8.2

(a) The projection of the input vector $\mathbf{x}$ onto the weight vector $\mathbf{w}$ is defined by the inner product

$$\mathbf{x}^{\mathsf{T}}\mathbf{w} = \mathbf{w}^{\mathsf{T}}\mathbf{x}.$$

With $\mathbf{x}$ being a random vector, this inner product is clearly a random variable whose variance is defined by the expectation $\mathbb{E}\big[\big(\mathbf{x}^{\mathsf{T}}\mathbf{w}\big)^2\big]$, where it is recognized that $\mathbf{x}$ is supposed to have zero mean. This expectation can be expressed as follows:

$$\begin{aligned}\sigma^2 &= \mathbb{E}\big[\big(\mathbf{x}^{\mathsf{T}}\mathbf{w}\big)^2\big] \\ &= \mathbb{E}\big[\big(\mathbf{w}^{\mathsf{T}}\mathbf{x}\big)\big(\mathbf{x}^{\mathsf{T}}\mathbf{w}\big)\big] \\ &= \mathbf{w}^{\mathsf{T}}\mathbb{E}\big[\mathbf{x}\mathbf{x}^{\mathsf{T}}\big]\mathbf{w} \\ &= \mathbf{w}^{\mathsf{T}}\mathbf{R}\mathbf{w}\end{aligned}$$

where the weight vector $\mathbf{w}$ is treated as "fixed" and

$$\mathbf{R} = \mathbb{E}\big[\mathbf{x}\mathbf{x}^{\mathsf{T}}\big]$$

is the correlation function of the input vector $\mathbf{x}$.

Imposing the constraint that the weight vector $\mathbf{w}$ satisfies the condition

$$\mathbf{w}^{\mathsf{T}}\mathbf{w} = 1$$

we are, in effect, applying "normalization" the weight vector. Hence, maximization of the variance σ^2, subject to this constraint, yields the Lagrangian

$$J(\mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{R}\mathbf{w} - \lambda(\mathbf{w}^{\mathsf{T}}\mathbf{w} - 1)$$

where λ is the Lagrange multiplier

(b) Differentiating the Lagrangian $J(\mathbf{w})$ with respect to the weight vector $\mathbf{w}$ yields

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = 2\mathbf{R}\mathbf{w} - 2\lambda\mathbf{w} \tag{1}$$

Setting this partial derivative to zero, we get

$$\mathbf{R}\mathbf{w} = \lambda\mathbf{w}$$

Eqn. (1) is how eigendecomposition is usually formulated with λ playing the role of an eigenvalue of the correlation matrix $\mathbf{R}$ and $\mathbf{w}$ playing the role of associated eigenvector.

(c) With the input vector $\mathbf{x}$ having the dimension m, the correlation matrix $\mathbf{R}$ is an m-by-m matrix with a total of m eigenvalues. In other words, there are m possible Lagrange multipliers to be considered. In eigendecomposition, the eigenvectors are form an orthonormal set, with each eigenvector normalized to have a Euclidean norm of unity. Hence, expanding the scope of the Lagrangian

to accommodate the essence of the eigendecomposition, we may expand the cost function of (1) as follows:

$$J(\mathbf{w}_i) = \mathbf{w}_i^{\mathsf{T}}\mathbf{R}\mathbf{w}_i - \lambda_{ii}(\mathbf{w}_i^{\mathsf{T}}\mathbf{w}_i - 1) - \sum_{j=1}^{i-1}\lambda_{ij}\mathbf{w}_i^{\mathsf{T}}\mathbf{w}_j \tag{2}$$

where $i = 1, 2, \ldots, m$. The expanded composition of the Lagrangian in (2) is explained as follows:

1. There are m eigenvectors, denoted by $\mathbf{w}_i$, $i = 1, 2, \ldots, m$.
2. The eigenvectors form an orthonormal set, as already justified, which is taken care of by introducing a corresponding set of Lagrange multipliers, as described here:
   - λ_{ii} assigned to the "eigenvector" $\mathbf{w}_i$; and
   - λ_{ij} assigned to the inner product $\mathbf{w}_i^{\mathsf{T}}\mathbf{w}_j$.

Now that we have explained the composition of the cost function $J(\mathbf{w}_i)$ on (2), we may differentiate both sides of it with respect to $\mathbf{w}_i$, obtaining

$$\frac{\partial J(\mathbf{w}_i)}{\partial \mathbf{w}_i} = 2\mathbf{R}\mathbf{w}_i - 2\lambda_{ii}\mathbf{w}_i - \sum_{j=1}^{i-1}\lambda_{ij}\mathbf{w}_j$$

Setting this partial derivation to zero yields the relationship:

$$(\mathbf{R} - \lambda_{ii}\mathbf{I})\mathbf{w}_i = \frac{1}{2}\sum_{j=1}^{i-1}\lambda_{ij}\mathbf{w}_j, \quad i = 1, 2, \ldots, m. \tag{3}$$

