Priors and Latent Variables

DD2421

Bob L. T. Sturm

Check your understanding!

Consider a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ of N independent and identically distributed observations where each $\mathbf{x}_n$ is a p-dimensional real vector. Assume the random variable Y_n is distributed Laplacian with a mean $\boldsymbol{\beta}^T\mathbf{x}_n$ and known scale parameter $b > 0$. In other words, $Y_n|\mathbf{x}_n, \boldsymbol{\beta} \sim \mathcal{L}(\boldsymbol{\beta}^T\mathbf{x}_n, b)$. Define the a priori distribution of the parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ *multivariate* Gaussian with parameters mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$.

Check your understanding!

Consider a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ of N independent and identically distributed observations where each $\mathbf{x}_n$ is a p-dimensional real vector. Assume the random variable Y_n is distributed Laplacian with a mean $\boldsymbol{\beta}^T\mathbf{x}_n$ and known scale parameter $b > 0$. In other words, $Y_n|\mathbf{x}_n, \boldsymbol{\beta} \sim \mathcal{L}(\boldsymbol{\beta}^T\mathbf{x}_n, b)$. Define the a priori distribution of the parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ *multivariate* Gaussian with parameters mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$.

- $Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta}) = \frac{1}{2b} \exp\left[-\frac{|y_n - \mathbf{x}_n^T\boldsymbol{\beta}|}{b}\right]$
- $P(\boldsymbol{\beta}) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$

Check your understanding!

Derive the maximum likelihood (ML) estimate of $\boldsymbol{\beta}$.

- $Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta}) = \frac{1}{2b} \exp\left[-\frac{|y_n - \mathbf{x}_n^T \boldsymbol{\beta}|}{b}\right]$
- $P(\boldsymbol{\beta}) = \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$

The log-likelihood of the dataset is given by

$$\log Pr(\mathcal{D}|\boldsymbol{\beta}) = \sum_{n=1}^{N} \log Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta})$$

The maximum likelihood estimate of $\boldsymbol{\beta}$ is defined

$$\boldsymbol{\beta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\beta}} \log Pr(\mathcal{D}|\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta})$$
$$= \arg\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log \left(\frac{1}{2b} \exp\left[-\frac{|y_n - \mathbf{x}_n^T \boldsymbol{\beta}|}{b}\right]\right)$$

Check your understanding!

$$
\begin{aligned}
\boldsymbol{\beta}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log\left(\frac{1}{2b}\exp\left[-\frac{|y_n - \mathbf{x}_n^T\boldsymbol{\beta}|}{b}\right]\right) \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} -\log(2b) - \frac{1}{b}|y_n - \mathbf{x}_n^T\boldsymbol{\beta}| \\
&= \arg\min_{\boldsymbol{\beta}} N\log(2b) + \frac{1}{b}\sum_{n=1}^{N} |y_n - \mathbf{x}_n^T\boldsymbol{\beta}| \\
&= \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\|_1
\end{aligned}
$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ and $\mathbf{X} = [\mathbf{x}_1|\mathbf{x}_2|\ldots|\mathbf{x}_N]$.

Check your understanding!

Which of the following statements is not true? (There may be more than one.)

1. As b increases and N remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.
2. As N increases and b remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.
3. As b decreases and N remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.
4. As N decreases and b remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.

$$Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta}) = \frac{1}{2b} \exp\left[-\frac{|y_n - \mathbf{x}_n^T \boldsymbol{\beta}|}{b}\right]$$

Check your understanding!

Which of the following statements is not true? (There may be more than one.)

1. As b increases and N remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.
2. As N increases and b remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer. $\leftarrow$
3. As b decreases and N remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.$\leftarrow$
4. As N decreases and b remains constant, the ML estimate of $\boldsymbol{\beta}$ becomes poorer.

$$Pr(y_n|\mathbf{x}_n, \boldsymbol{\beta}) = \frac{1}{2b} \exp\left[-\frac{|y_n - \mathbf{x}_n^T\boldsymbol{\beta}|}{b}\right]$$

Outline

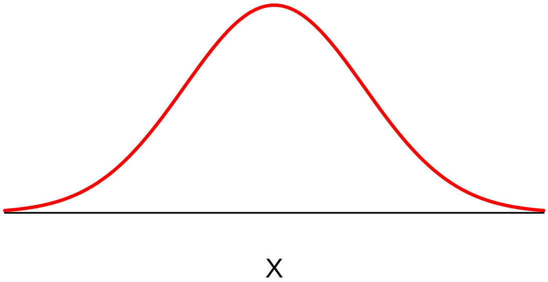
1. Incorporating Priors
   - Maximum a Posteriori Estimation
   - Bayesian Non-Parametric Methods
   - Model Selection and Occam's Razor

