Learning as Inference

TTT4185 Machine Learning for Signal Processing

Giampiero Salvi

Department of Electronic Systems
NTNU

HT2021

Outline

1. Learning as Inference
2. Point Estimates
   - Maximum Likelihood Estimation
   - Maximum a Posteriori Estimation
3. Bayesian Methods
4. Curse of Dimensionality

Outline

Probabilistic Classification and Regression

- In both cases estimate posterior

$$P(t \,|\, \mathbf{x}) = \frac{P(\mathbf{x} \,|\, t) P(t)}{P(\mathbf{x})}$$

- Classification: t is discrete
- Regression: t is continuous

Probabilistic Classification and Regression

- In both cases estimate posterior

$$P(t \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid t)P(t)}{P(\mathbf{x})}$$

- Classification: t is discrete
- Regression: t is continuous

Until now we assumed we knew:

- $P(t) \leftarrow$ *Prior*
- $P(\mathbf{x} \mid t) \leftarrow$ *Likelihood*
- $P(\mathbf{x}) \leftarrow$ *Evidence*

Probabilistic Classification and Regression

- In both cases estimate posterior

$$P(t \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid t)P(t)}{P(\mathbf{x})}$$

- Classification: t is discrete
- Regression: t is continuous

Until now we assumed we knew:

- $P(t) \leftarrow$ *Prior*
- $P(\mathbf{x} \mid t) \leftarrow$ *Likelihood*
- $P(\mathbf{x}) \leftarrow$ *Evidence*

How can we obtain this information from observations (data)?

Learning as Inference

Given:

- the training data $\mathcal{D} = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$
- a new observation $\mathbf{x}$

Estimate the posterior probability of the answer t:

$$P(t|\mathbf{x}, \mathcal{D})$$

Discriminative vs Generative Models

Discriminative:

- learn the posterior $P(t|\mathbf{x}, \mathcal{D})$ directly
- examples: linear regression, logistic regression

Generative:

- learn a model of data generation: priors $P(t|\mathcal{D})$ and likelihoods $P(\mathbf{x}|t, \mathcal{D})$
- use Bayes rule to obtain posterior $P(t|\mathbf{x}, \mathcal{D})$
- example: classification

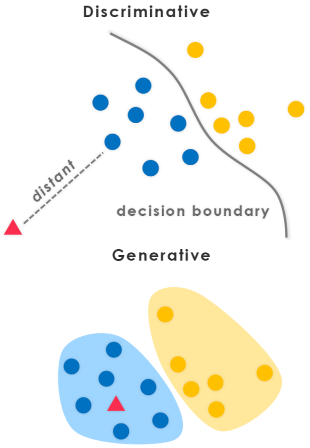


Figure from Nguyen *et al.*

Parametric vs Non-parametric Inference

Parametric:

- First make the model parameters explicit: $P(t|\mathbf{x}) = P(t|\mathbf{x}, \theta)$
- estimate the optimal parameters $\hat{\theta}$ using the data (point estimate)
- compute the posterior $P(t|\mathbf{x}, \hat{\theta})$

Learning corresponds to finding $\hat{\theta}$

Parametric vs Non-parametric Inference

Parametric:

- First make the model parameters explicit: $P(t|\mathbf{x}) = P(t|\mathbf{x}, \theta)$
- estimate the optimal parameters $\hat{\theta}$ using the data (point estimate)
- compute the posterior $P(t|\mathbf{x}, \hat{\theta})$

Learning corresponds to finding $\hat{\theta}$

Non-Parametric:

- Use a parametric model as before: $P(t|\mathbf{x}) = P(t|\mathbf{x}, \theta)$
- but estimate the posterior of the parameters given the data: $P(\theta|\mathcal{D})$
- Compute the posterior $P(t|\mathbf{x}, \mathcal{D})$ by marginalizing out the parameters θ

The number of parameters can grow with the data!

Three Approaches

Parametric:

- Maximum Likelihood (ML)
- Maximum A Posteriori (MAP)

Non-parametric:

- Bayesian methods

Fundamental Assumption: i.i.d.

Observations are independent and identically distributed:

$$\mathcal{D} = \{\mathbf{o}_1, \dots, \mathbf{o}_N\}$$

The likelihood of the whole data set can be factorized:

$$P(\mathcal{D}) = P(\mathbf{o}_1, \dots, \mathbf{o}_N) = \prod_{i=1}^{N} P(\mathbf{o}_i)$$

And the log-likelihood becomes:

$$\log P(\mathcal{D}) = \sum_{i=1}^{N} \log P(\mathbf{o}_i)$$

Outline

Maximum Likelihood Estimate

- define parametric form for the distributions:

$$P(\mathbf{x}|t) \equiv P(\mathbf{x}|t,\theta) \quad \text{or} \quad P(t|\mathbf{x}) \equiv P(t|\mathbf{x},\theta)$$

- find optimal value for the parameter θ_{ML} by maximizing the likelihood of the data:

$$\theta_{\text{ML}} = \arg\max_{\theta} P(\mathcal{D}|\theta)$$

- approximate the distribution given the data with this distribution:

$$P(\mathbf{x}|t,\mathcal{D}) \approx P(\mathbf{x}|t,\theta_{\text{ML}}) \quad \text{or} \quad P(t|\mathbf{x},\mathcal{D}) \approx P(t|\mathbf{x},\theta_{\text{ML}})$$

Parameter Estimation vs Decision Theory

Decision theory:

- $\mathbf{x}$ and θ are know
- maximize likelihood or posterior to find t

Parameter Estimation:

- $\mathbf{x}$ and t are know (supervised learning)
- maximize likelihood or posterior to find θ

Parameter Estimation vs Decision Theory

Decision theory:

- $\mathbf{x}$ and θ are know
- maximize likelihood or posterior to find t

Parameter Estimation:

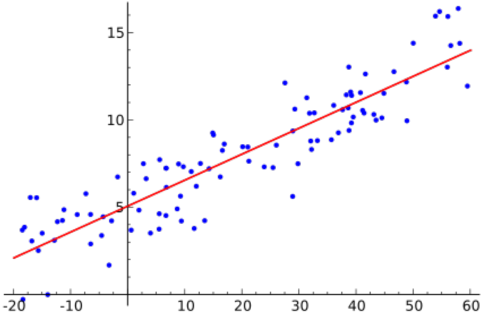
- $\mathbf{x}$ and t are know (supervised learning)
- maximize likelihood or posterior to find θ

Same models and same kind of optimization

Classical Linear Regression

Model (deterministic):

$$
\begin{aligned}
\hat{t} = y(\mathbf{x}, \mathbf{w}) &= w_0 + w_1 x_1 + \dots + w_d x_d \\
&= \begin{bmatrix} w_0 & w_1 & \dots & w_d \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \\
&= \mathbf{w}^T \mathbf{x}
\end{aligned}
$$



Minimize sum of square errors

$$
\mathbf{w}_{\text{opt}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2 = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} (t_i - \mathbf{w}^T \mathbf{x}_i)^2
$$

Probabilistic Linear Regression

Model (deterministic):

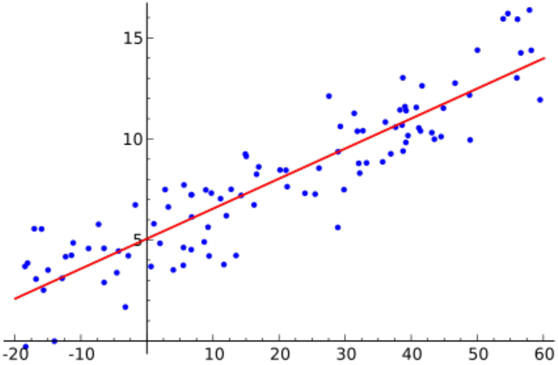
$$\hat{t} = y(\mathbf{x}, \mathbf{w}) + \epsilon = \mathbf{w}^T \mathbf{x} + \epsilon$$

