Probabilistic Artificial Intelligence

Solutions to Problem Set 6

Dec 8, 2019

1. Bayesian Linear Regression

You are given the following observations:

$$\mathbf{X} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 2.0 \\ 2.0 & 1.0 \\ 2.0 & 2.0 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 2.4 \\ 4.3 \\ 3.1 \\ 4.9 \end{bmatrix}$$

Assume the generative model is $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$ and $\sigma_y^2 = 0.1$.

$\hat{w} = \arg\max \sum_{i=1}^{4} \log(1+\exp(-y_i w^\top x_i))$

1. Find the maximum likelihood estimate $\mathbf{w}_{MLE}$ given the data.
2. Now assume we have a prior over the weights $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$ with $\sigma_w^2 = 0.05$. Find the maximum *a posteriori* (MAP) estimate $\mathbf{w}_{MAP}$ given the data and the prior.
3. Use the predictive posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{y})$ to get a prediction for $p(\mathbf{y}^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \mathcal{N}(\mu_{y^*}, \sigma_{y^*}^2)$ with $\mathbf{x}^* = \begin{bmatrix} 3 & 3 \end{bmatrix}$. Report μ_{y^*} and $\sigma_{y^*}^2$.
4. How would you have to change the prior $p(\mathbf{w})$, such that $\mathbf{w}_{MAP} \to \mathbf{w}_{MLE}$?

$\begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{bmatrix}$

$\begin{bmatrix} 10 & 9 \\ 9 & 10 \end{bmatrix}$

Solution

1. We compute the maximum likelihood estimate $\mathbf{w}_{MLE}$ as follows:

$\frac{1}{100-81} \begin{bmatrix} 10 & -9 \\ -9 & 10 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 2 \end{bmatrix}$

$\frac{1}{19}$

$$\mathbf{w}_{MLE} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = [0.63 \;\; 1.83]$$

$\frac{1}{19} \begin{bmatrix} 1 & -8 & 11 & 2 \\ 1 & 11 & -8 & 2 \end{bmatrix} \begin{bmatrix} 2.4 \\ 4.3 \\ 3.1 \\ 4.9 \end{bmatrix}$

$[0.63 \; 1.83]$

2. Given that

$P(W \mid X, y) = P(W) \; \hat{W}$

$$p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{w}; \mu_{\mathbf{w}}, \Sigma_{\mathbf{w}})$$

$$\textit{with } \Sigma_{\mathbf{w}} = \left(\frac{1}{\sigma_w^2} \mathbf{I} + \frac{1}{\sigma_y^2} \mathbf{X}^\top \mathbf{X} \right)^{-1}$$

$$\mu_{\mathbf{w}} = \frac{1}{\sigma_y^2} \Sigma_{\mathbf{w}} \mathbf{X}^\top \mathbf{y}$$

With $\sigma_y^2 = 0.1$ and $\sigma_w^2 = 0.05$, we compute

$$\Sigma_{\mathbf{w}} = \begin{bmatrix} 0.019 & -0.014 \\ -0.014 & 0.019 \end{bmatrix}.$$

Then $\mathbf{w}_{MAP}$:

$$\mathbf{w}_{MAP} = \frac{1}{\sigma_y^2}\Sigma_{\mathbf{w}}\mathbf{X}^\top \mathbf{y} = [0.91 \;\; 1.31]$$

3. Given

$$\begin{aligned} p(y^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \mathcal{N}(y^*; \mu_y^*, \sigma_y^{2*}) \\ \textit{with} \;\; \sigma_{y^*}^2 = \sigma_y^2 + \mathbf{x}^{*\top}\Sigma_{\mathbf{w}}^*\mathbf{x}^* \\ \mu_{y^*} = \mathbf{x}^{*\top}\mu_{\mathbf{w}}^* \end{aligned}$$

Given $x^* = \begin{bmatrix} 3 & 3 \end{bmatrix}$ and

$$\Sigma_{\mathbf{w}} = \begin{bmatrix} 0.019 & -0.014 \\ -0.014 & 0.019 \end{bmatrix}$$

and $\mathbf{w}_{MAP} = [0.91 \;\; 1.31]$, then

$$\begin{aligned} \mu_{y^*} &= \mathbf{x}^{*\top}\mathbf{w}_{MAP} = 6.66 \\ \sigma_{y^*}^2 &= \sigma_y^2 + \mathbf{x}^{*\top}\Sigma_{\mathbf{w}}\mathbf{x}^* = 0.186 \end{aligned}$$

