X,w)p(w) \propto$$

$$\text{Const} \cdot \exp\left(-\frac{1}{2}(Y - Xw)^T(Y - Xw)\right)\exp\left(-\frac{1}{2\alpha}w^T w\right) =$$

Bayesian linear regression: training

Full Bayesian inference: $p(w|X,Y)$

Likelihood: $p(Y|X,w) = \mathcal{N}(Y|Xw, I)$ Prior: $p(w) = \mathcal{N}(w|0, \alpha I),\ \alpha > 0$

$$p(w|X,Y) \propto p(Y|X,w)p(w) \propto$$

$$\mathrm{Const}\cdot\exp\left(-\frac{1}{2}(Y - Xw)^T(Y - Xw)\right)\exp\left(-\frac{1}{2\alpha}w^T w\right) =$$

$$\mathrm{Const}\cdot\exp\underbrace{\left(-\frac{1}{2}w^T(X^T X + \frac{1}{\alpha}I)w + w^T X^T Y\right)}_{\text{quadratic form w.r.t weights}}$$

quadratic form w.r.t weights → **normal distribution**

Bayesian linear regression: training

Full Bayesian inference: $p(w|X,Y)$

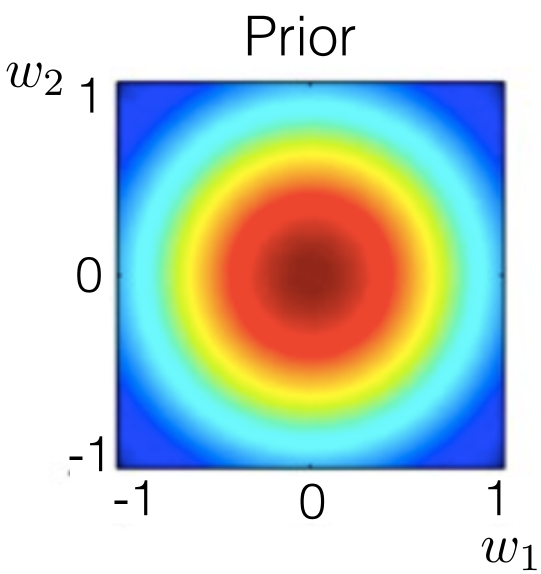
Likelihood: $p(Y|X,w) = \mathcal{N}(Y|Xw,I)$ Prior: $p(w) = \mathcal{N}(w|0,\alpha I),\ \alpha > 0$

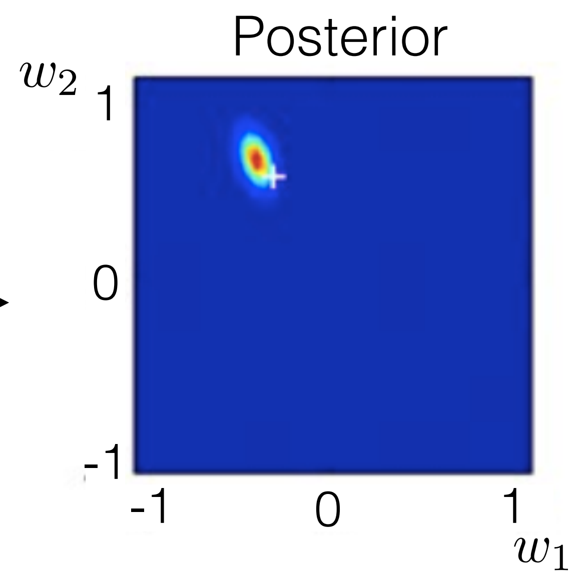
$$p(w|X,Y) = \mathcal{N}(w|w_{MP},\Sigma)$$

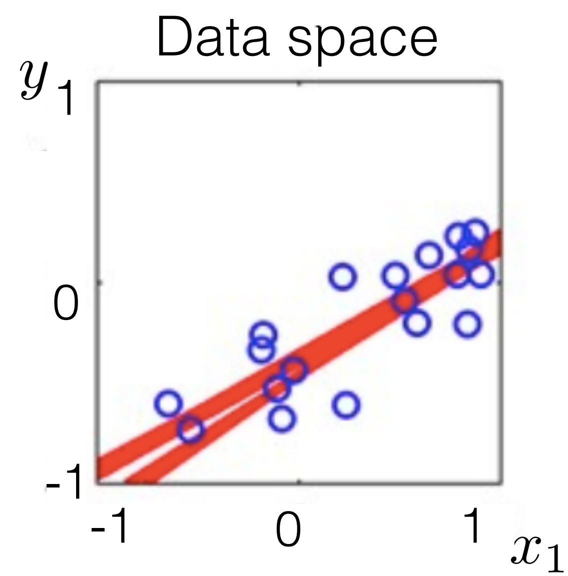
$$w_{MP} = (X^TX + \frac{1}{\alpha}I)^{-1}X^TY$$