But now:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Therefore:

$$\begin{aligned} t &\sim \mathcal{N}(\mu_T(\mathbf{x}), \sigma_T^2(\mathbf{x})) \\ &= \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma^2) \end{aligned}$$



Learning: find $\mathbf{w}$ that maximizes $P(T|X, \mathbf{w}, \sigma^2)$

Maximize the posterior directly $\implies$ discriminative method

MLE for Probabilistic Linear Regression

$$\log p(T|X, \mathbf{w}, \sigma^2) \quad = \quad \log \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2)$$

MLE for Probabilistic Linear Regression

$$\begin{aligned}
\log p(T|X, \mathbf{w}, \sigma^2) &= \log \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2)
\end{aligned}$$

MLE for Probabilistic Linear Regression

$$
\begin{aligned}
\log p(T|X, \mathbf{w}, \sigma^2) &= \log \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t_i - \mathbf{w}^T \mathbf{x}_i)^2}{2\sigma^2}} \right]
\end{aligned}
$$

MLE for Probabilistic Linear Regression

$$
\begin{aligned}
\log p(T|X, \mathbf{w}, \sigma^2) &= \log \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t_i - \mathbf{w}^T\mathbf{x}_i)^2}{2\sigma^2}} \right] \\
&= \sum_i \left[-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(t_i - \mathbf{w}^T\mathbf{x}_i)^2}{2\sigma^2} \right]
\end{aligned}
$$

MLE for Probabilistic Linear Regression

$$
\begin{aligned}
\log p(T|X, \mathbf{w}, \sigma^2) &= \log \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t_i - \mathbf{w}^T \mathbf{x}_i)^2}{2\sigma^2}} \right] \\
&= \sum_i \left[-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(t_i - \mathbf{w}^T \mathbf{x}_i)^2}{2\sigma^2} \right]
\end{aligned}
$$

$$
\arg\max_{\mathbf{w}} \left[p(T|X, \mathbf{w}, \sigma^2) \right] = \arg\min_{\mathbf{w}} \sum_i (t_i - \mathbf{w}^T \mathbf{x}_i)^2
$$

Maximizing $p(T|X, \mathbf{w}, \sigma^2)$ equivalent to minimizing sum of squares!

Source of confusion

We did Maximum a Posteriori (MAP) regression

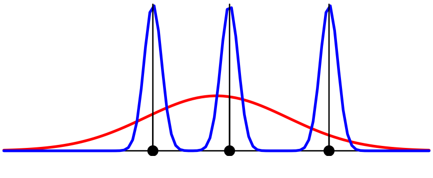
$$t_{\text{MAP}} = \arg\max_t p(t|\mathbf{x}, \theta_{\text{ML}})$$

with parameters θ estimated by Maximum Likelihood (ML):

$$\theta_{\text{ML}} = \arg\max_\theta p(D|\theta) = \arg\max_\theta \prod_i p(\mathbf{x}_i|t_i, \theta)$$

ML and overfitting

- same solution as sum of squares
- $\Rightarrow$ same problems with overfitting
- we would like regularization



Maximum a posteriori

- assume that parameter θ is stochastic variable
- define a prior distribution over θ
- maximize posterior $P(\theta|\mathcal{D})$ over the parameter

Maximum a Posteriori Estimation

$$\theta_{\mathsf{MAP}} = \arg\max_{\theta} p(\theta|\mathcal{D})$$

Maximum a Posteriori Estimation

$$\begin{aligned} \theta_{\text{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\ &= \arg\max_{\theta} \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \end{aligned}$$

Maximum a Posteriori Estimation

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\ &= \arg\max_{\theta} \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \\ &= \arg\max_{\theta} p(\theta)p(\mathcal{D}|\theta)\end{aligned}$$

Maximum a Posteriori Estimation

$$
\begin{aligned}
\theta_{\mathsf{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \\
&= \arg\max_{\theta} p(\theta)p(\mathcal{D}|\theta) \\
&= \arg\max_{\theta} \left[p(\theta) \prod_{i=1}^{N} p(\mathbf{o}_i|\theta) \right]
\end{aligned}
$$

Maximum a Posteriori Estimation

$$
\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \\
&= \arg\max_{\theta} p(\theta)p(\mathcal{D}|\theta) \\
&= \arg\max_{\theta} \left[p(\theta) \prod_{i=1}^{N} p(\mathbf{o}_i|\theta) \right] \\
&= \arg\max_{\theta} \left[\log p(\theta) + \sum_{i=1}^{N} \log p(\mathbf{o}_i|\theta) \right]
\end{aligned}
$$

Maximum a Posteriori Estimation

$$
\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \\
&= \arg\max_{\theta} p(\theta)p(\mathcal{D}|\theta) \\
&= \arg\max_{\theta} \left[p(\theta) \prod_{i=1}^{N} p(\mathbf{o}_i|\theta) \right] \\
&= \arg\max_{\theta} \left[\log p(\theta) + \sum_{i=1}^{N} \log p(\mathbf{o}_i|\theta) \right]
\end{aligned}
$$

- $\log p(\theta)$ works as regularization

MAP for Linear Regression

Model (deterministic):

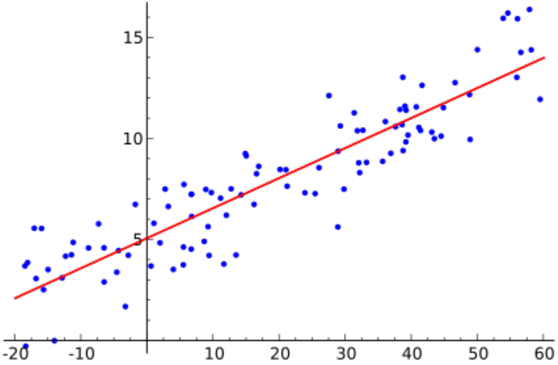
$$t = \mathbf{w}^T\mathbf{x} + \epsilon$$

With:

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

Therefore:

$$t \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma_\epsilon^2)$$



MAP for Linear Regression

Model (deterministic):

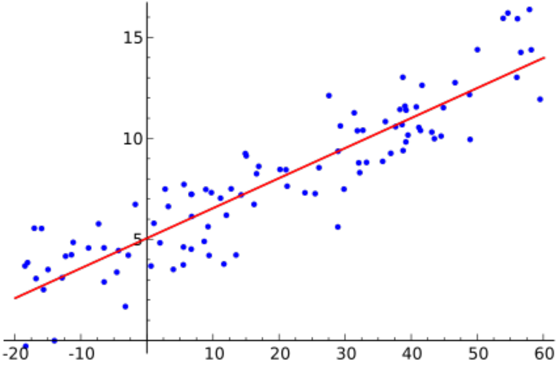
$$t = \mathbf{w}^T \mathbf{x} + \epsilon$$

With:

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

Therefore:

$$t \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma_\epsilon^2)$$



But now we define the a priori probability over $\mathbf{w}$: $p(\mathbf{w})$

Example: zero-mean spherical Gaussian prior

Example: zero-mean spherical Gaussian on $\mathbf{w} = [w_0, \ldots, w_{d-1}]$

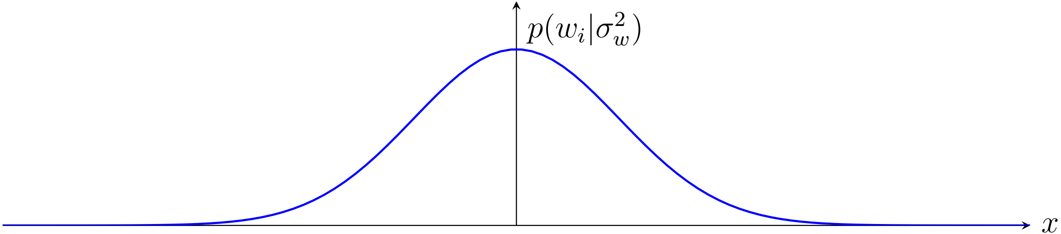
$$p(\mathbf{w}|\sigma_w^2) = \mathcal{N}(0, \sigma_w^2\mathbf{I}) = \frac{1}{(2\pi\sigma_w^2)^{\frac{d}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_w^2}\right)$$