4. We would have to let $\sigma_w^2 \to \infty$.

2. Gaussian Processes

A Gaussian process is a prior over functions $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, where $m(x) = \mathbb{E}[f(x)]$ is called a *mean function* and $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ is called a *kernel function*. The predictive posterior is

$$\begin{aligned} & p(\mathbf{y}^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \mathcal{N}(\mu_y^*, \sigma_y^{2*}) \\ \text{with} \quad & \mu_y^* = \mathbf{k}(\mathbf{X}, \mathbf{x}^*)^\top \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_y^2\mathbf{I}\right]^{-1}\mathbf{y} \\ \text{and} \quad & \sigma_y^{2*} = \sigma_y^2 + k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{X}, \mathbf{x}^*)^\top \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_y^2\mathbf{I}\right]^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}^*) \end{aligned}$$

where $\mathbf{k}(\mathbf{X}, \mathbf{x}^*)$ is a column vector with elements $\mathbf{k}(\mathbf{X}, \mathbf{x}^*)_i = k(\mathbf{x}_i, \mathbf{x}^*)$ and $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is a symmetric square matrix with elements $\mathbf{K}(\mathbf{X}, \mathbf{X})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Show that the prior $p(\mathbf{w})$ from task 1 is equivalent to a GP prior.

Hint: Choose $m(\mathbf{x}) = 0$ and $k(\mathbf{x}, \mathbf{x}') = \lambda\mathbf{x}^\top\mathbf{x}'$, and find a λ such that the predictive posteriors match.

Hint 2: The Searle identities might be useful [Matrix cookbook, Section 3.2.5].

Solution

Proof. Given the following definitions for Bayesian linear regression (BLR) and Gaussian processes (GP)

$$\text{BLR:}\quad \mu_y^* = x^{*\top}\frac{1}{\sigma_y^2}\left(\frac{1}{\sigma_{\mathbf{w}}^2}+\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$

$$\text{GP:}\quad \mu_y^* = \mathbf{k}(\mathbf{X},\mathbf{x}^*)^\top\left(\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_y^2\mathbf{I}\right)^{-1}\mathbf{y}$$

we first show that $\mu_{y,\text{ BLR}}^* = \mu_{y,\text{ GP}}^*$.

Let $\mathbf{k}(\mathbf{x},\mathbf{x}') = \lambda\mathbf{x}^\top\mathbf{x}'$. Then

$$\begin{aligned}
\mu_{y,\text{ GP}}^* &= \lambda\mathbf{x}^{*\top}\mathbf{X}^\top(\mathbf{X}\mathbf{X}\mathbf{X}^\top+\sigma_y^2\mathbf{I})^{-1}\mathbf{y} \\
&= \frac{\lambda}{\sigma_y^2}\mathbf{x}^{*\top}\mathbf{X}^\top(\mathbf{I}+\frac{\mathbf{x}}{\sigma_y^2}\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y} && \text{Matrix Cookbook Eq. 162} \\
&= \mathbf{x}^{*\top}(\frac{\sigma_y^2}{\mathbf{x}}\mathbf{I}+\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\
&= \mathbf{x}^{*\top}\frac{1}{\sigma_y^2}\left(\frac{1}{\lambda}\mathbf{I}+\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y} && \text{Choose } \lambda=\sigma_{\mathbf{w}}^2 \\
&= \mu_{y,\text{ BLR}}^*.
\end{aligned}$$

Given

$$\text{BLR:}\quad \sigma_y^{2*} = \sigma_y^2 + \mathbf{x}^{*\top}\left(\frac{1}{\sigma_{\mathbf{w}}^2}+\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}^*$$

$$\text{GP:}\quad \sigma_y^{2*} = \sigma_y^2 + k(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{k}(\mathbf{X},\mathbf{x}^*)^\top\left(\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_y^2\mathbf{I}\right)^{-1}\mathbf{k}(\mathbf{X},\mathbf{x}^*)$$

we now show that $\sigma_{y,\text{ BLR}}^{2*} = \sigma_{y,\text{ GP}}^{2*}$.