The optimal solution of this set of m simultaneous equations is realized only when

$$\lambda_{ij} = 0 \quad \text{for all } j = 1, 2, \ldots, i-1.$$

That is, the $\mathbf{w}_i$ form an orthogonal set for $i = 1, 2, \ldots, m$; under this condition, (3) simplifies to

$$\mathbf{R}\mathbf{w}_i = \lambda_{ii}\mathbf{w}_i, \quad i = 1, 2, \ldots, m,$$

with the two conditions of orthonormality defined by

$$\mathbf{w}_i^{\mathsf{T}}\mathbf{w}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

8.4

(a) Consider the Lagrangian

$$J(\mathbf{w}) = (\mathbf{w}^{\mathsf{T}}\mathbf{x})^2 - \lambda(\mathbf{w}^{\mathsf{T}}\mathbf{w} - 1).$$

The gradient of $J(\mathbf{w})$ with respect to the weight vector $\mathbf{w}$ is defined by

$$\begin{aligned} g(\mathbf{w}) &= \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) \\ &= \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^{\mathsf{T}} \mathbf{x}\mathbf{x}^{\mathsf{T}} \mathbf{w} - \lambda(\mathbf{w}^{\mathsf{T}} \mathbf{w} - 1)) \\ &= 2\mathbf{x}\mathbf{x}^{\mathsf{T}} \mathbf{w} - 2\lambda \mathbf{w}. \end{aligned} \tag{4}$$

With the inner product $\mathbf{x}^{\mathsf{T}}\mathbf{w} = \mathbf{w}^{\mathsf{T}}\mathbf{x}$ representing a scaling factor, we may clearly express the matrix product $\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w}$ in the equivalent form $\mathbf{w}^{\mathsf{T}}\mathbf{x}\mathbf{x}$. Hence, we may rewrite (4) as

$$\begin{aligned} g(\mathbf{w}) &= 2(\mathbf{w}^{\mathsf{T}}\mathbf{x})\mathbf{x} - 2\lambda \mathbf{w} \\ &= 2\mathbf{x}(\mathbf{x}^{\mathsf{T}}\mathbf{w}) - 2\lambda \mathbf{w} \end{aligned}$$

which is the desired result.

(b) Setting the gradient $g(\mathbf{w}) = \mathbf{0}$, we obtain

$$\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w} = \lambda \mathbf{w},$$

where the outer product $\mathbf{x}\mathbf{x}^{\mathsf{T}}$ plays the role of a matrix representing the instantaneous value of the correlation matrix, and λ plays the role of eigenvalue associated with eigenvector $\mathbf{w}$. Multiplying both sides of this equation by $\mathbf{w}^{\mathsf{T}}$, we have

$$\mathbf{w}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w} = \lambda \mathbf{w}^{\mathsf{T}}\mathbf{w}.$$

Now, imposing the normalizing condition

$$\mathbf{w}^{\mathsf{T}}\mathbf{w} = 1 \quad \text{for all } \mathbf{w},$$

the eigenvalue λ assumes the value

$$\lambda = \mathbf{w}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w} \tag{5}$$

Next, eliminating λ between (4) and (5) yields

$$g(\mathbf{w}) = 2\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w} - 2\mathbf{w}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{w}\mathbf{w}. \tag{6}$$

For $\mathbf{x} = \mathbf{x}(n)$ and $\mathbf{w} = \hat{\mathbf{w}}(n)$, 6 takes the form

$$g(\hat{\mathbf{w}}(n)) = 2\mathbf{x}(n)\mathbf{x}^{\mathsf{T}}(n)\hat{\mathbf{w}}(n) - 2\hat{\mathbf{w}}^{\mathsf{T}}(n)\mathbf{x}(n)\mathbf{x}^{\mathsf{T}}(n)\hat{\mathbf{w}}(n)\hat{\mathbf{w}}(n)$$

Hence, using this instantaneous gradient value in the weight update formula

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{1}{2}\eta g\hat{\mathbf{w}}(n))$$

we obtain the recursive formula

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta[\mathbf{x}(n)\mathbf{x}^{\mathsf{T}}(n)\hat{\mathbf{w}}(n) - \hat{\mathbf{w}}^{\mathsf{T}}(n)\mathbf{x}(n)\mathbf{x}^{\mathsf{T}}(n)\hat{\mathbf{w}}(n)\hat{\mathbf{w}}(n)]$$

which is the desired result.