Example: zero-mean spherical Gaussian prior

Example: zero-mean spherical Gaussian on $\mathbf{w} = [w_0, \dots, w_{d-1}]$

$$p(\mathbf{w}|\sigma_w^2) = \mathcal{N}(0, \sigma_w^2 \mathbf{I}) = \frac{1}{(2\pi\sigma_w^2)^{\frac{d}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_w^2}\right) =$$

$$= \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{w_i^2}{2\sigma_w^2}\right)$$



MAP estimate with zero-mean spherical Gaussian prior

Instead of $\log p(T|X,\mathbf{w})$ as in MLE, we optimize $\log p(\mathbf{w}|T,X)$:

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|T,X) = \arg\max_{\mathbf{w}} \log \left[p(T|X,\mathbf{w})p(\mathbf{w})\right]$$

MAP estimate with zero-mean spherical Gaussian prior

Instead of $\log p(T|X, \mathbf{w})$ as in MLE, we optimize $\log p(\mathbf{w}|T, X)$:

$$
\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log p(\mathbf{w}|T, X) = \arg\max_{\mathbf{w}} \log \left[p(T|X, \mathbf{w}) p(\mathbf{w}) \right] \\
\ldots &= \arg\max_{\mathbf{w}} \left[\sum_n \log p(t_n|\mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w}) \right] =
\end{aligned}
$$

MAP estimate with zero-mean spherical Gaussian prior

Instead of $\log p(T|X, \mathbf{w})$ as in MLE, we optimize $\log p(\mathbf{w}|T, X)$:

$$
\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log p(\mathbf{w}|T, X) = \arg\max_{\mathbf{w}} \log\left[p(T|X, \mathbf{w})p(\mathbf{w})\right] \\
\dots &= \arg\max_{\mathbf{w}} \left[\sum_n \log p(t_n|\mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w})\right] = \\
\dots &= \arg\min_{\mathbf{w}} \Bigg[\underbrace{\sum_n \left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit to the data (ML)}} + \underbrace{\frac{\sigma_\epsilon^2}{\sigma_w^2}\mathbf{w}^T\mathbf{w}}_{\text{keep } \mathbf{w} \text{ simple}}\Bigg]
\end{aligned}
$$

MAP estimate with zero-mean spherical Gaussian prior

Instead of $\log p(T|X, \mathbf{w})$ as in MLE, we optimize $\log p(\mathbf{w}|T, X)$:

$$
\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log p(\mathbf{w}|T, X) = \arg\max_{\mathbf{w}} \log\left[p(T|X, \mathbf{w})p(\mathbf{w})\right] \\
\ldots &= \arg\max_{\mathbf{w}} \left[\sum_n \log p(t_n|\mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w})\right] = \\
\ldots &= \arg\min_{\mathbf{w}} \left[\underbrace{\sum_n \left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit to the data (ML)}} + \underbrace{\frac{\sigma_\epsilon^2}{\sigma_w^2}\mathbf{w}^T\mathbf{w}}_{\text{keep } \mathbf{w} \text{ simple}}\right]
\end{aligned}
$$

Equivalent to ridge regression with $\lambda = \frac{\sigma_\epsilon^2}{\sigma_w^2}$

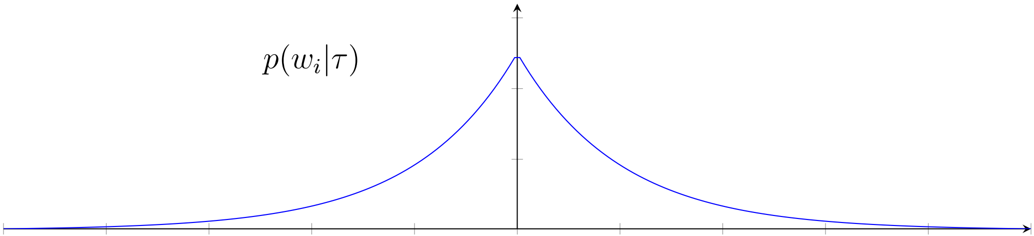
Example: Prior for LASSO

- LASSO: Least Absolute Shrinkage and Selection Operator
- We want the regularization to be $\lambda \sum_i |w_i|$ instead of $\lambda \sum_i w_i^2$.

Example: Prior for LASSO

- LASSO: Least Absolute Shrinkage and Selection Operator
- We want the regularization to be $\lambda \sum_i |w_i|$ instead of $\lambda \sum_i w_i^2$.
- Following the same arguments as before, we will need a product of zero-mean Laplace priors:

$$p(\mathbf{w}|\tau) = \prod_i \text{Laplace}(w_i, 0, \tau) = \prod_i \frac{1}{2\tau} \exp\left(-\frac{|w_i|}{\tau}\right)$$



Conjugate Prior

Definition:

if posterior and prior in the same family of functions

Examples:

Likelihood	Conjugate prior
Bernoulli	Beta
Binomial	Beta
Categorical	Dirichlet
Normal	Normal
Normal	Normal-inverse Gamma

Conjugate Priors and Iterative learning

- we start with prior $p(\theta)$
- we use a data set $\mathcal{D}_1$ to estimate posterior $p(\theta|\mathcal{D}_1)$

If new data $\mathcal{D}_2$ becomes available:

- we can use $p(\theta|\mathcal{D}_1)$ as prior
- and use $\mathcal{D}_2$ to estimate new posterior $p(\theta|\mathcal{D}_1, \mathcal{D}_2)$

Conjugate Priors and Iterative learning

- we start with prior $p(\theta)$
- we use a data set $\mathcal{D}_1$ to estimate posterior $p(\theta|\mathcal{D}_1)$

If new data $\mathcal{D}_2$ becomes available:

- we can use $p(\theta|\mathcal{D}_1)$ as prior
- and use $\mathcal{D}_2$ to estimate new posterior $p(\theta|\mathcal{D}_1, \mathcal{D}_2)$

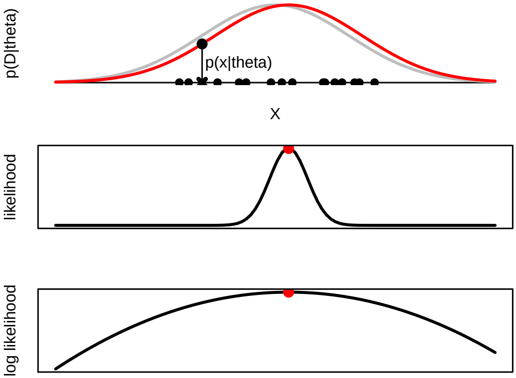
Notes:

- It is simple because $p(\theta|\mathcal{D}_1)$ has the same shape as $p(\theta)$
- we need to keep the whole posterior, not only point estimate θ_{MAP}

Outline

ML, MAP and Point Estimates

- Both ML and MAP produce point estimates of θ
- Assumption: there is a true value for θ
- advantage: once $\hat{\theta}$ is found, everything is known



Limitations of MAP Estimate

- shift problem to defining the parameters of the prior (λ in Ridge and LASSO regression)
- uncertainty in the posterior $p(t|\mathbf{x}, \mathbf{w}_{\text{OPT}})$ is still σ_ϵ^2 and is independent of $\mathbf{x}$

Bayesian estimation (non-parametric models)

$$
\begin{array}{lllll}
\text{ML:} & \mathcal{D} & \rightarrow & \theta_{\text{ML}} & \rightarrow & P(\mathbf{o}_{\text{new}}|\theta_{\text{ML}}) \\
\text{MAP:} & \mathcal{D}, P(\theta) & \rightarrow & \theta_{\text{MAP}} & \rightarrow & P(\mathbf{o}_{\text{new}}|\theta_{\text{MAP}}) \\
\text{Bayes:} & \mathcal{D}, P(\theta) & \rightarrow & P(\theta|\mathcal{D}) & \rightarrow & P(\mathbf{o}_{\text{new}}|\mathcal{D})
\end{array}
$$