2. Unsupervised Learning
   - Classification vs Clustering
   - Heuristic Example: K-means
   - Expectation Maximization

Outline

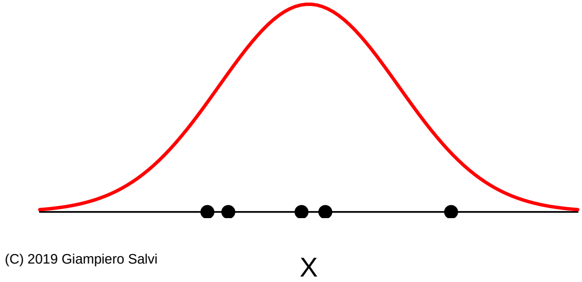
Problem: few data points

10 repetitions with 5 points each



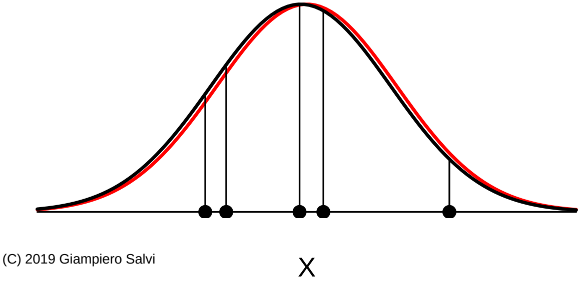
Problem: few data points

10 repetitions with 5 points each



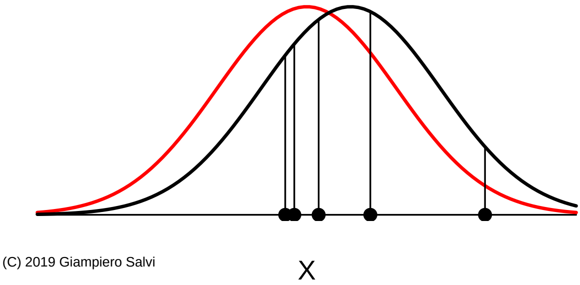
Problem: few data points

10 repetitions with 5 points each



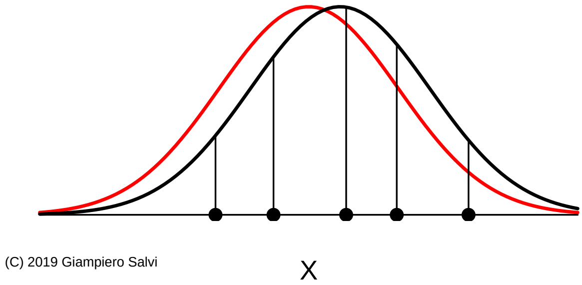
Problem: few data points

10 repetitions with 5 points each



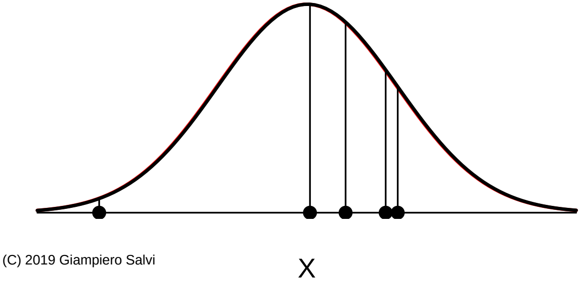
Problem: few data points

10 repetitions with 5 points each



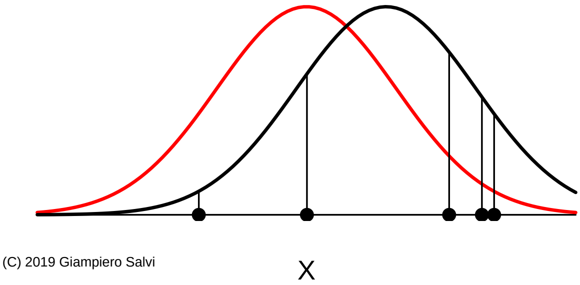
Problem: few data points

10 repetitions with 5 points each



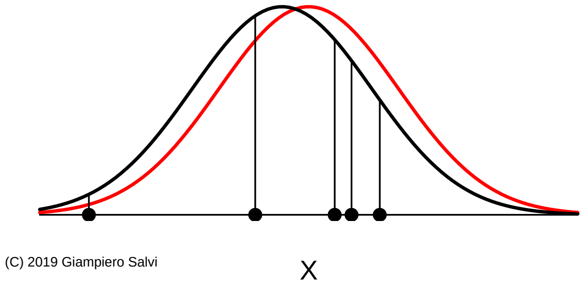
Problem: few data points

10 repetitions with 5 points each



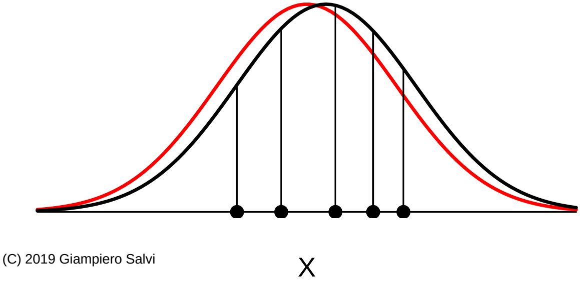
Problem: few data points

10 repetitions with 5 points each



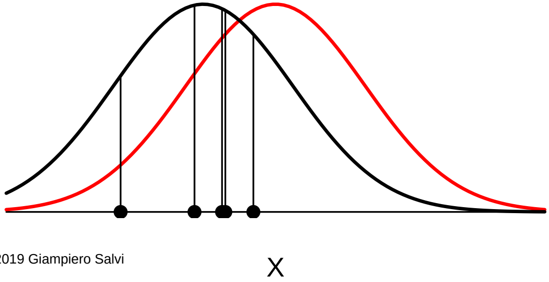
Problem: few data points

10 repetitions with 5 points each



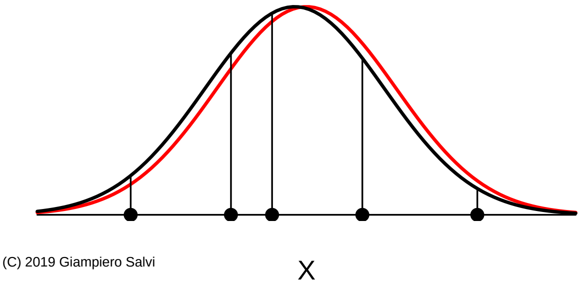
Problem: few data points

10 repetitions with 5 points each



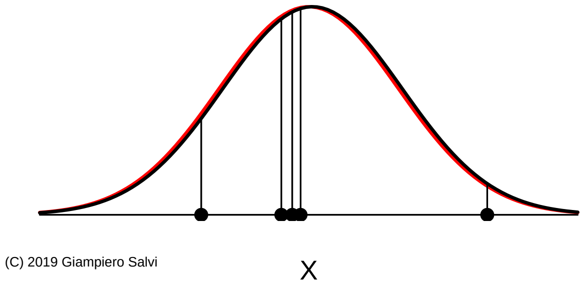
Problem: few data points

10 repetitions with 5 points each



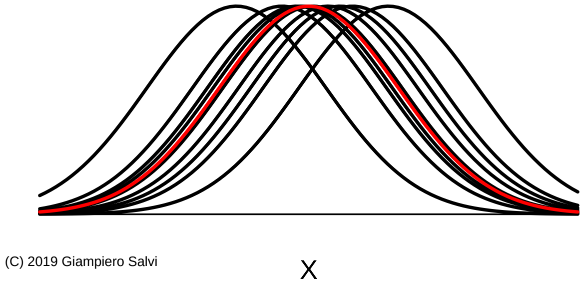
Problem: few data points

10 repetitions with 5 points each



Problem: few data points

10 repetitions with 5 points each



Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

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

Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

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

Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

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

Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

$$\begin{aligned}
\theta_{\mathsf{MAP}} &= \arg\max_{\theta} Pr(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} \frac{Pr(\theta)Pr(\mathcal{D}|\theta)}{Pr(\mathcal{D})} \\
&= \arg\max_{\theta} Pr(\theta)Pr(\mathcal{D}|\theta) \\
&= \arg\max_{\theta} \left[Pr(\theta) \prod_{n=1}^{N} Pr(x_n|\theta) \right]
\end{aligned}$$

Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

$$\begin{aligned}
\theta_{\mathsf{MAP}} &= \arg\max_\theta Pr(\theta|\mathcal{D}) \\
&= \arg\max_\theta \frac{Pr(\theta)Pr(\mathcal{D}|\theta)}{Pr(\mathcal{D})} \\
&= \arg\max_\theta Pr(\theta)Pr(\mathcal{D}|\theta) \\
&= \arg\max_\theta \left[Pr(\theta)\prod_{n=1}^{N} Pr(x_n|\theta)\right] \\
&= \arg\max_\theta \left[\log Pr(\theta) + \sum_{n=1}^{N} \log Pr(x_n|\theta)\right]
\end{aligned}$$

Maximum a Posteriori Estimation

- Recall: ML estimation chooses θ to maximize probability of $\mathcal{D}$.
- MAP estimation chooses the most likely θ given $\mathcal{D}$.

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

- $\log Pr(\theta)$ works as a *regularizer*.

MAP for Linear Regression

Model (deterministic):

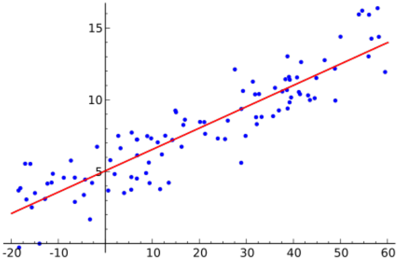
$$y = \mathbf{w}^T \mathbf{x} + \epsilon$$

With:

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

Therefore:

$$Y|X \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma_Y^2)$$



MAP for Linear Regression

Model (deterministic):

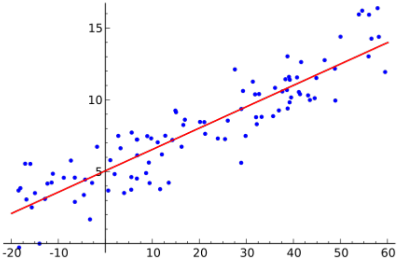
$$y = \mathbf{w}^T\mathbf{x} + \epsilon$$

With:

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

Therefore:

$$Y|X \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma_Y^2)$$



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

Example: D-dimension zero-mean spherical Gaussian prior

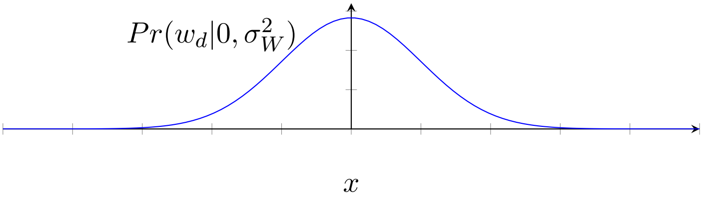
We assume in this case $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_W^2 \mathbf{I}_D)$

$$Pr(\mathbf{w}|\mathbf{0}, \sigma_W^2) = \frac{1}{(2\pi\sigma_W^2)^{\frac{D}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_W^2}\right)$$

Example: D-dimension zero-mean spherical Gaussian prior

We assume in this case $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_W^2 \mathbf{I}_D)$

$$
\begin{aligned}
Pr(\mathbf{w}|\mathbf{0}, \sigma_W^2) &= \frac{1}{(2\pi\sigma_W^2)^{\frac{D}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_W^2}\right) = \\
&= \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_W^2}} \exp\left(-\frac{w_d^2}{2\sigma_W^2}\right)
\end{aligned}
$$



Example: D-dimension zero-mean spherical Gaussian prior

Instead of $\log Pr(y|x,\mathbf{w})$ as in MLE, we optimize $\log Pr(\mathbf{w}|y,x)$:

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y,x) = \arg\max_{\mathbf{w}} \log\left[Pr(y|x,\mathbf{w})Pr(\mathbf{w})\right]$$

Example: D-dimension zero-mean spherical Gaussian prior

Instead of $\log Pr(y|x, \mathbf{w})$ as in MLE, we optimize $\log Pr(\mathbf{w}|y, x)$:

$$\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y, x) = \arg\max_{\mathbf{w}} \log \left[Pr(y|x, \mathbf{w}) Pr(\mathbf{w}) \right] \\
&= \arg\max_{\mathbf{w}} \left[\sum_{n=1}^{N} \log Pr(y_i|\mathbf{x}_i, \mathbf{w}) + \log Pr(\mathbf{w}) \right]
\end{aligned}$$

Example: D-dimension zero-mean spherical Gaussian prior

Instead of $\log Pr(y|x, \mathbf{w})$ as in MLE, we optimize $\log Pr(\mathbf{w}|y, x)$:

$$\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y, x) = \arg\max_{\mathbf{w}} \log \left[Pr(y|x, \mathbf{w}) Pr(\mathbf{w})\right] \\
&= \arg\max_{\mathbf{w}} \left[\sum_{n=1}^{N} \log Pr(y_i|\mathbf{x}_i, \mathbf{w}) + \log Pr(\mathbf{w})\right] \\
&= \arg\min_{\mathbf{w}} \left[\frac{1}{2\sigma_Y^2} \sum_{n=1}^{N} \left(y_n - \mathbf{w}^T \mathbf{x}_n\right)^2 + \frac{1}{2\sigma_W^2} \mathbf{w}^T \mathbf{w}\right]
\end{aligned}$$

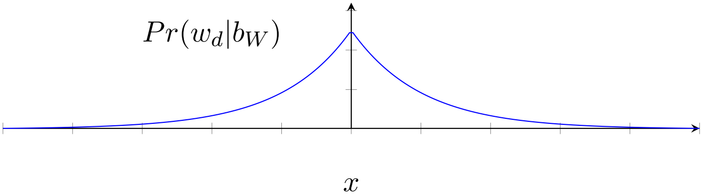
Example: D-dimension zero-mean spherical Gaussian prior

Instead of $\log Pr(y|x,\mathbf{w})$ as in MLE, we optimize $\log Pr(\mathbf{w}|y,x)$:

$$
\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y,x) = \arg\max_{\mathbf{w}} \log\left[Pr(y|x,\mathbf{w})Pr(\mathbf{w})\right] \\
&= \arg\max_{\mathbf{w}} \left[\sum_{n=1}^{N} \log Pr(y_i|\mathbf{x}_i,\mathbf{w}) + \log Pr(\mathbf{w})\right] \\
&= \arg\min_{\mathbf{w}} \left[\frac{1}{2\sigma_Y^2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2 + \frac{1}{2\sigma_W^2}\mathbf{w}^T\mathbf{w}\right] \\
&= \arg\min_{\mathbf{w}} \left[\underbrace{\frac{1}{2\sigma_Y^2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit the data (ML)}} + \underbrace{\frac{1}{2\sigma_W^2}\mathbf{w}^T\mathbf{w}}_{\text{keep } \mathbf{w} \text{ short}}\right]
\end{aligned}
$$

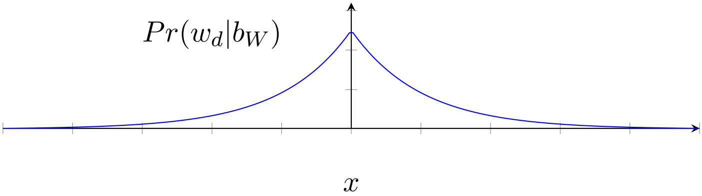
Example: each dimension is iid zero-mean Laplace

In this case $Pr(\mathbf{w}|\mathbf{b_W}) = \prod_d \frac{1}{2b_W} \exp\left(-\frac{|w_d|}{b_W}\right)$