$$\Sigma = X^TX + \frac{1}{\alpha}I$$

Training visualization







$$p(w) = \mathcal{N}(w|0, \alpha I)$$

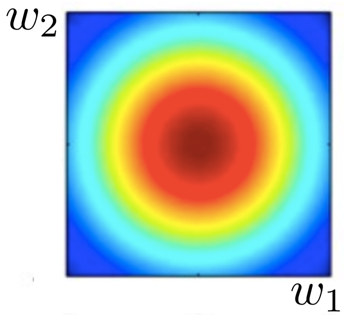
$$p(w|X,Y) = \mathcal{N}(w|w_{MP}, \Sigma)$$

$$p(y|x,w) = \mathcal{N}(y|w_1 + w_2 x_1, 1)$$

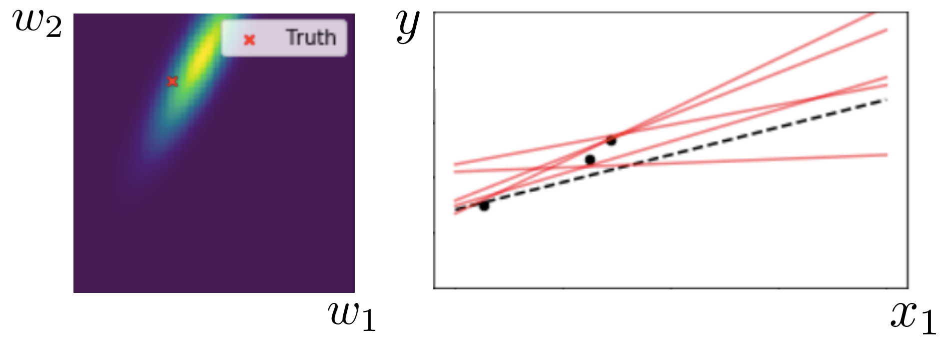
Image from C. Bishop. Pattern Recognition and Machine Learning

Training: increasing amount of data

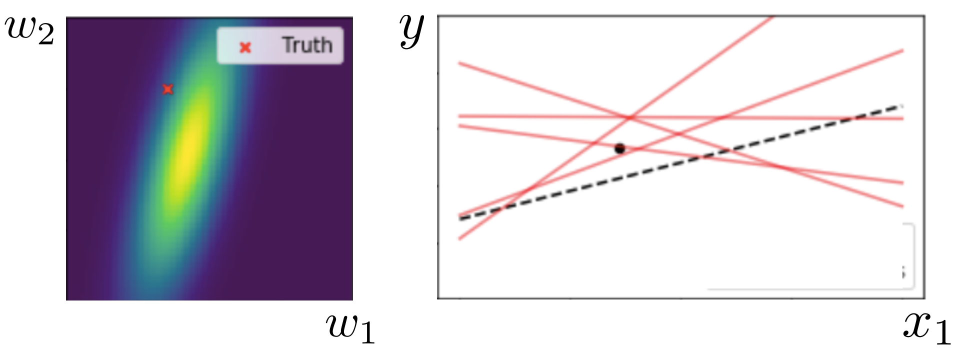
Prior:



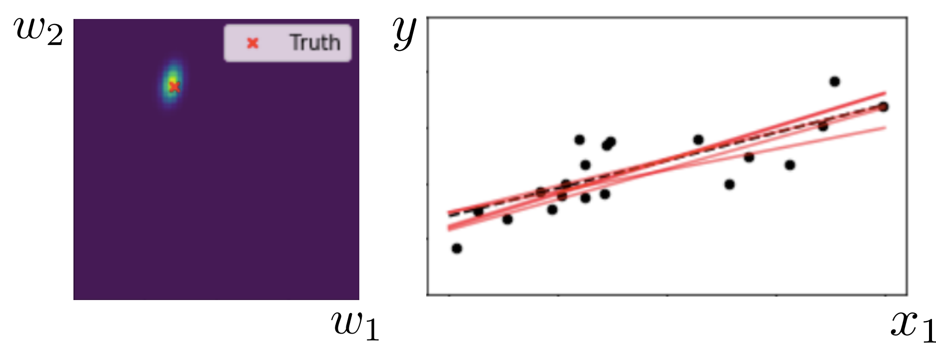
3 data points (N=3):