1. consider θ as a random variable (same as MAP)
2. characterize θ with the posterior distribution $P(\theta|\mathcal{D})$ given the data
3. compute new predictive posterior $P(\mathbf{o}_{\text{new}}|\mathcal{D})$ marginalizing over θ (predictive posterior)

$$
P(\mathbf{o}_{\text{new}}|\mathcal{D}) = \int_{\theta \in \Theta} P(\mathbf{o}_{\text{new}}|\theta) P(\theta|\mathcal{D}) d\theta
$$

Bayesian Linear Regression

Setup:

$$\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$$

Model (same as MAP):

- $t_1, \dots, t_n$ independent given $\mathbf{w}$
- $t_i \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}_i, \sigma_\epsilon^2)$
- $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}), \quad \mathbf{w} = \{w_0, w_1, \dots, w_d\}$
- we assume σ_ϵ^2 and σ_w^2 are know: $\theta = \{\mathbf{w}\}$

Goal:

Estimate $p(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathcal{D})$

Bayesian Linear Regression

$$
\begin{aligned}
p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D}) &= \int_{\mathbf{w}\in W} p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D}, \mathbf{w})p(\mathbf{w}|\mathbf{x}_{\text{new}}, \mathcal{D})d\mathbf{w} \\
&= \int_{\mathbf{w}\in W} p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}
\end{aligned}
$$

Results obtained with many passages:

- if prior $p(\mathbf{w})$ is Gaussian, then posterior $p(\mathbf{w}|\mathcal{D})$ is still Gaussian
- because the likelihood $p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{w})$ is Gaussian, the predictive posterior $p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D})$ is Gaussian as well.
- all the results can be obtained in closed form (in this case)

Complete Derivations

From mathematicalmonk's YouTube channel:

- problem and model definition
  `https://youtu.be/1WvnpjljKXA`
- posterior $p(\mathbf{w}|\mathcal{D})$, part 1–2
  `https://youtu.be/nrd4AnDLR3U`
  `https://youtu.be/qz2U8coNwV4`
- predictive posterior $p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D})$, part 1–3
  `https://youtu.be/xyuSiKXttxw`
  `https://youtu.be/vTcsacTqlfQ`
  `https://youtu.be/LCISTY9S6SQ`

Closed Form Solutions

Posterior $p(\mathbf{w}|\mathcal{D}) = \mathcal{N}(\mu, \mathbf{\Sigma})$, with:

$$\begin{aligned}\mathbf{\Sigma} &= \frac{1}{\sigma_\epsilon^2} X^T X + \frac{1}{\sigma_w^2}\mathbf{I} \\ \mu &= \frac{1}{\sigma_\epsilon^2}\mathbf{\Sigma}^{-1} X^T T\end{aligned}$$

Closed Form Solutions

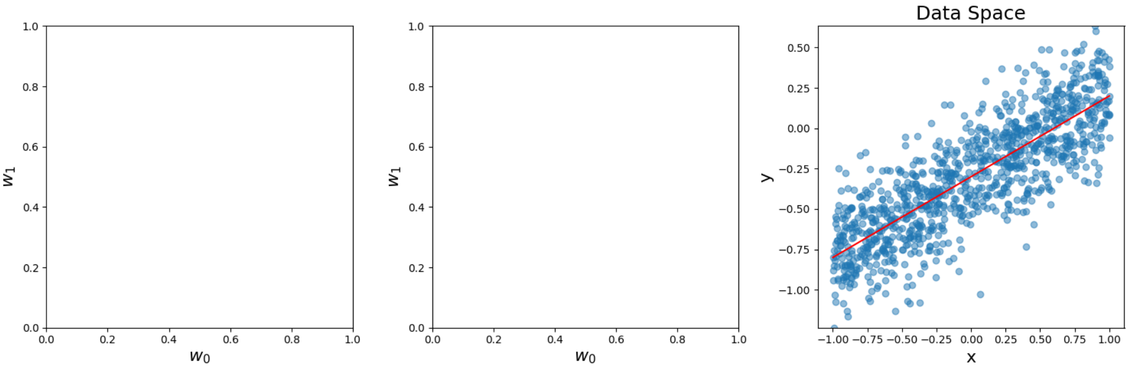
Posterior $p(\mathbf{w}|\mathcal{D}) = \mathcal{N}(\mu, \mathbf{\Sigma})$, with:

$$\begin{aligned} \mathbf{\Sigma} &= \frac{1}{\sigma_\epsilon^2} X^T X + \frac{1}{\sigma_w^2} \mathbf{I} \\ \mu &= \frac{1}{\sigma_\epsilon^2} \mathbf{\Sigma}^{-1} X^T T \end{aligned}$$

Predictive posterior

$$p(t_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D}) = \mathcal{N}(\mu^T \mathbf{x}_{\text{new}}, \sigma_\epsilon^2 + \mathbf{x}_{\text{new}}^T \mathbf{\Sigma} \mathbf{x}_{\text{new}})$$

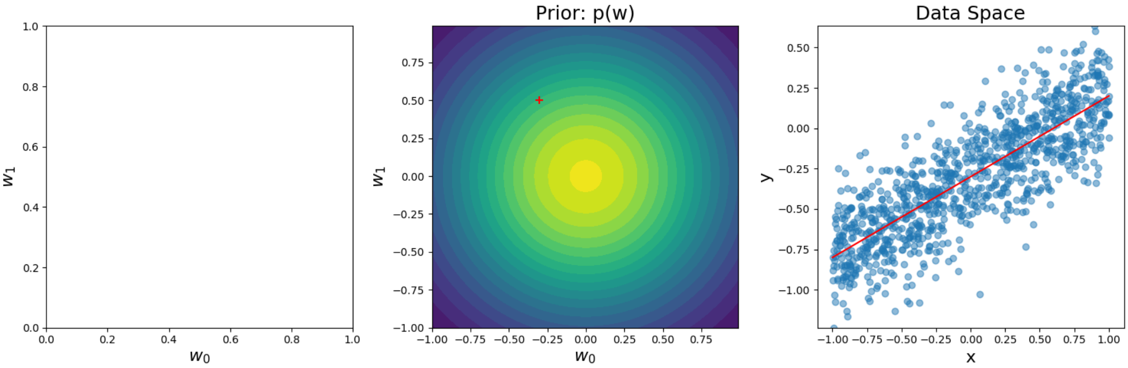
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

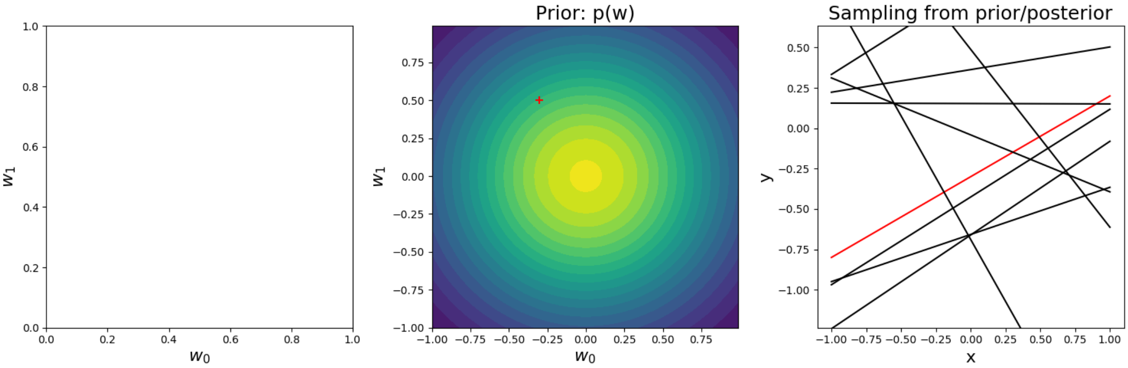
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