Example: each dimension is iid zero-mean Laplace

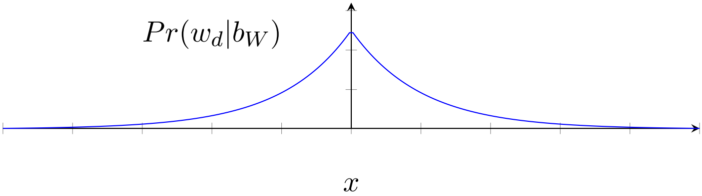
In this case $Pr(\mathbf{w}|\mathbf{b_W}) = \prod_d \frac{1}{2b_W} \exp\left(-\frac{|w_d|}{b_W}\right)$



$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y,x) = \arg\max_{\mathbf{w}} \log \left[Pr(y|x,\mathbf{w})Pr(\mathbf{w})\right]$$

Example: each dimension is iid zero-mean Laplace

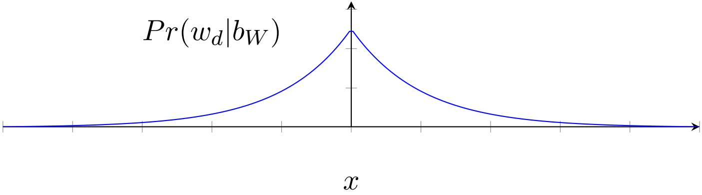
In this case $Pr(\mathbf{w}|\mathbf{b_W}) = \prod_d \frac{1}{2b_W} \exp\left(-\frac{|w_d|}{b_W}\right)$



$$\begin{aligned}\mathbf{w}_{\mathsf{MAP}} &= \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y,x) = \arg\max_{\mathbf{w}} \log\left[Pr(y|x,\mathbf{w})Pr(\mathbf{w})\right] \\ &= \arg\min_{\mathbf{w}} \left[\frac{1}{2\sigma_Y^2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2 + \frac{1}{b_W}\sum_{d=1}^{D}|w_d|\right]\end{aligned}$$

Example: each dimension is iid zero-mean Laplace

In this case $Pr(\mathbf{w}|\mathbf{b_W}) = \prod_d \frac{1}{2b_W} \exp\left(-\frac{|w_d|}{b_W}\right)$



$$\begin{aligned}\mathbf{w}_{\mathsf{MAP}} &= \arg\max_{\mathbf{w}} \log Pr(\mathbf{w}|y,x) = \arg\max_{\mathbf{w}} \log\left[Pr(y|x,\mathbf{w})Pr(\mathbf{w})\right] \\ &= \arg\min_{\mathbf{w}} \left[\underbrace{\frac{1}{2\sigma_Y^2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit the data (ML)}} + \underbrace{\frac{1}{b_W}\sum_{d=1}^{D}|w_d|}_{\text{keep } \mathbf{w} \text{ ``simple''}}\right]\end{aligned}$$

Impact of different priors

For linear regression $y = \mathbf{w}^T\mathbf{x} + \epsilon$, MAP estimation of $\mathbf{w}$ performs differently considering the prior we assume!

$$\mathbf{w}^*(p) = \arg\min_{\mathbf{w}} \left[\underbrace{\sum_{n=1}^{N} \left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit the data (ML)}} + \underbrace{\lambda \sum_{d=1}^{D} |w_d|^p}_{\text{regularize}} \right]$$

- $\lambda = 0$ or $W_d \sim$ Uniform: just fit the data

Impact of different priors

For linear regression $y = \mathbf{w}^T\mathbf{x} + \epsilon$, MAP estimation of $\mathbf{w}$ performs differently considering the prior we assume!

$$\mathbf{w}^*(p) = \arg\min_{\mathbf{w}} \left[\underbrace{\sum_{n=1}^{N} \left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit the data (ML)}} + \underbrace{\lambda \sum_{d=1}^{D} |w_d|^p}_{\text{regularize}} \right]$$

- $\lambda = 0$ or $W_d \sim$ Uniform: just fit the data
- $\lambda > 0$ and assuming iid Normal ($p = 2$): fit the data and make $\mathbf{w}$ short (also known as *ridge regression*)

Impact of different priors

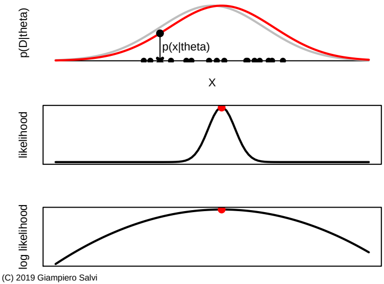
For linear regression $y = \mathbf{w}^T\mathbf{x} + \epsilon$, MAP estimation of $\mathbf{w}$ performs differently considering the prior we assume!

$$\mathbf{w}^*(p) = \arg\min_{\mathbf{w}} \left[\underbrace{\sum_{n=1}^{N} \left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit the data (ML)}} + \lambda \underbrace{\sum_{d=1}^{D} |w_d|^p}_{\text{regularize}} \right]$$

- $\lambda = 0$ or $W_d \sim$ Uniform: just fit the data
- $\lambda > 0$ and assuming iid Normal ($p = 2$): fit the data and make $\mathbf{w}$ short (also known as *ridge regression*)
- $\lambda > 0$ and assuming iid Laplace ($p = 1$): fit the data and make $\mathbf{w}$ sparse (also know as the *Least Absolute Shrinkage and Selection Operator (LASSO)*)

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, assume everything is known



Limitations (Linear Regression)

- With MAP estimation, the problem shifts to defining the parameters of the prior $Pr(\mathbf{w})$ (equivalently choosing λ in ridge or LASSO regression)
- We still have uncertainty in the posterior $Pr(y|\mathbf{x}, \mathbf{w}^*)$ but this is not explicit.
- We don't want $Pr(y|\mathbf{x}, \mathbf{w})$, but instead $Pr(y|\mathbf{x}, \mathcal{D})$.

Bayesian estimation (non-parametric models)

$$\begin{array}{llclcl}
\text{ML:} & \mathcal{D} & \rightarrow & \theta_{\text{ML}} & \rightarrow & Pr(y|\mathbf{x}, \theta_{\text{ML}}) \\
\text{MAP:} & \mathcal{D}, Pr(\theta) & \rightarrow & \theta_{\text{MAP}} & \rightarrow & Pr(y|\mathbf{x}, \theta_{\text{MAP}}) \\
\text{Bayes:} & \mathcal{D}, Pr(\theta) & \rightarrow & Pr(\theta|\mathcal{D}) & \rightarrow & Pr(y|\mathbf{x}, \mathcal{D})
\end{array}$$

1. consider θ as a random variable (same as MAP)
2. characterize θ with the posterior distribution $Pr(\theta|\mathcal{D})$ given the data
3. compute new predicting posterior $Pr(y|\mathbf{x}, \mathcal{D})$ marginalizing over θ (predictive posterior)

$$Pr(y|\mathbf{x}, \mathcal{D}) = \int_{\theta \in \Theta} Pr(y|\mathbf{x}, \theta) Pr(\theta|\mathcal{D}) d\theta$$

Bayesian Linear Regression

Setup:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

Model (same as MAP):

- $Y_1, \ldots, Y_n$ conditionally independent given $\mathbf{w}$
- $Y|X \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma_Y^2)$
- $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_W, \boldsymbol{\Sigma}_{\boldsymbol{W}}), \quad \mathbf{w} \in \mathbb{R}^D$
- we assume values for σ_Y^2, $\boldsymbol{\mu}_W$, and $\boldsymbol{\Sigma}_{\boldsymbol{W}}$, so $\theta \equiv \mathbf{w}$

Goal: Estimate $Pr(y|\mathbf{x}, \mathcal{D})$

Bayesian Linear Regression

$$
\begin{aligned}
Pr(y|\mathbf{x},\mathcal{D}) &= \int_{\mathbb{R}^D} Pr(y|\mathbf{x},\mathcal{D},\mathbf{w})Pr(\mathbf{w}|\mathbf{x},\mathcal{D})d\mathbf{w} \\
&= \int_{\mathbb{R}^D} Pr(y|\mathbf{x},\mathbf{w})Pr(\mathbf{w}|\mathcal{D})d\mathbf{w}
\end{aligned}
$$

(assuming $Y|\mathbf{W}=\mathbf{w}$ is independent of $\mathcal{D}$, and $\mathbf{W}$ is independent of $\mathbf{x}$). Results obtained with a lot of passages:

- if $Pr(\mathbf{w})$ is Gaussian, then $Pr(\mathbf{w}|\mathcal{D})$ is Gaussian
- since $Pr(y|\mathbf{x},\mathbf{w})$ is Gaussian, $Pr(y|\mathbf{x},\mathcal{D})$ is also Gaussian
- Closed-form results in this case.

Find a walk-through of the derivations here

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 $p(y|\mathbf{x}, \mathcal{D})$, part 1–3
  `https://youtu.be/xyuSiKXttxw`
  `https://youtu.be/vTcsacTqlfQ`
  `https://youtu.be/LCISTY9S6SQ`

Closed Form Solutions

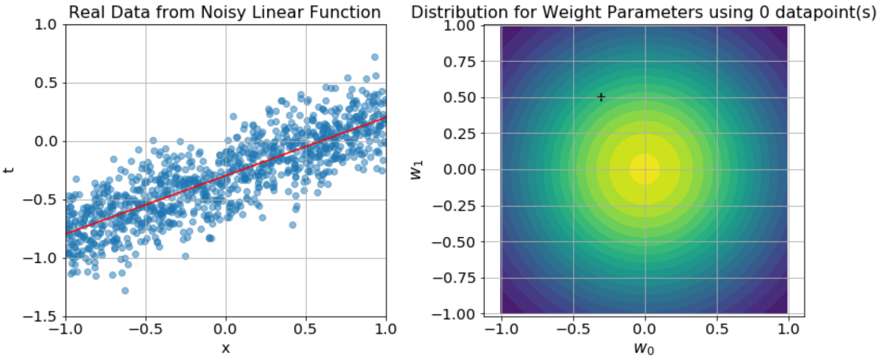
Define $\mathbf{\Phi}^T = [\mathbf{x}_1|\mathbf{x}_2|\ldots|\mathbf{x}_N]$ and $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ from $\mathcal{D}$. With the assumed set-up, the posterior $\mathbf{w}|\mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}|\mathcal{D}}, \mathbf{\Sigma}_{\mathbf{w}|\mathcal{D}})$, with:

$$\begin{aligned} \mathbf{\Sigma}_{\mathbf{w}|\mathcal{D}} &= \left[\mathbf{\Sigma}_W^{-1} + \frac{1}{\sigma_Y^2}\mathbf{\Phi}^T\mathbf{\Phi}\right]^{-1} \\ \boldsymbol{\mu}_{\mathbf{w}|\mathcal{D}} &= \mathbf{\Sigma}_{\mathbf{w}|\mathcal{D}}\left[\mathbf{\Sigma}_W^{-1}\boldsymbol{\mu}_W + \frac{1}{\sigma_Y^2}\mathbf{\Phi}^T\mathbf{y}\right] \end{aligned}$$