1 data point (N=1):



20 data points (N=20):



Images from http://krasserm.github.io/2019/02/23/bayesian-linear-regression/

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

Model:

$$p(Y, w|X) = p(Y|X, w)p(w) =$$
$$= \mathcal{N}(Y|Xw, I)\mathcal{N}(w|0, \alpha I)$$

Training:

$$p(w|X, Y) = \mathcal{N}(w|w_{MP}, \Sigma)$$

$$w_{MP} = (X^T X + \frac{1}{\alpha} I)^{-1} X^T Y$$

$$\Sigma = X^T X + \frac{1}{\alpha} I$$

Prediction?

Full Bayesian inference

Training stage:

$$p(w|X,Y) = \frac{p(Y|X,w)p(w)}{\int p(Y|X,\tilde{w})p(\tilde{w})d\tilde{w}}$$

Testing stage:

$$p(y_*|x_*,X,Y) = \int p(y_*|x_*,w)p(w|X,Y)dw = \mathbb{E}_{p(w|X,Y)}p(y_*|x_*,w)$$

x_* — new object

Modified slide from Ekaterina Lobacheva’s presentation

Bayesian linear regression: prediction

$$p(y_*|x_*, X, Y) = \int p(y_*|x_*, w)p(w|X, Y)dw =$$

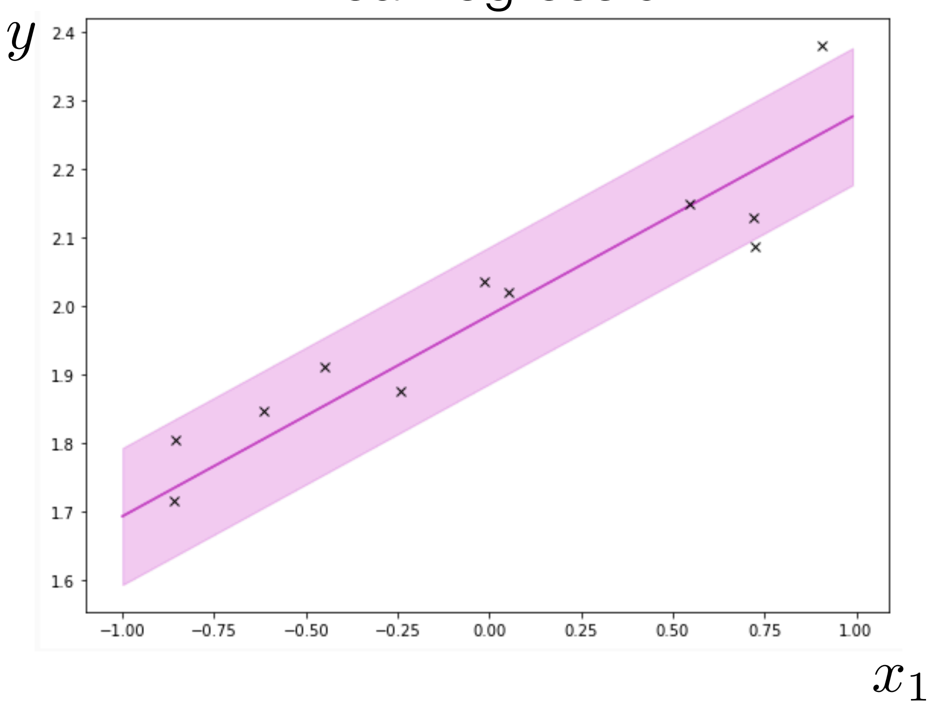
$$\int \mathcal{N}(y_*|x_*^T w, 1)\mathcal{N}(w|w_{MP}, \Sigma)dw =$$