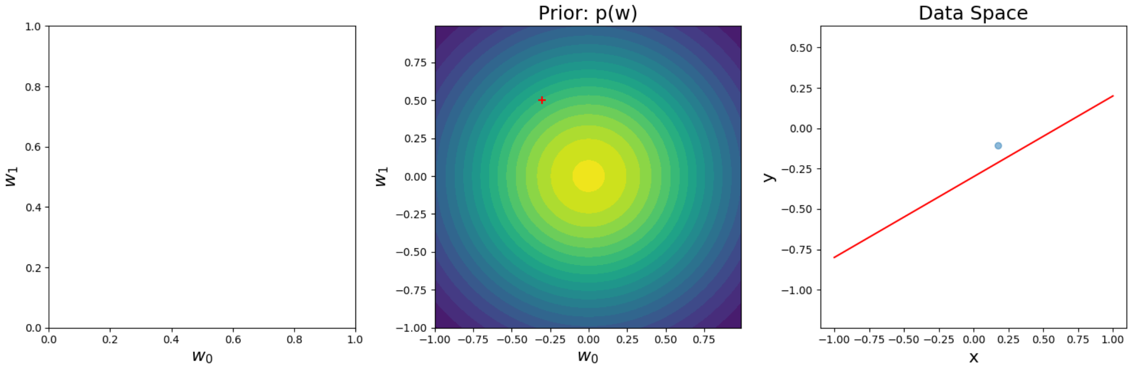
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

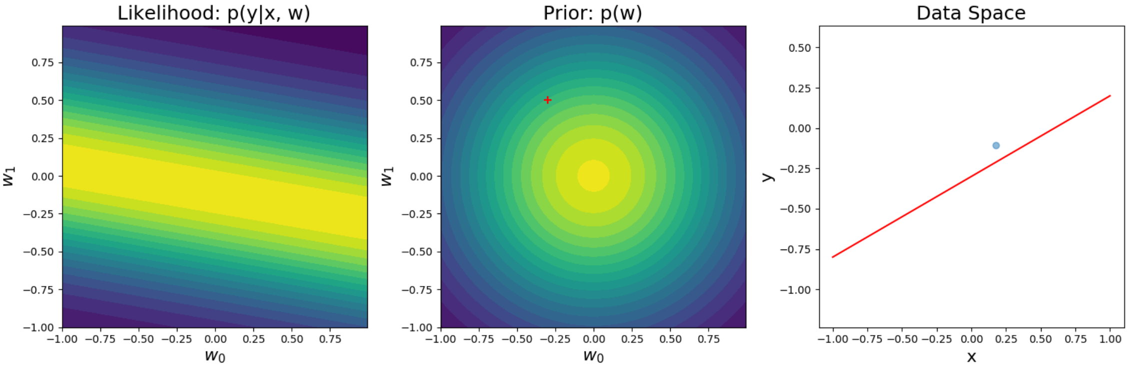
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

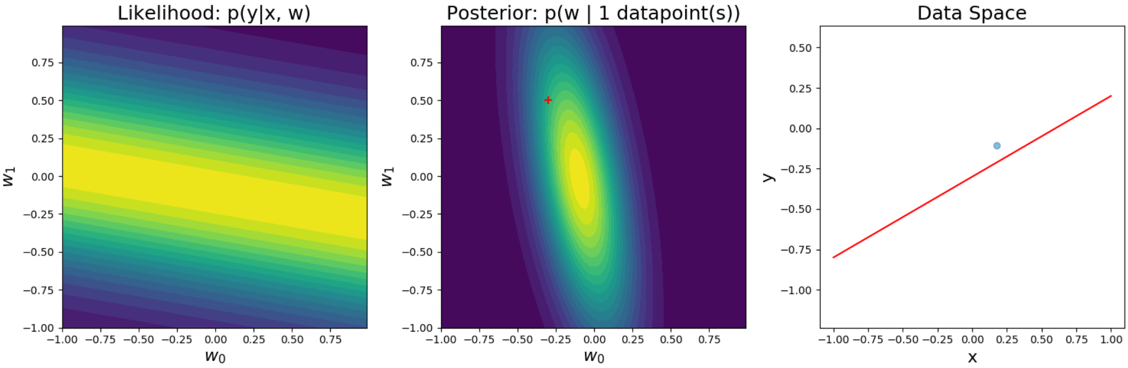
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

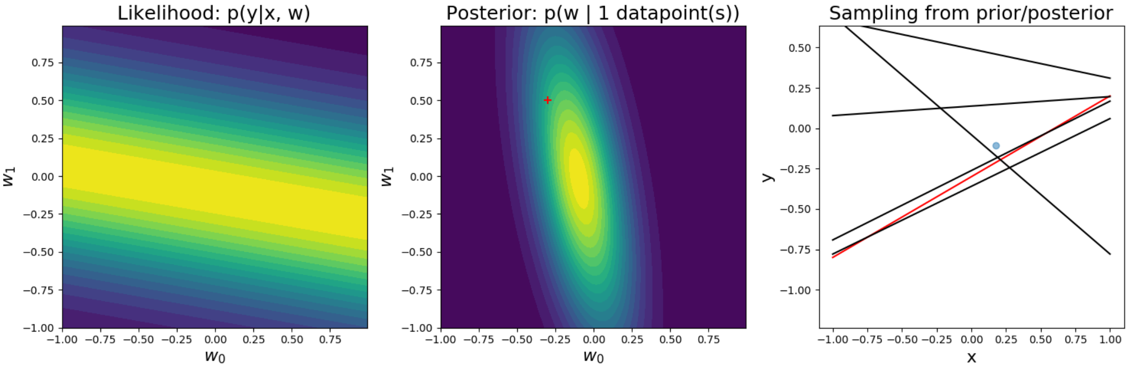
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

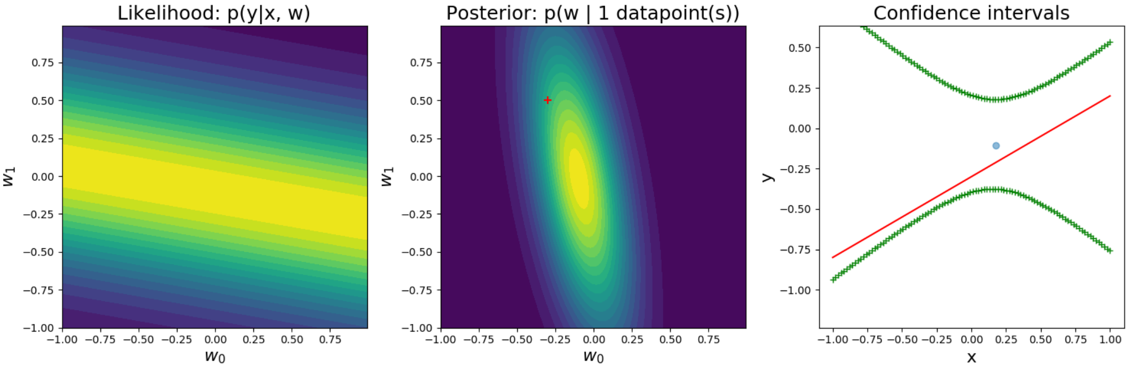
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

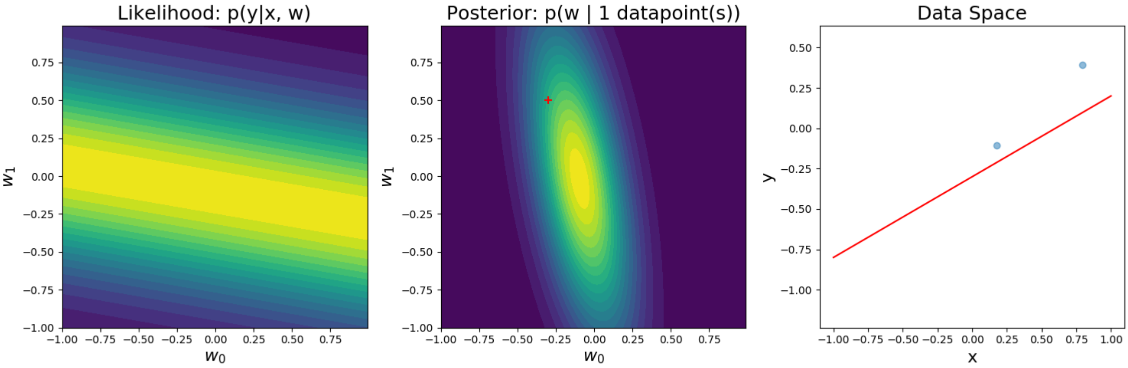
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