Predictive posterior:

$$Pr(y|\mathbf{x}, \mathcal{D}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}|\mathcal{D}}^T\mathbf{x}, \sigma_Y^2 + \mathbf{x}^T\mathbf{\Sigma}_{\mathbf{w}|\mathcal{D}}\mathbf{x})$$

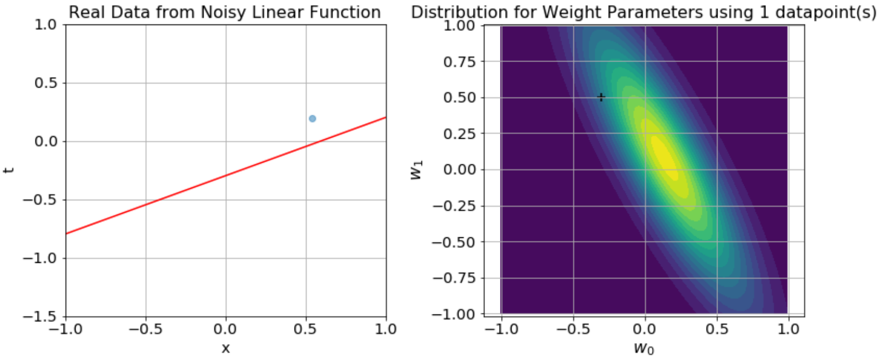
Bayesian Linear Regression: Example



Largely adapted from

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

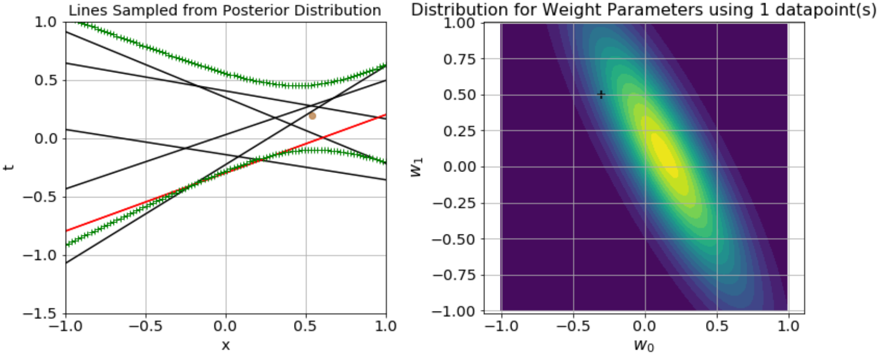
Bayesian Linear Regression: Example



Largely adapted from

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

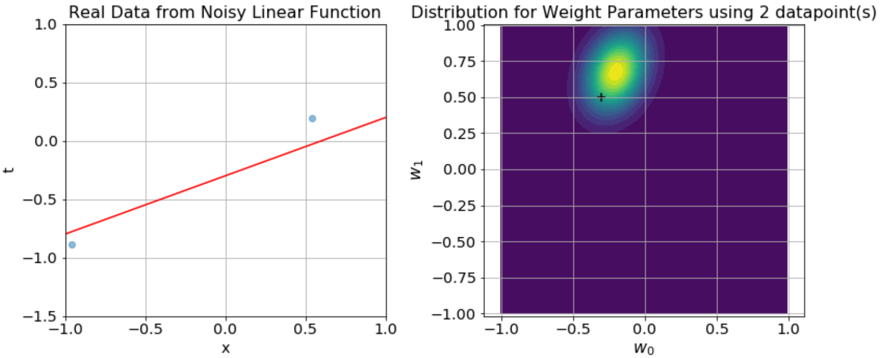
Bayesian Linear Regression: Example



Largely adapted from

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

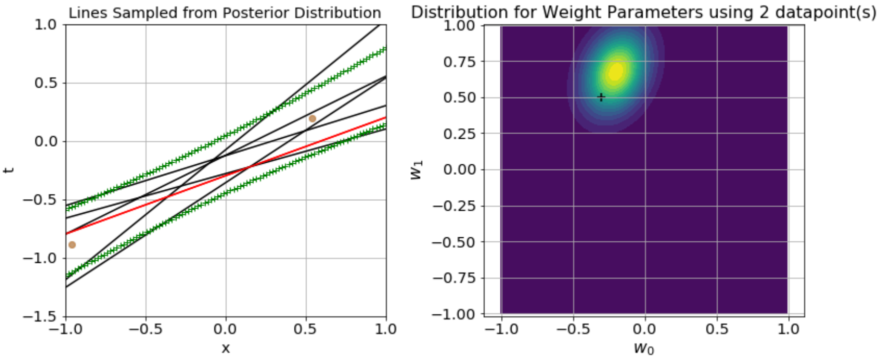
Bayesian Linear Regression: Example



Largely adapted from

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

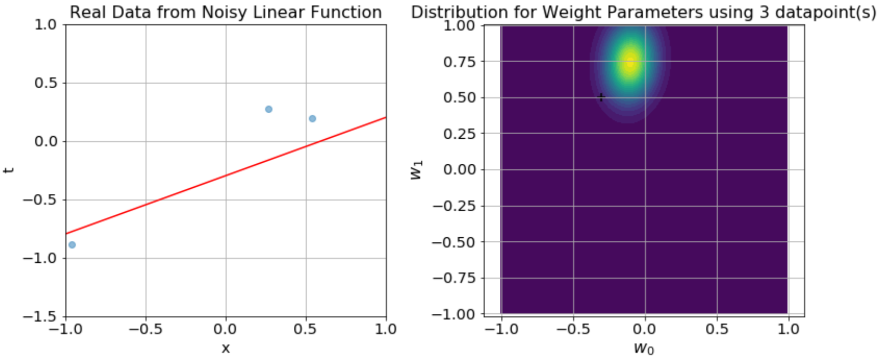
Bayesian Linear Regression: Example



Largely adapted from

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

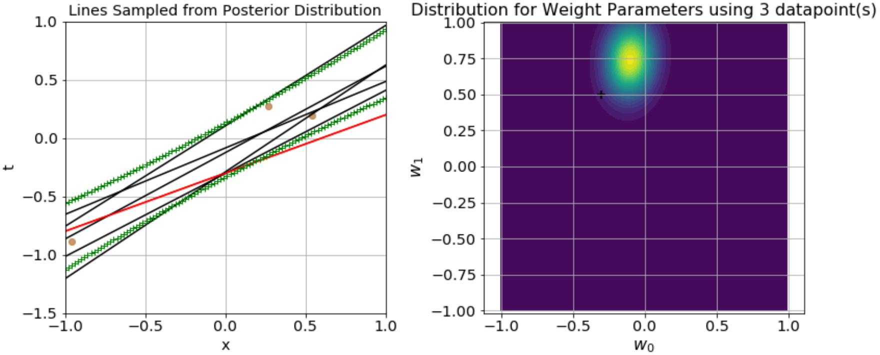
Bayesian Linear Regression: Example



Largely adapted from

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

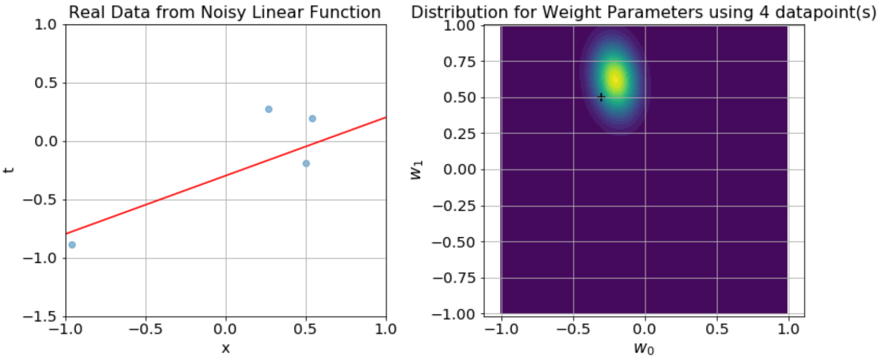
Bayesian Linear Regression: Example



Largely adapted from

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

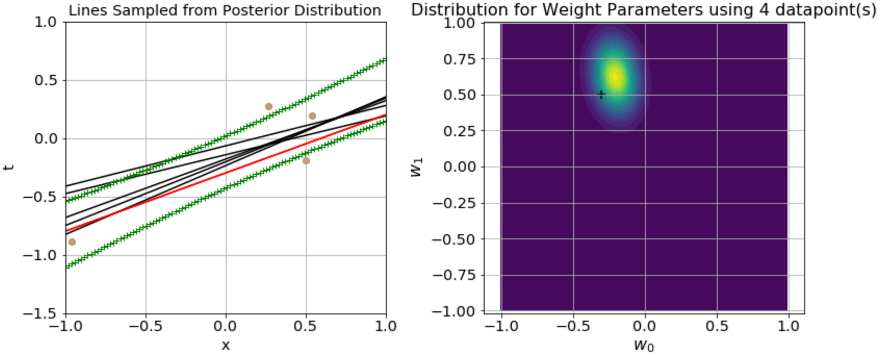
Bayesian Linear Regression: Example



Largely adapted from

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

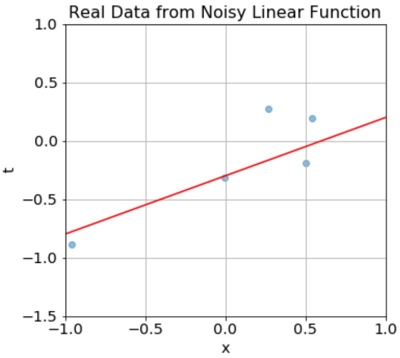
Bayesian Linear Regression: Example

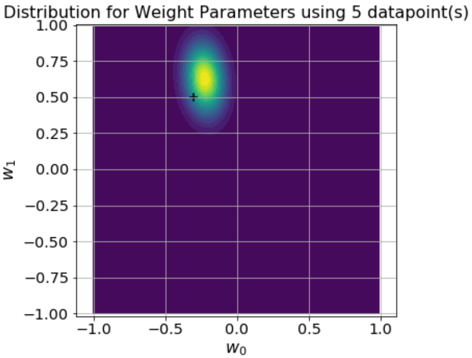


Largely adapted from

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

Bayesian Linear Regression: Example

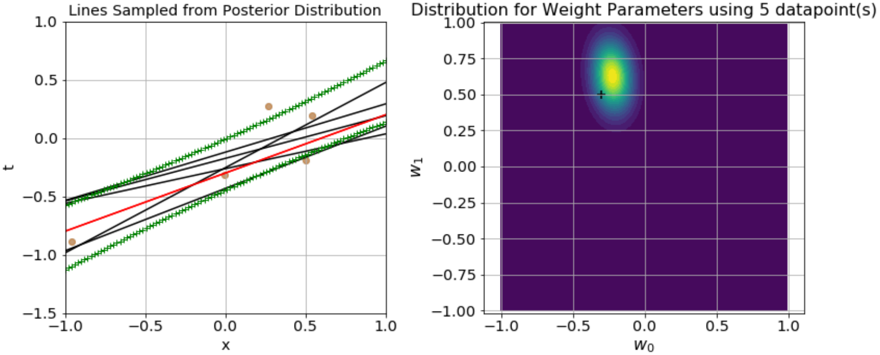




Largely adapted from

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

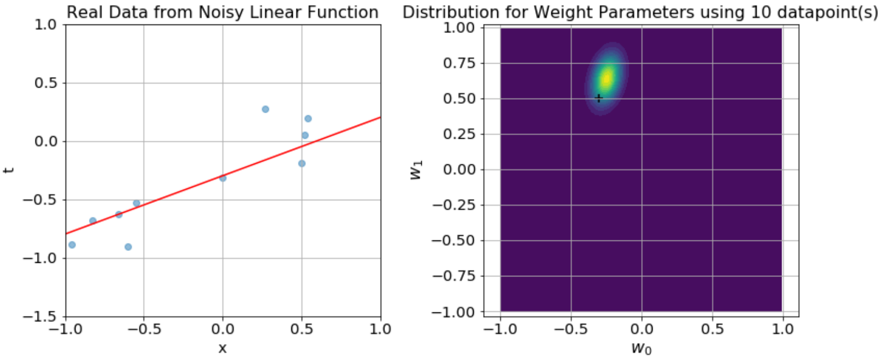
Bayesian Linear Regression: Example



Largely adapted from