$$\mathcal{N}(y_*|x_*^T w_{MP}, 1 + x_*^T \Sigma x_*)$$

x_* — new object

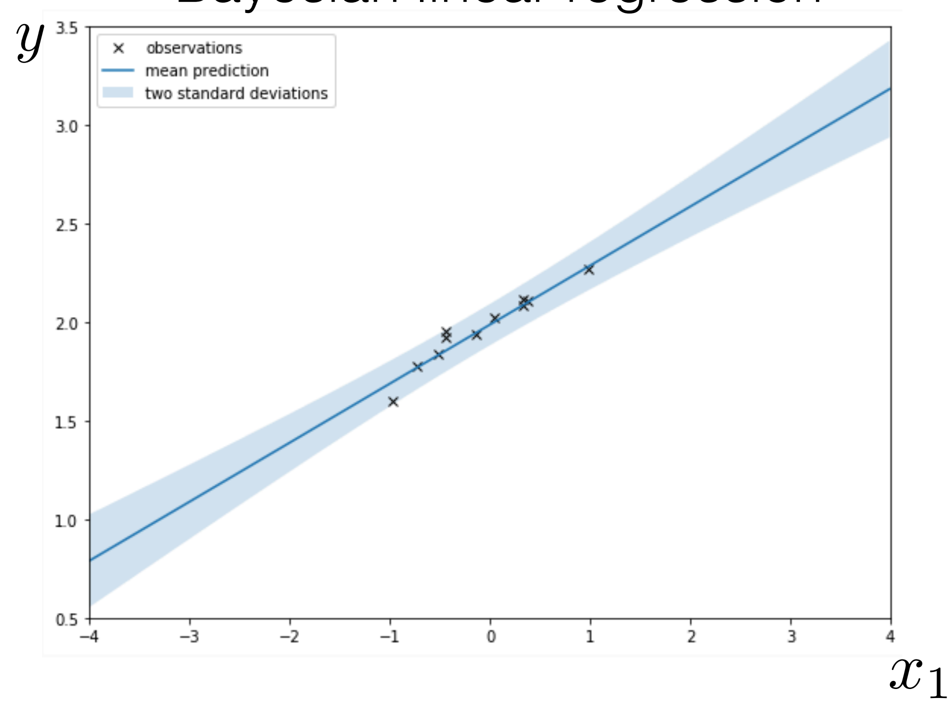
Prediction visualization

Linear regression



$$\mathcal{N}(y_* | x_*^T w_{MP}, 1)$$

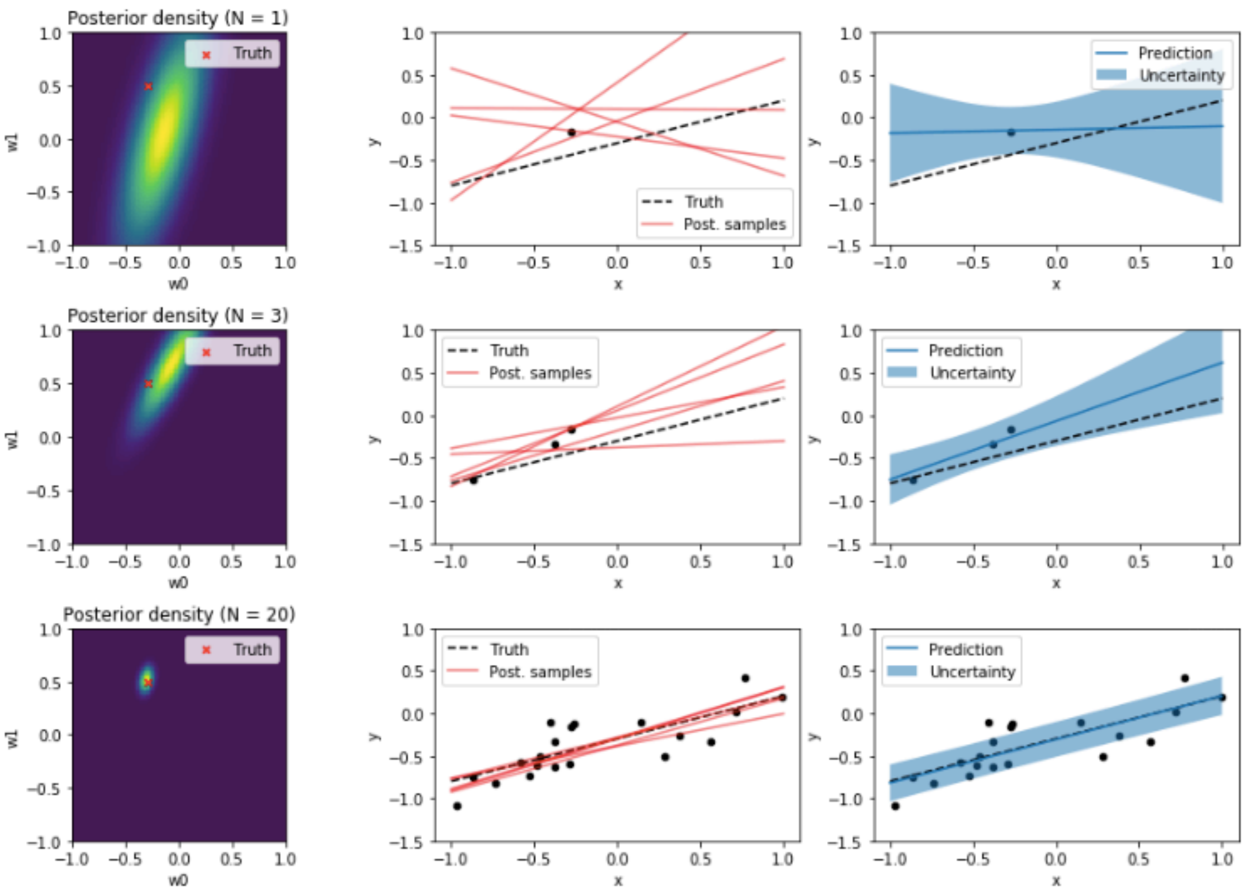
Bayesian linear regression



$$\mathcal{N}(y_* | x_*^T w_{MP}, 1 + x_*^T \Sigma x_*)$$

Image from https://jessicastringham.net/2018/01/10/bayesian-linreg-plots/

Prediction: increasing amount of data



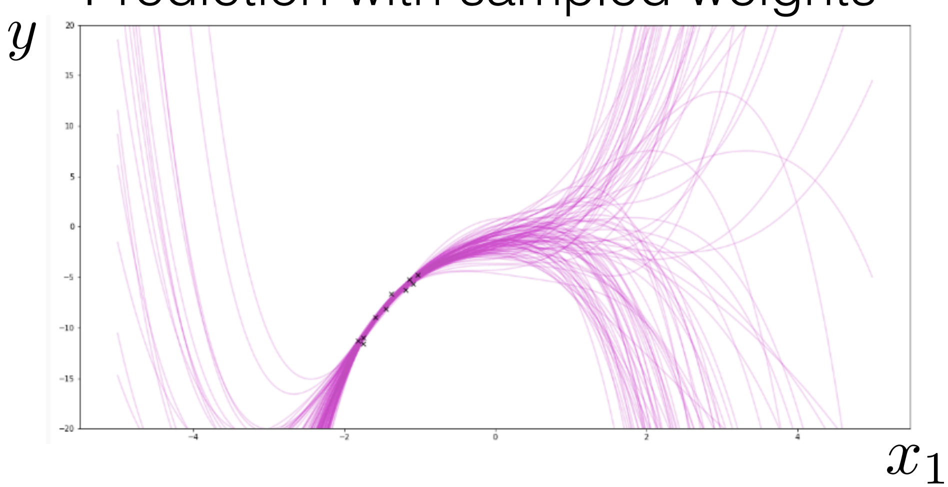