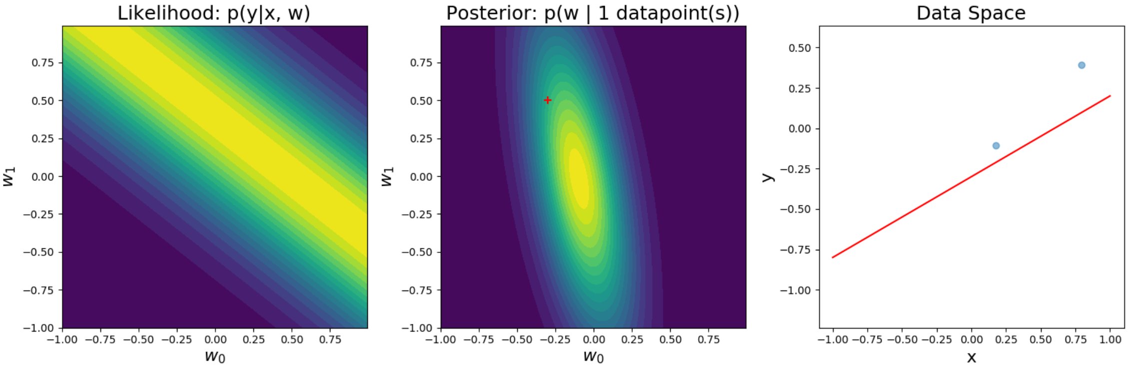
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

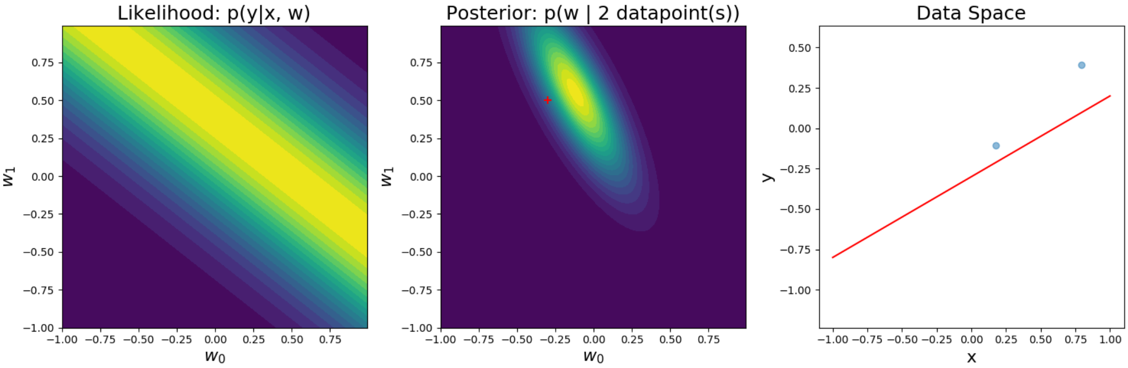
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

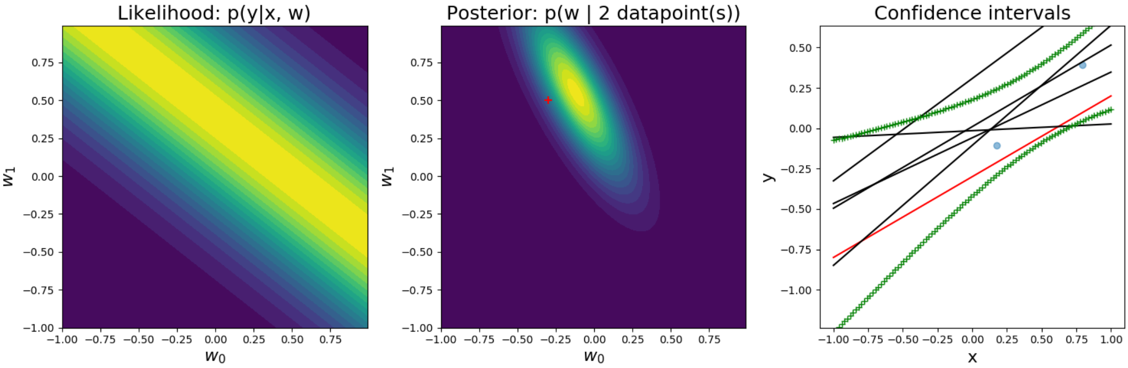
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

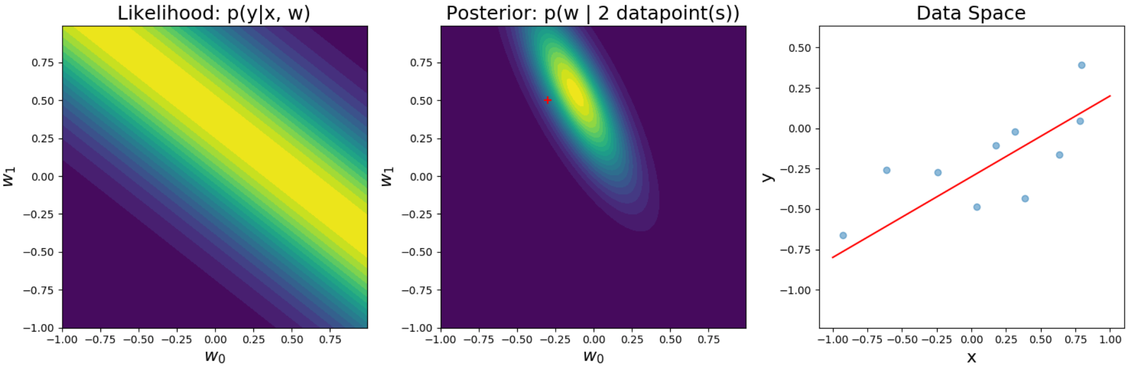
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

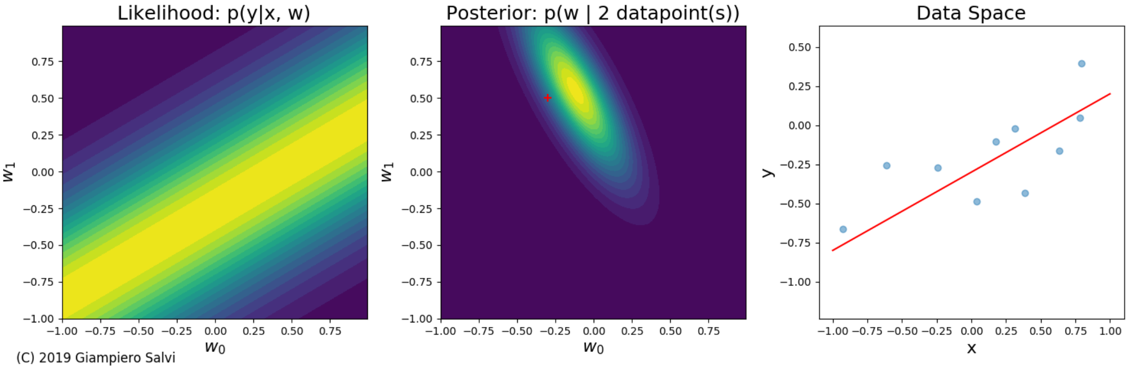
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