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

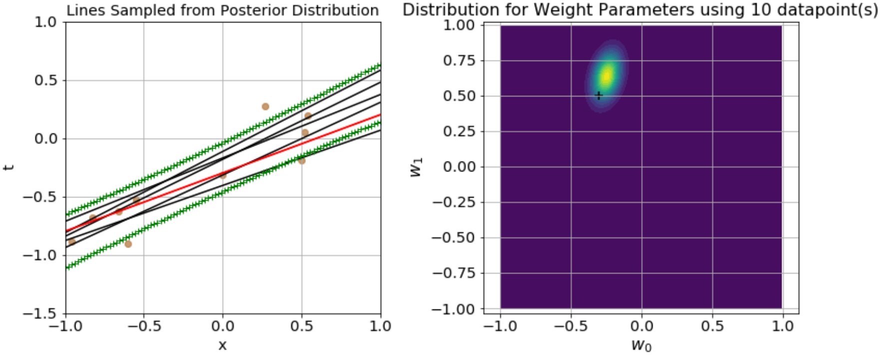
Bayesian Linear Regression: Example



Largely adapted from

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

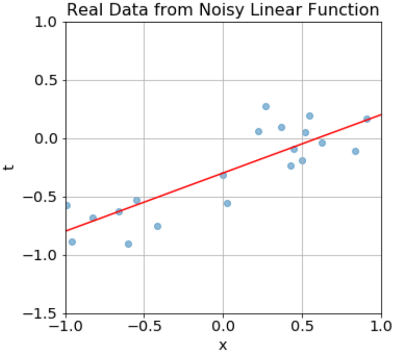
Bayesian Linear Regression: Example

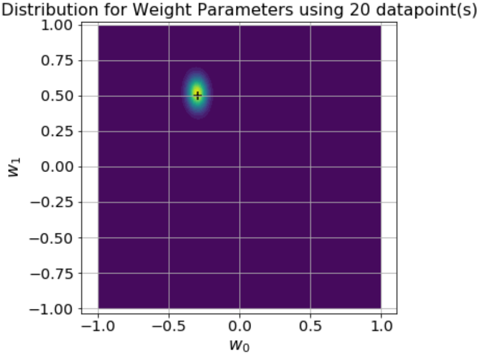


Largely adapted from

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

Bayesian Linear Regression: Example

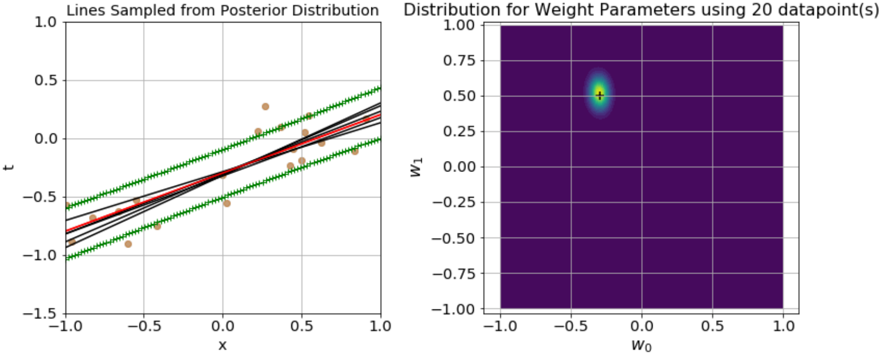




Largely adapted from

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

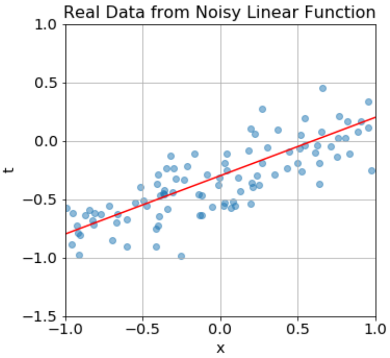
Bayesian Linear Regression: Example

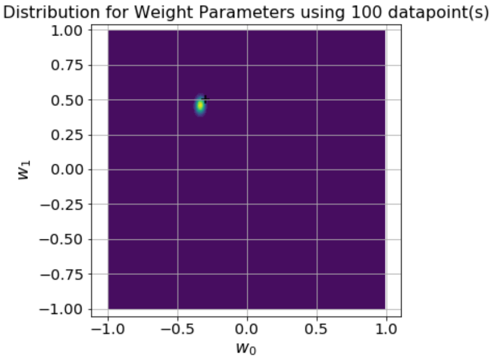


Largely adapted from

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

Bayesian Linear Regression: Example

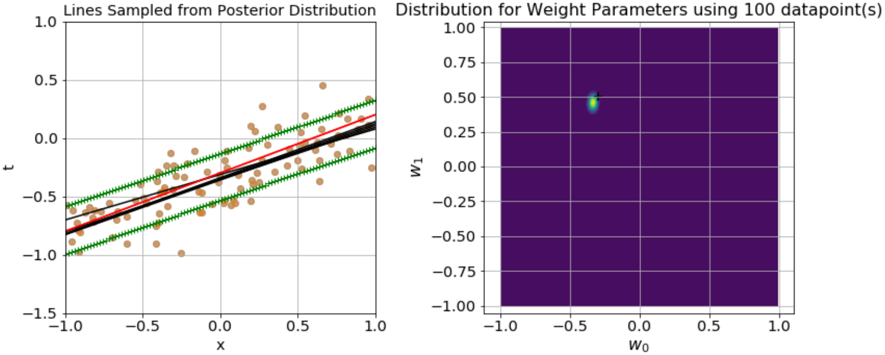




Largely adapted from

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

Bayesian Linear Regression: Example

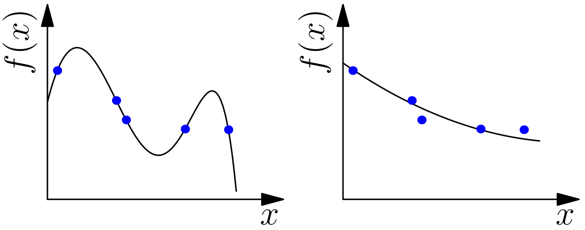


Largely adapted from

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

Consequences

- We are considering the uncertainty over the choice of $\mathbf{w}$ as well as the original uncertainty σ_Y^2.
- This provides a means to prevent overfitting



Occam's Razor

Choose the simplest explanation for the observed data

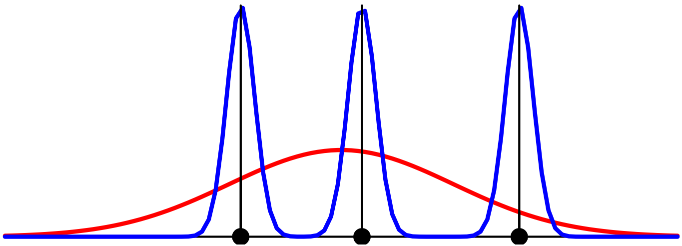
Important factors:

- number of model parameters
- number of data points
- model fit to the data

Overfitting and Maximum Likelihood

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

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



Occam's Razor and Bayesian Learning

Remember that:

$$Pr(y|\mathbf{x},\mathcal{D}) = \int_{\theta\in\Theta} Pr(y|\mathbf{x},\theta)Pr(\theta|\mathcal{D})d\theta$$

Occam's Razor and Bayesian Learning

Remember that:

$$Pr(y|\mathbf{x},\mathcal{D}) = \int_{\theta\in\Theta} Pr(y|\mathbf{x},\theta)Pr(\theta|\mathcal{D})d\theta$$

Intuition:

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

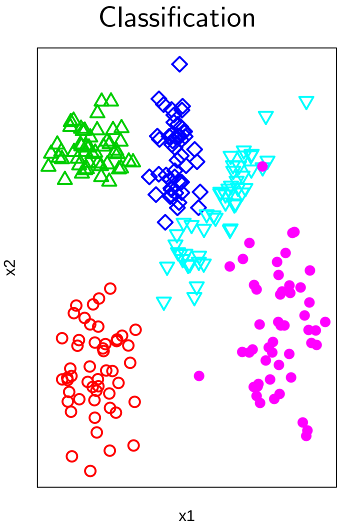
Limitations of Bayesian Non-parametric Methods

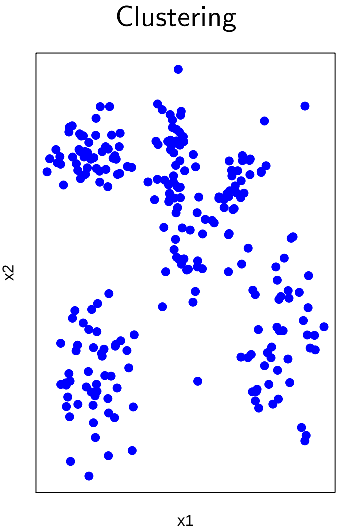
$$Pr(y|\mathbf{x}, \mathcal{D}) = \int_{\mathbb{R}^D} Pr(y|\mathbf{x}, \mathbf{w}) Pr(\mathbf{w}|\mathcal{D}) d\mathbf{w}$$

- closed form solution for $Pr(\mathbf{w}|\mathcal{D})$ is not always possible (conjugate priors)
- can use approximations with high computational cost (sampling methods) or complex solutions (variational methods)
- sometimes we will have a non-informative prior of $\mathbf{W}$, but Bayesian methods carry uncertainty estimates.

Outline

Clustering vs Classification





Heuristic Example: K-means

- describes each class with a centroid
- a point belongs to a class if the corresponding centroid is closest (Euclidean distance)
- iterative procedure
- guaranteed to converge
- not guaranteed to find the optimal solution
- used in *vector quantization* (since the 1950's)

K-means: algorithm

Data: No. of desired clusters K, data set $\mathcal{D} = \{\mathbf{x}_i \in \mathbb{R}^D\}$, distance function $f : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_+$, threshold ϵ

initialization: sample K centroids $\boldsymbol{\mu}_k^{(0)} \in \mathbb{R}^D$ at random; $t \leftarrow 0$;

repeat

assign each point $\mathbf{x}_i$ to closest centroid:

$$\mathcal{C}_k := \left\{\mathbf{x} \in \mathcal{D} : k = \arg\min_{k'} f(\mathbf{x}, \boldsymbol{\mu}_{k'}^{(t)})\right\}$$

update centroids as mean of each cluster of points:

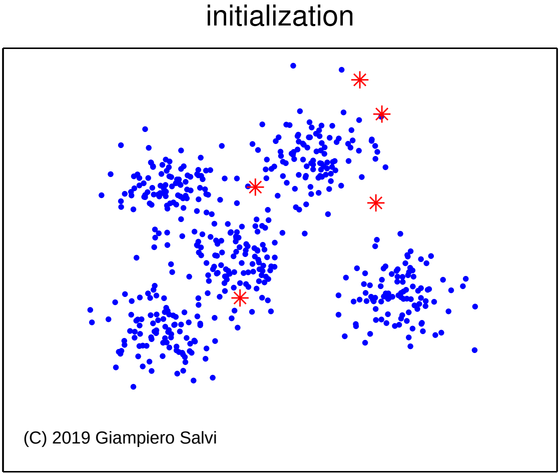
$$\boldsymbol{\mu}_k^{(t+1)} = \frac{1}{|\mathcal{C}_k|} \sum \mathcal{C}_k$$

iterate $t \leftarrow t + 1$

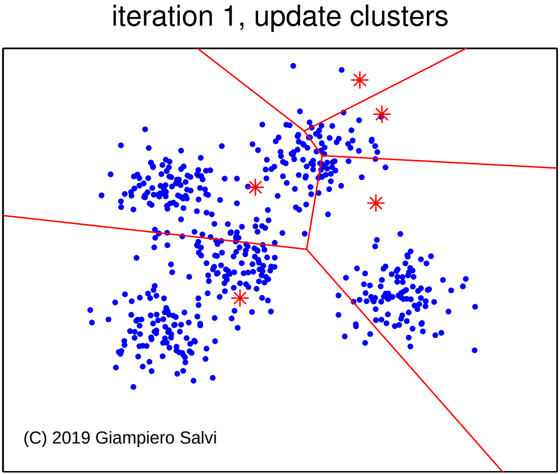
until $\max_k f(\boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\mu}_k^{(t)}) < \epsilon$;

return K clusters of data points.