$$\begin{aligned}
\sigma_{y,\text{ GP}}^{2*} &= \sigma_y^2 + \lambda\mathbf{x}^{*\top}\mathbf{x}^* - \lambda^2\mathbf{x}^{*\top}\mathbf{X}^\top(\lambda\mathbf{X}\mathbf{X}^\top\sigma_y^2\mathbf{I})^{-1}\mathbf{X}\mathbf{x}^* \\
&= \sigma_y^2 + \lambda\mathbf{x}^{*\top}\left(\mathbf{I}-\lambda\mathbf{X}^\top(\lambda\mathbf{X}\mathbf{X}^\top+\sigma_y^2\mathbf{I})^{-1}\right)\mathbf{x}^* \\
&= \sigma_y^2 + \lambda\mathbf{x}^{*\top}\left(\mathbf{I}-\frac{\lambda}{\sigma_y^2}(\mathbf{I}+\frac{\lambda}{\sigma_y^2}\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\right)\mathbf{x}^* && \text{Matrix Cookbook Eq. 166} \\
&= \sigma_y^2 + \lambda\mathbf{x}^{*\top}\left(\mathbf{I}+\frac{\lambda}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}^* \\
&= \sigma_y^2 + \mathbf{x}^{*\top}\left(\frac{1}{\lambda}\mathbf{I}+\frac{1}{\sigma_y^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}^* && \text{Choose } \lambda=\sigma_{\mathbf{w}}^2 \\
&= \sigma_{y,\text{ BLR}}^{2*}
\end{aligned}$$

$\square$

3. Q-Learning

Assume this grid world, where the agent always starts in state s_{11} and $\{s_{13}, s_{31}, s_{33}\}$ are terminal states. The action space is $\mathcal{A} = \{\texttt{right}, \texttt{left}, \texttt{down}, \texttt{up}\}$ and the transitions are noiseless. Reaching s_{33} yields a reward of 1.0, while reaching s_{13} or s_{31} yields -1.0 reward. The other states do not yield any rewards.

Assume that we initialize all Q-values in non-terminal states with 0.5 and in terminal states with 0. Unless otherwise stated, use a discount factor of $\gamma = 0.5$ and a step size of $\alpha = 0.5$.

1. We observe two episodes:

$$s_{11} \to s_{12} \to s_{13}$$
$$s_{11} \to s_{12} \to s_{22} \to s_{32} \to s_{33}$$

   Apply Q-learning to these episodes. Show the updates to the Q-values at each step.

2. We have run two Q-learning algorithms to convergence with different discount factors γ_1 and γ_2. The first one gives us $Q_1(s_{11}, \texttt{right}) = 0.9$, the second one $Q_1(s_{11}, \texttt{right}) = 0.7$. What can you say about the relative magnitudes of γ_1 and γ_2?

3. How many parameters do we have to learn for the complete tabular Q-function in this problem? How many would we need to learn for a model-based RL approach (i.e., if we would learn the transition probabilities $P(s^{t+1} \mid s^t, a^t)$)? Which approach is more efficient? 9x4
   9x

Solution

1. Given the update rule of Q-learning with state s, action a and reward r defined as

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{\alpha'} Q(s', a')),$$

we apply Q-learning to the episodes as follows:

$$Q(s_{11}, \texttt{right}) \leftarrow 0.5 \times Q(s_{11}, \texttt{right}) + 0.5 \times (0 + 0.5 \times 0.5) = 0.375$$
$$Q(s_{12}, \texttt{right}) \leftarrow 0.5 \times Q(s_{12}, \texttt{right}) + 0.5 \times (-1 + 0.5 \times 0) = -0.25$$

and

$$Q(s_{11}, \texttt{right}) \leftarrow 0.5 \times 0.375 + 0.5 \times (0 + 0.5 \times 0.5) = 0.3125$$
$$Q(s_{12}, \texttt{down}) \leftarrow 0.5 \times Q(s_{12}, \texttt{down}) + 0.5 \times (0 + 0.5 \times 0.5) = 0.375$$
$$Q(s_{22}, \texttt{down}) \leftarrow 0.5 \times Q(s_{22}, \texttt{down}) + 0.5 \times (0 + 0.5 \times 0.5) = 0.375$$
$$Q(s_{32}, \texttt{right}) \leftarrow 0.5 \times Q(s_{32}, \texttt{right}) + 0.5 \times (1 + 0.5 \times 0) = 0.75$$

The resulting Q-table is

	right	left	down	up
s_{11}	0.3125	0.5	0.5	0.5
s_{12}	-0.25	0.5	0.375	0.5
s_{13}	0	0	0	0
s_{21}	0.5	0.5	0.5	0.5
s_{22}	0.5	0.5	0.375	0.5
s_{23}	0.5	0.5	0.5	0.5
s_{31}	0	0	0	0
s_{32}	0.75	0.5	0.5	0.5
s_{33}	0	0	0	0

2. $\gamma_1 > \gamma_2$.

3. The model-based reinforcement learning approach we have to learn $|X|^2 \times |A| = 324$ parameters. The tabular Q-function has $|X| \times |A| = 36$ values. Q-learning has a time complexity of $O(|A|)$ and a space complexity of $O(|X| \times |A|)$ and is thus the more efficient approach.