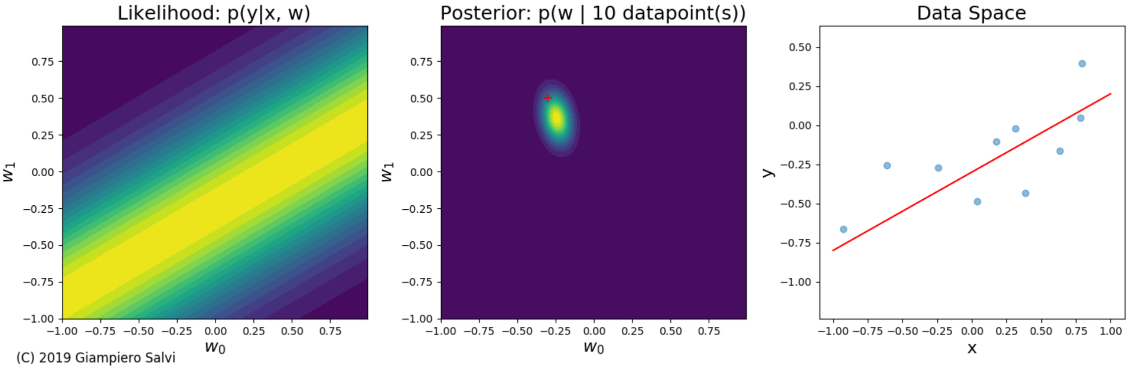
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

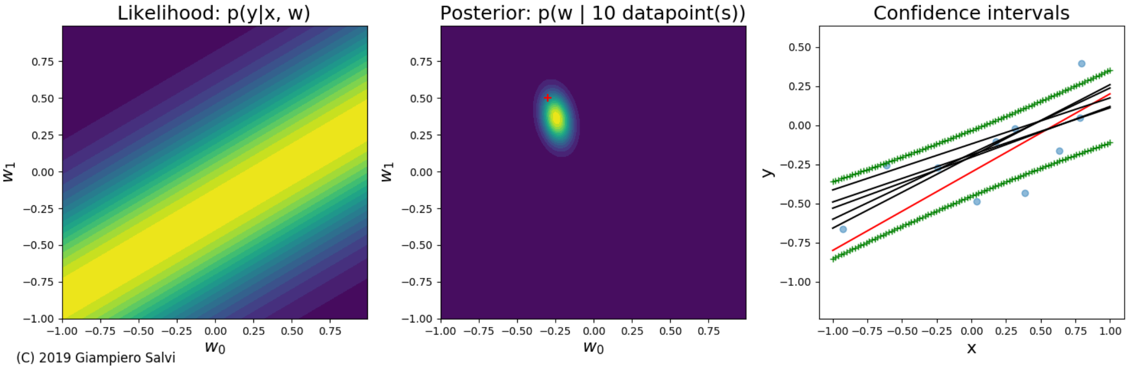
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

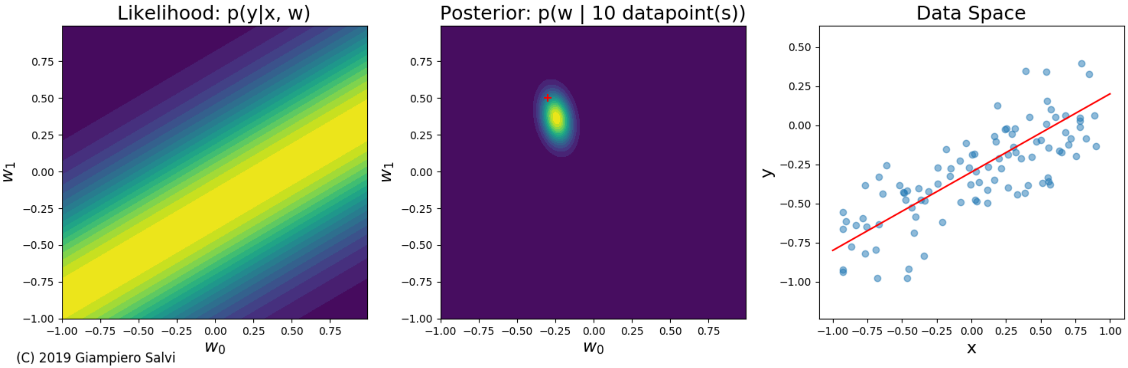
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

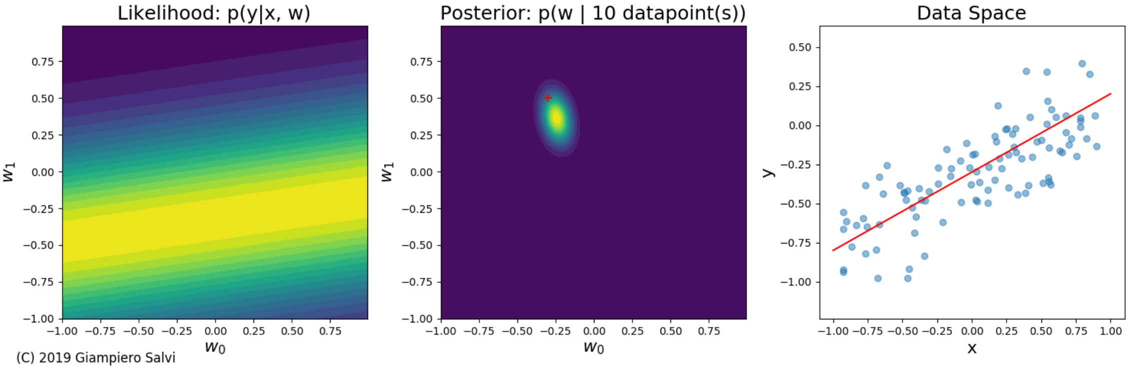
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

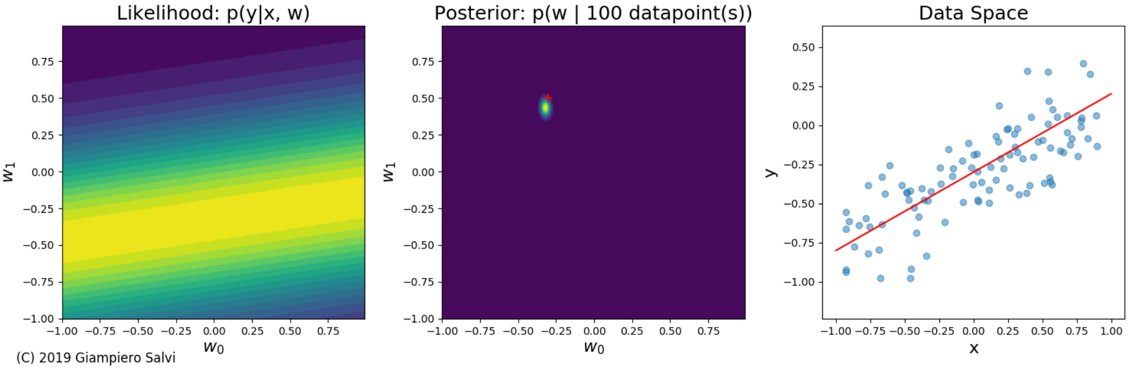
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