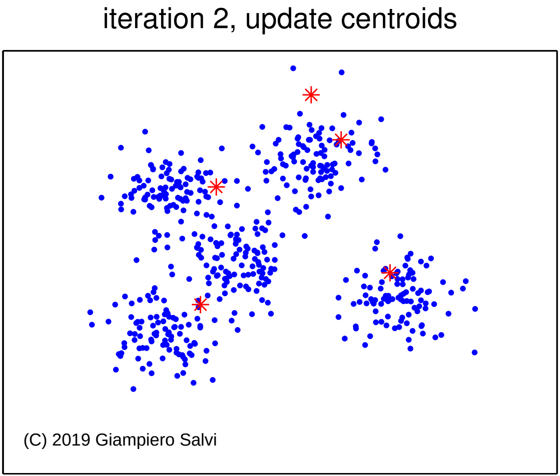
K-means: example



K-means: example

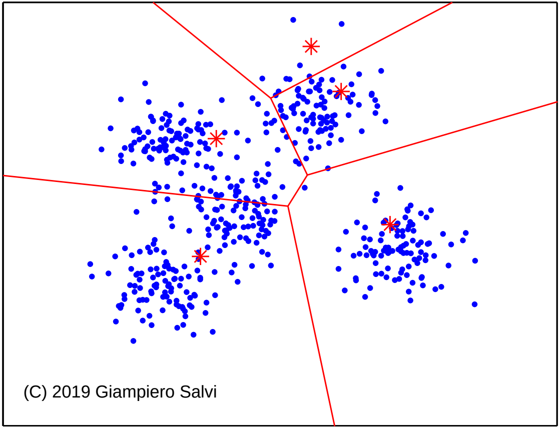


K-means: example

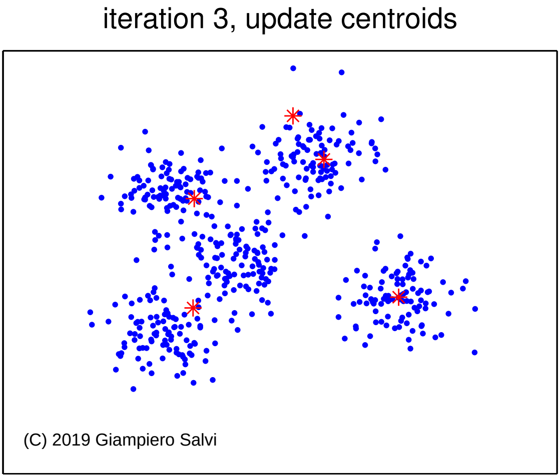


K-means: example

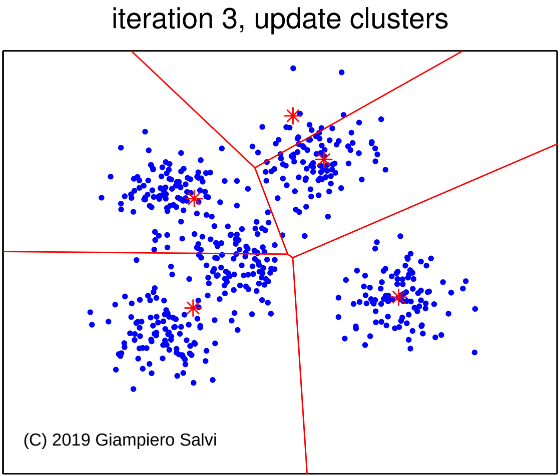
iteration 2, update clusters



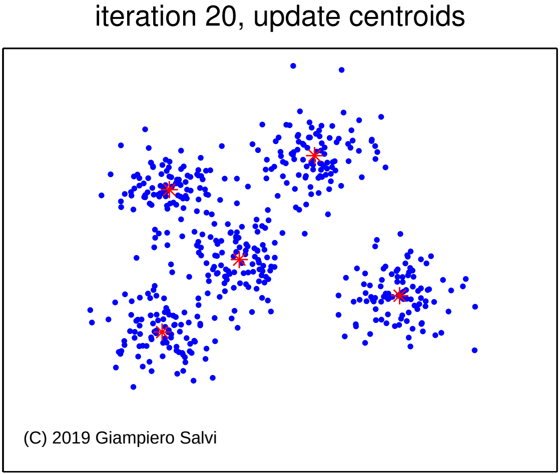
K-means: example



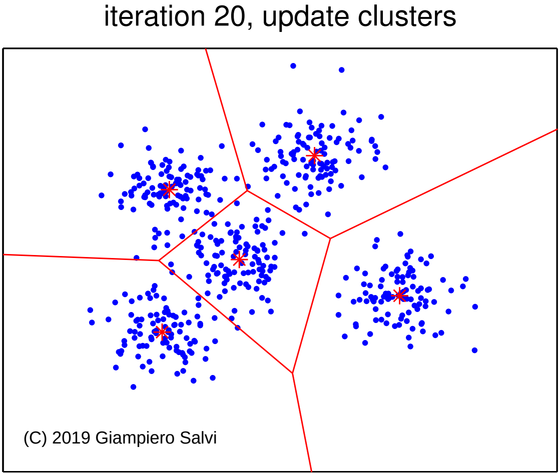
K-means: example



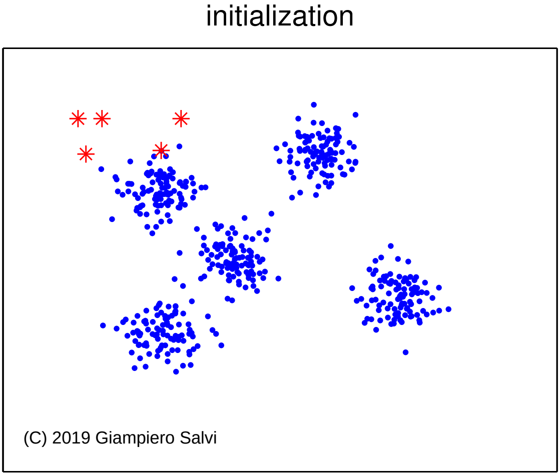
K-means: example



K-means: example

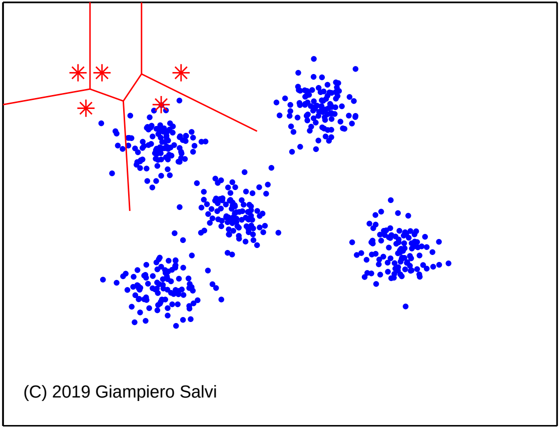


K-means: sensitivity to initial conditions

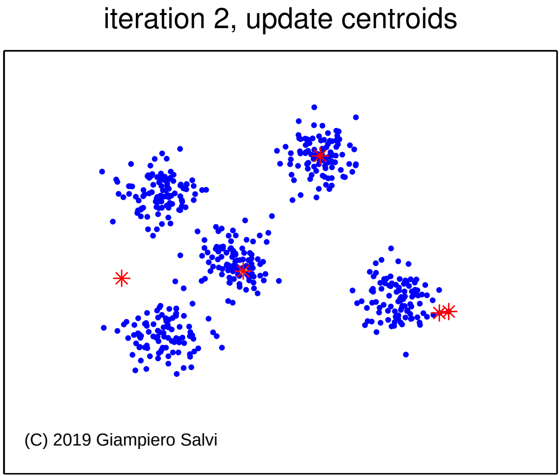


K-means: sensitivity to initial conditions

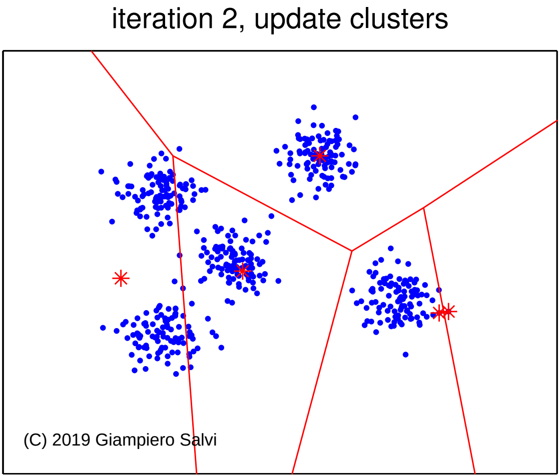
iteration 1, update clusters



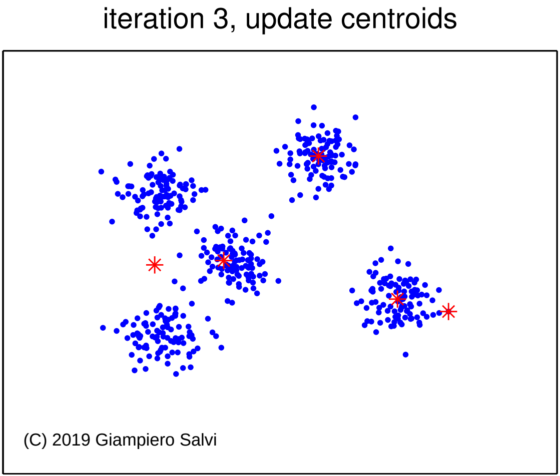
K-means: sensitivity to initial conditions



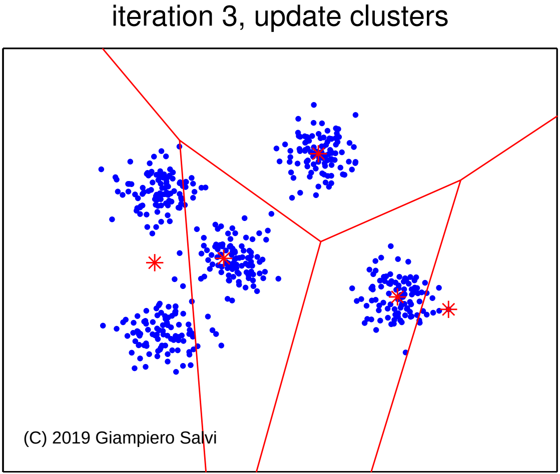
K-means: sensitivity to initial conditions



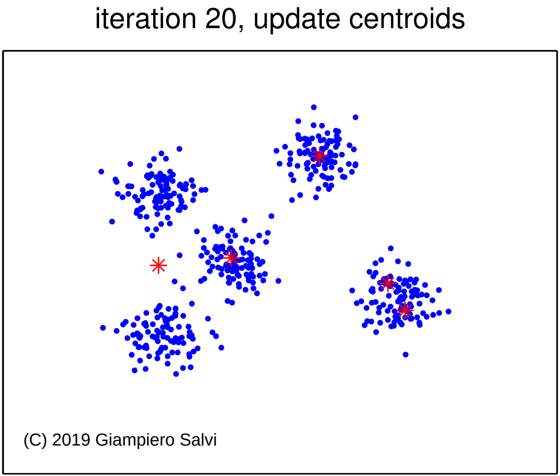
K-means: sensitivity to initial conditions



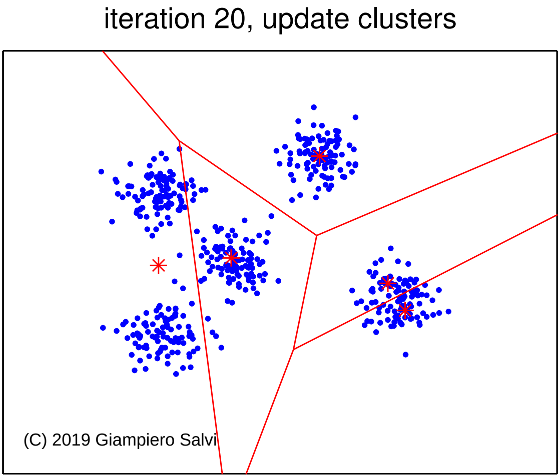
K-means: sensitivity to initial conditions



K-means: sensitivity to initial conditions



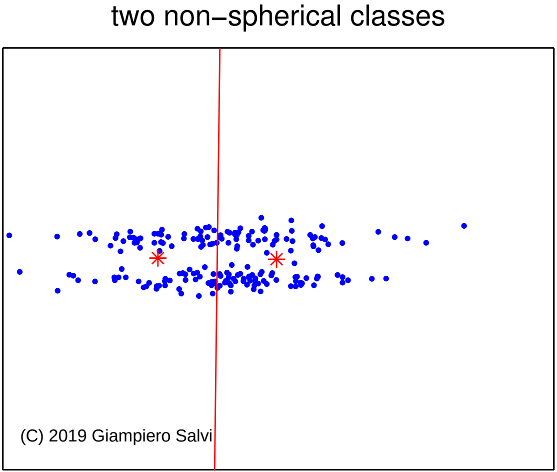
K-means: sensitivity to initial conditions



K-means: limits of Euclidean distance

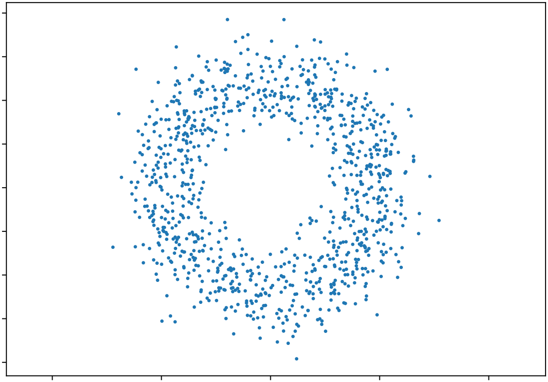
- the Euclidean distance is isotropic (same in all directions in $\mathbb{R}^p$)
- this favors spherical clusters
- the size of the clusters is controlled by the distances between them

K-means: non-spherical classes



Example: doughnut data

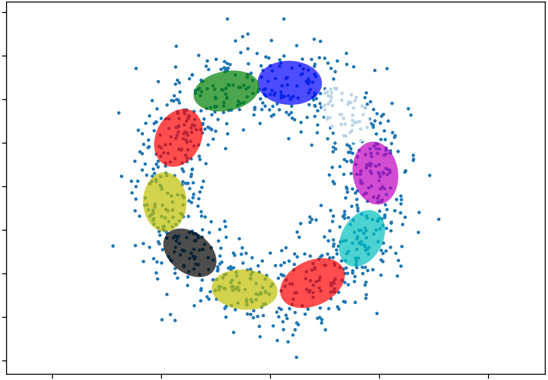
$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$



(C) 2019 Giampiero Salvi

Example: doughnut data

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$
$$Pr(\mathbf{x}|\theta_k), \forall k \in [1, K]$$