Images from http://krasserm.github.io/2019/02/23/bayesian-linear-regression/

Prediction: polynomial features

Modify training data: add polynomial features

$$p(y|x, w) = \mathcal{N}(y|w_1 + w_2 x_1 + w_3 x_1^2 + \ldots w_6 x_1^5, 1)$$

Prediction with sampled weights



Mean ± Std of predictions

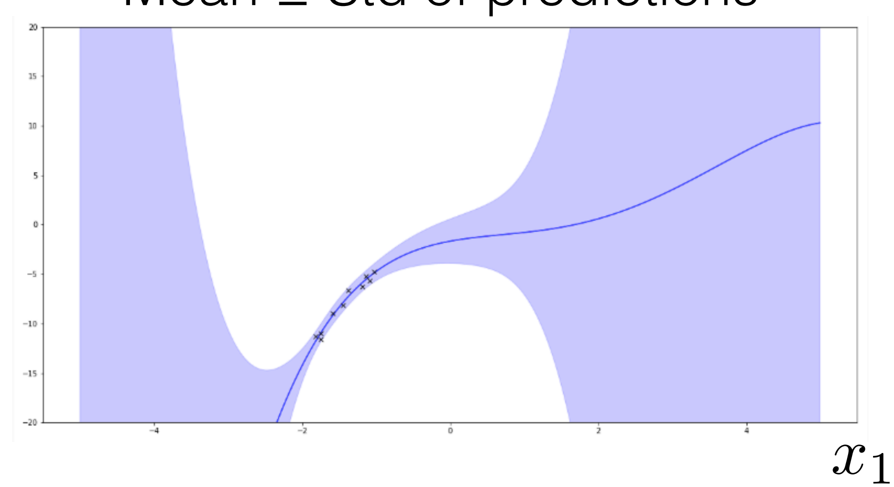


Image from https://jessicastringham.net/2018/01/10/bayesian-linreg-plots/

Bayesian linear regression

Given:

$X \in \mathbb{R}^{N \times d}$ — input data

$Y \in \mathbb{R}^{N}$ — target values

Model:

$$p(Y, w|X) = p(Y|X, w)p(w) =$$
$$= \mathcal{N}(Y|Xw, I)\mathcal{N}(w|0, \alpha I)$$

Training:

$$p(w|X, Y) = \mathcal{N}(w|w_{MP}, \Sigma)$$

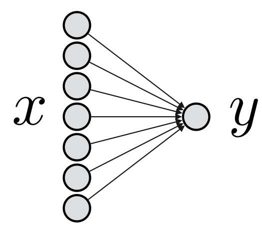
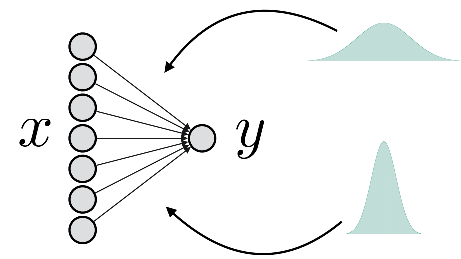
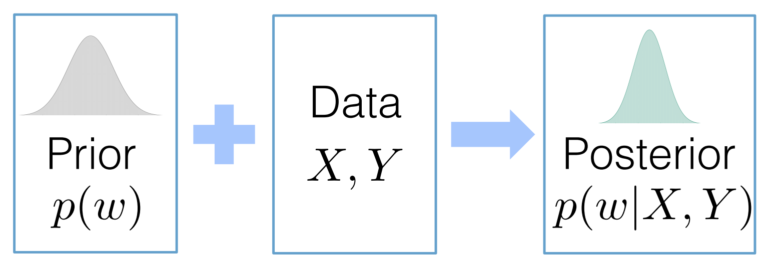
$$w_{MP} = (X^T X + \tfrac{1}{\alpha} I)^{-1} X^T Y$$

$$\Sigma = X^T X + \tfrac{1}{\alpha} I$$

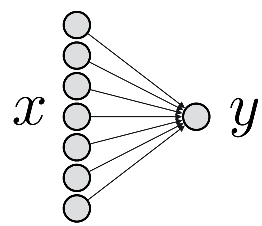
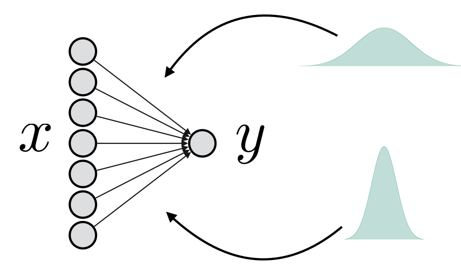
Prediction on a new object x_* :

$$p(y_*|x_*, X, Y) =$$
$$= \mathcal{N}(y_*|x_*^T w_{MP}, 1 + x_*^T \Sigma x_*)$$

Putting everything together

	Linear regression	Bayesian linear regression
Weights:	deterministic weights 	stochastic weights 
Training:	Data X, Y → point estimate w_{MP}	 Prior $p(w)$ + Data X, Y → Posterior $p(w\|X, Y)$
Prediction on a new object x_* :	$p(y_*\|x_*, w_{MP})$	$\mathbb{E}_{p(w\|X,Y)} p(y_*\|x_*, w)$

Putting everything together

	Linear regression	Bayesian linear regression
Weights:	deterministic weights 	stochastic weights 
Training:	$w_{MP} = (X^TX + \frac{1}{\alpha}I)^{-1}X^TY$	$p(w \mid X, Y) = \mathcal{N}(w \mid w_{MP}, \Sigma)$ $w_{MP} = (X^TX + \frac{1}{\alpha}I)^{-1}X^TY$ $\Sigma = X^TX + \frac{1}{\alpha}I$
Prediction on a new object x_* :	$\mathcal{N}(y_* \mid x_*^T w_{MP}, 1)$	$\mathcal{N}(y_* \mid x_*^T w_{MP}, 1 + x_*^T \Sigma x_*)$

Summary

- Conventional training of linear regression is equivalent to ML / MP Bayesian inference
- We can perform full Bayesian inference for linear regression, and obtain weight variance and covariance, in addition to mean values
- Bayesian regression provides more informative predictive uncertainty