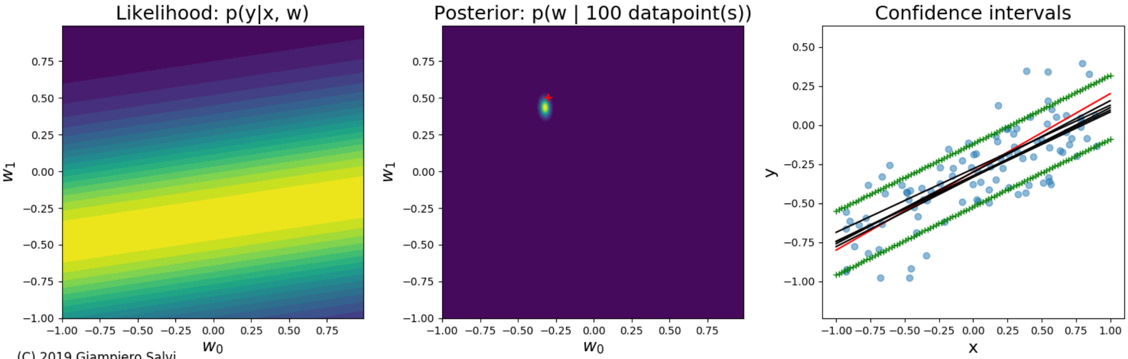
Bayesian Linear Regression: Example



Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

Bayesian Linear Regression: Example



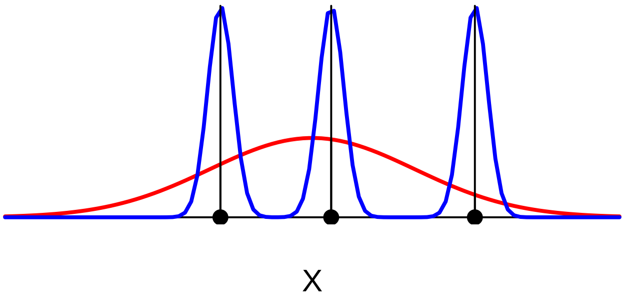
(C) 2019 Giampiero Salvi

Largely adapted from `https://zjost.github.io/bayesian-linear-regression/`

Inspired by Fig 3.7 in Bishop's Pattern Recognition and Machine Learning

Overfitting and Maximum Likelihood

we can make the likelihood arbitrary large by increasing the number of parameters



Occam's Razor and Bayesian Learning

Remember that:

$$p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D}) = \int_{\theta \in \Theta} p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \theta) p(\theta|\mathcal{D}) d\theta$$

Occam's Razor and Bayesian Learning

Remember that:

$$p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathcal{D}) = \int_{\theta \in \Theta} p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \theta) p(\theta|\mathcal{D}) d\theta$$

Intuition:

More complex models fit the data very well (large $p(\mathcal{D}|\theta)$ and $p(\theta|\mathcal{D})$ but only for small regions of the parameter space Θ.

Limitations

- not always possible to compute posterior (conjugate priors)
- approximations with high computational cost (sampling methods) or complex solutions (variational methods)
- sometime we want to have non-informative priors
- for unbounded continuous variables this can be difficult

Outline

Curse of dimensionality

1-dimension

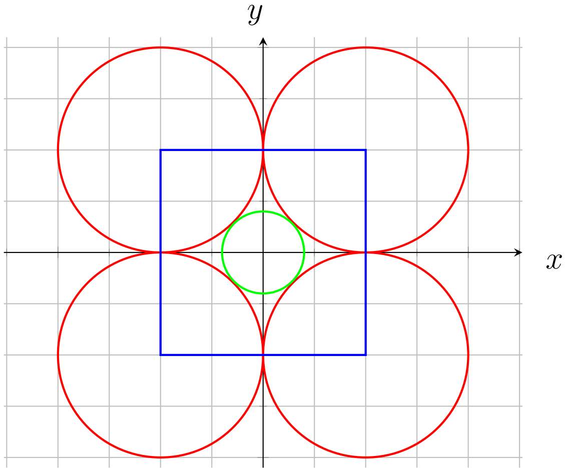
$$y(x, w) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$
$$(4 \text{ parameters})$$

D-dimension

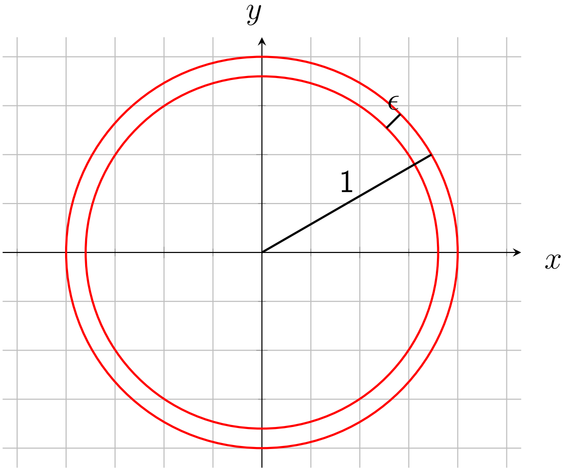
$$y(x, w) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$
$$(1 + D + D^2 + D^3 \text{parameters})$$

High dimensions and intuition

- radius of red circles $= 1$
- side of blue square $= 2$
- what is the radius of the green circle?
- what is the radius of the sphere in 3D?
- how about higher dimensions?



High dimensions and intuition

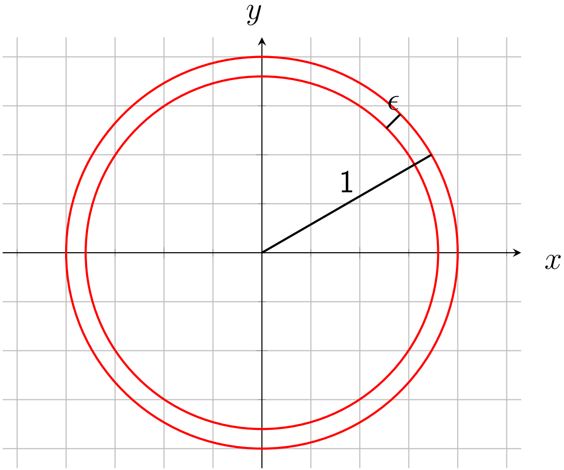


- What is ratio between the volume between the spheres and the volume of the large sphere?

$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = \dots$$

- In D dimensions $V_D(r) = K_D r^D$
- Examples:
  - 2D: $K_2 = \pi$
  - 3D: $K_3 = \frac{4}{3}\pi$
  - $\dots$

High dimensions and intuition



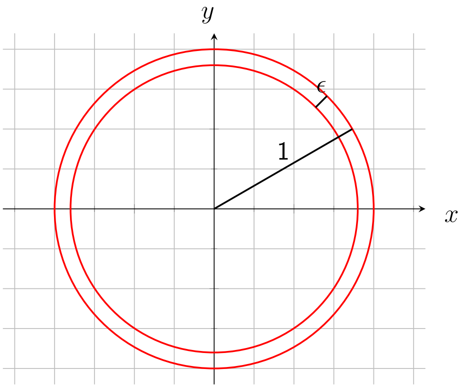
- What is ratio between the volume between the spheres and the volume of the large sphere?

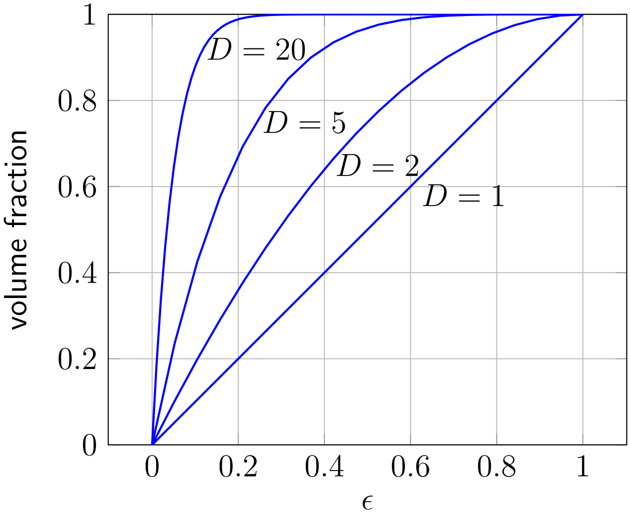
$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = \dots$$

- In D dimensions $V_D(r) = K_D r^D$

$$\begin{aligned} \dots &= \frac{K_D 1^D - K_D(1-\epsilon)^D}{K_D 1^D} \\ &= 1 - (1-\epsilon)^D \end{aligned}$$

High dimensions and intuition





Example: Euclidean Distance

Two points in D dimensions:

$$\mathbf{a} = (a_1, a_2, \dots, a_D)$$
$$\mathbf{b} = (b_1, b_2, \dots, b_D)$$

Euclidean square distance

$$d^2(\mathbf{a}, \mathbf{b}) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots (a_D - b_D)^2$$

If $D = 1000$ it is enough that just a few coordinates differ.