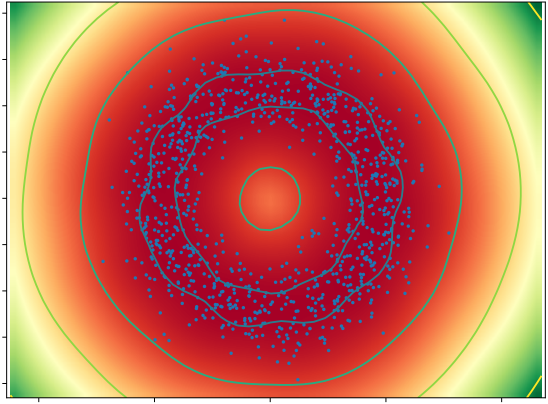


(C) 2019 Giampiero Salvi

Example: doughnut data

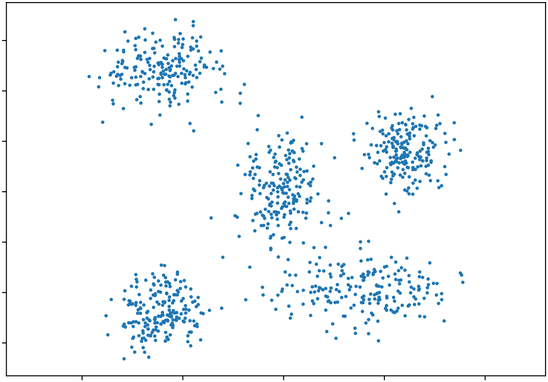
$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$

$$Pr(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k Pr(x|\theta_k)$$



Clustering Example

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$

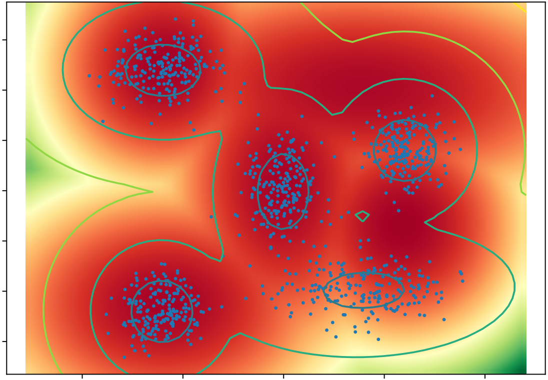


(C) 2019 Giampiero Salvi

Clustering Example

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$

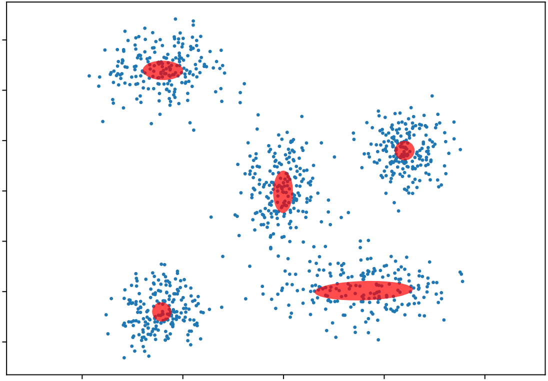
$$Pr(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k Pr(x|\theta_k)$$



(C) 2019 Giampiero Salvi

Clustering Example

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$
$$Pr(\mathbf{x}|\theta_k), \forall k \in [1, K]$$



(C) 2019 Giampiero Salvi

Fitting complex distributions

We can try to fit a mixture of K distributions:

$$Pr(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k Pr(\mathbf{x}|\theta_k)$$

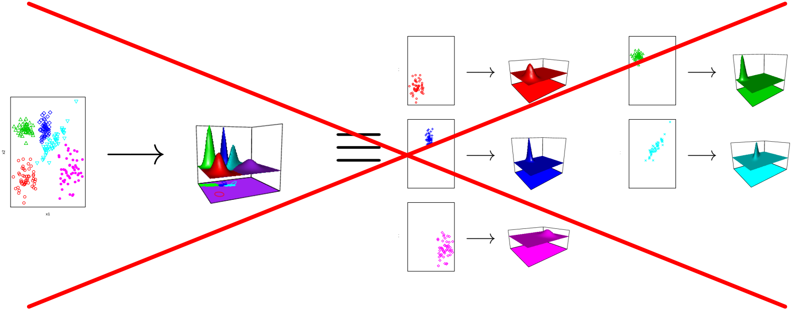
with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

Problem:

We do not know which point has been generated by which component of the mixture

We cannot optimize $Pr(\mathbf{x}|\theta)$ directly

No Class Independence Assumption



Solution: Expectation Maximization

Expectation Maximization

Fitting model parameters with missing (latent) variables

$$Pr(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k Pr(\mathbf{x}|\theta_k),$$

with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

- very general idea (applies to many different probabilistic models)
- augment data with latent variables: $h_i \in \{1, \ldots, K\}$ is the assignment of data point $\mathbf{x}_i$ to a component of the mixture
- optimize likelihood of the complete data over N data points

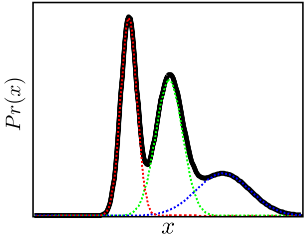
$$Pr(\mathbf{x}_1, \ldots, \mathbf{x}_N, h_1, \ldots, h_N|\theta)$$

Example: Mixture of Gaussians

This distribution is a weighted sum of K Gaussian distributions

$$Pr(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$$

where $\pi_1 + \cdots + \pi_K = 1$, $\pi_k > 0$ $(k = 1, \ldots, K)$.

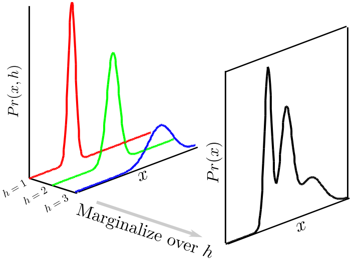


This model can describe **complex multi-modal** probability distributions by combining simpler distributions.

Mixture of Gaussians as a marginalization

We can interpret the Mixture of Gaussians model with the introduction of a discrete hidden/latent variable h and $Pr(x,h)$:

$$Pr(x) = \sum_{k=1}^{K} Pr(x, h=k) = \sum_{k=1}^{K} Pr(x \,|\, h=k) Pr(h=k)$$
$$= \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$$



← **mixture density**

Figures taken from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians

For each sample x_i introduce a *hidden variable* h_i

$$h_i = \begin{cases} 1 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x|\mu_1, \sigma_1^2) \\ 2 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x|\mu_2, \sigma_2^2) \end{cases}$$

and come up with initial values

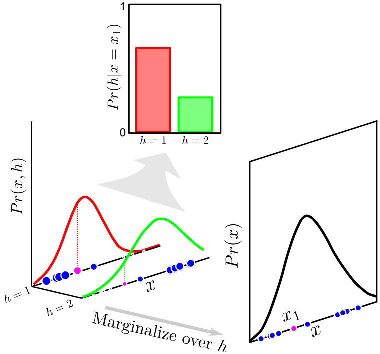
$$\Theta^{(0)} = (\pi_1^{(0)}, \mu_1^{(0)}, \sigma_1^{(0)}, \mu_2^{(0)}, \sigma_2^{(0)})$$

for each of the parameters.

EM is an *iterative algorithm* which updates $\Theta^{(t)}$ using the following two steps...

EM for two Gaussians: E-step

The responsibility of k-th Gaussian for each sample x (indicated by the size of the projected data point)



Look at each sample x along hidden variable h in the E-step

Figure from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians: E-step (cont.)

E-step: Compute the posterior probability (*responsibility*) that x_i was generated by component k given the current estimate of the parameters $\Theta^{(t)}$.

for each data point $i = 1, \ldots, N$

for each mixture component $k = 1, 2$

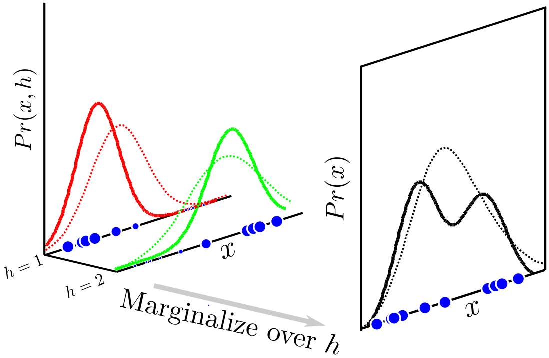
$$\gamma_{ik}^{(t)} := Pr(h_i = k \,|\, x_i, \Theta^{(t)}) = \frac{Pr(h_i = k, \Theta^{(t)}) Pr(x_i | h_i = k, \Theta^{(t)})}{Pr(x_i | \Theta^{(t)})}$$

$$= \frac{\pi_k^{(t)} \mathcal{N}(x_i | \mu_k^{(t)}, \sigma_k^{(t)})}{\pi_1^{(t)} \mathcal{N}(x_i | \mu_1^{(t)}, \sigma_1^{(t)}) + \pi_2^{(t)} \mathcal{N}(x_i | \mu_2^{(t)}, \sigma_2^{(t)})}$$

Note: *Responsibilities* $\gamma_{i1}^{(t)} + \gamma_{i2}^{(t)} = 1$ and mixture weights $\pi_1 + \pi_2 = 1$

EM for two Gaussians: M-step

Fitting the Gaussian model for each of k-th constinuent.
Sample x_i contributes according to the *responsibility* γ_{ik}.



(dashed and solid lines for fit before and after update)

Look along samples x for each h in the M-step

EM for two Gaussians: M-step (cont.)

M-step: Compute the maximum likelihood estimates of $\Theta^{(t)}$ given the data membership distributions (the $\gamma_{i1}^{(t)}$, $\gamma_{i2}^{(t)}$):

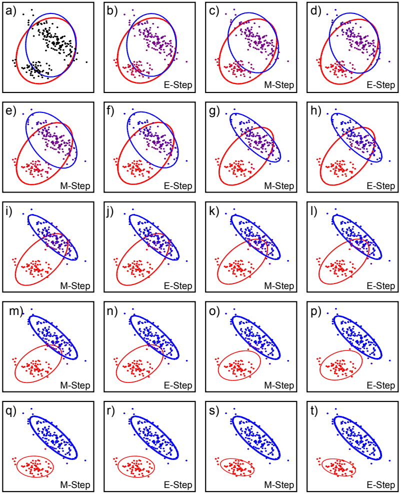
for $k = 1, 2$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^N \gamma_{ik}^{(t)} x_i}{\sum_{i=1}^N \gamma_{ik}^{(t)}}$$

$$\sigma_k^{(t+1)} = \sqrt{\frac{\sum_{i=1}^N \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^N \gamma_{ik}^{(t)}}}$$

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^N \gamma_{ik}^{(t)}}{N}$$

EM in practice



EM properties

Similar to K-means

- guaranteed to find a local maximum of the complete data likelihood
- somewhat sensitive to initial conditions

Better than K-means

- Gaussian distributions can model clusters with different shapes
- all data points are smoothly used to update all parameters
- Can include a prior $Pr(\Theta)$ to introduce regularization

Summary

1. Incorporating Priors
   - Maximum a Posteriori Estimation
   - Bayesian Non-Parametric Methods
   - Model Selection and Occam's Razor

2. Unsupervised Learning
   - Classification vs Clustering
   - Heuristic Example: K-means
   - Expectation Maximization