9.5

The batch version of the SOM algorithm is defined by

$$\mathbf{w}_j = \frac{\sum_i \pi_{j,i}\mathbf{x}_i}{\sum_i \pi_{j,i}} \quad \text{for some prescribed neuron } j \tag{7}$$

where $\pi_{j,i}$ is the discretized version of the pdf $\pi(\nu)$ of noise vector ν. The $\pi_{j,i}$ plays a role analogous to that of the neighborhood function $h_{j,i(\mathbf{x})}$. Indeed, we may substitute $h_{j,i(\mathbf{x})}$ for $\pi_{j,i}$ in (7). We are interested in rewriting (7) in a form that highlights the role of the so-called Voronoi cells; such cells located about a set of points in the input space correspond to a partition of that space according to the nearest-neighbor rule based on the Euclidean metric. To this end we note that the dependence of the neighborhood function $h_{j,i(\mathbf{x})}$ and therefore $\pi_{j,i}$ on the input pattern $\mathbf{x}$ is indirect, with the dependence being through he Voronoi cell in which $\mathbf{x}$ lies. Hence for all input patterns that lie in a particular Voronoi cell the same neighborhood function applies. Let each Voronoi cell be identified by an indicator function $I_{i,k}$ interpreted as follows:

$I_{i,k} = 1$ if the input pattern $\mathbf{x}_i$ lies in the Voronoi cell corresponding to winning neuron k. Then in light of these considerations we may rewrite (7) in the new form

$$\mathbf{w}_j = \frac{\sum_k \pi_{j,k} \sum_i I_{i,k}\mathbf{x}_i}{\sum_k \pi_{j,k} \sum_i I_{i,k}} \tag{8}$$

Now let $\mathbf{m}_k$ denote the centroid of the Voronoi cell of neuron k and N_k denote the number of input patterns that lie in that cell. We may then simplify (8) as

$$\mathbf{w}_j = \frac{\sum_k \pi_{j,k} N_k \mathbf{m}_k}{\sum_k \pi_{j,k} N_k} = \sum_k W_{j,k}\mathbf{m}_k \tag{9}$$

where $W_{j,k}$ is a weighting function defined by

$$W_{j,k} = \frac{\pi_{j,k}N_k}{\sum_k \pi_{j,k}N_k} \quad \sum_k W_{j,k} = 1 \quad \text{for all } j$$

Eqn. (9) bears a close resemblance to the Watson-Nadaraya regression estimator defined in Eq. (5.61) of the textbook. Indeed, in light of this analogy, we may offer the following observations:

- The SOM algorithm is similar to nonparametric regression in a statistical sense.
- Except for the normalizing factor N_k, the discretized pdf $\pi_{j,i}$ and therefore the neighborhood function $h_{j,i}$ plays the role of a kernel in the Watson-Nadaraya estimator.
- The width of the neighborhood function plays the role of the span of the kernel.

9.6

In its basic form, Hebbs postulate of learning states that the adjustment Δw_{kj} applied to the synaptic weight w_{kj} is defined by

$$\Delta w_{kj} = \eta y_k x_j$$

where y_k is the output signal produced in response to the input signal x_j, and η is the learning-rate parameter.

The weight update for the maximum eigenfilter includes the term $\eta y_k x_j$ and, additionally, a stabilizing term defined by $-y_k^2 w_{kj}$. The term $\eta y_k x_j$ provides amplification

In contrast, in the SOM algorithm two modifications are made to Hebbs postulate of learning:

1. The stabilizing term is set equal to $-y_k w_{kj}$.
2. The output y_k of neuron k is set equal to a neighborhood function.

The net result of these two modifications is to make the weight update for the SOM algorithm assume a form similar to that in competitive learning rather than Hebbian learning.

10.4

Consider a multilayer perceptron with a single hidden layer. Let w_{ji} denote the synaptic weight of hidden neuron j connected to source node i in the input layer. Let $x_{i|\alpha}$ denote the i-th component of the input vector $\mathbf{x}$, given example α. Then the induced local field of neuron j is

$$v_{j|\alpha} = \sum_i w_{ji} x_{i|\alpha}$$

Correspondingly, the output of hidden neuron j for example α is given by

$$y_{j|\alpha} = \varphi(v_{j|\alpha})$$

where $\varphi(\cdot)$ is the logistic function $\varphi(v) = \frac{1}{1+e^{-v}}$

Consider next the output layer of the network. Let w_{kj} denote the synaptic weight of output neuron k connected to hidden neuron j. The induced local field of output neuron k is

$$v_{k|\alpha} = \sum_j w_{kj} y_{j|\alpha}$$

The k-th output of the network is therefore

$$y_{k|\alpha} = \varphi(v_{k|\alpha})$$

the output $y_{k|\alpha}$ is assigned a probabilistic interpretation by writing $y_{k|\alpha} = p_{k|\alpha}$. Accordingly, we may view $y_{k|\alpha}$ as an estimate of the conditional probability that the proposition k is true, given the example α at the input. On this basis, we may interpret

$$1 - y_{k|\alpha} = 1 - p_{k|\alpha}$$

as the estimate of the conditional probability that the proposition k is false, given the input example α. Correspondingly, let $q_{k|\alpha}$ denote the actual (true) value of the conditional probability that the proposition k is true, given the input example α. This means that $(1 - q_{k|\alpha})$ is the actual value of the conditional probability that the proposition k is false, given the input example α. Thus, we may define the Kullback-Leibler divergence for the multilayer perceptron as

$$D_{p\|q} = \sum_{\alpha} p_{\alpha} \sum_{k} \left[q_{k|\alpha} \log \frac{q_{k|\alpha}}{p_{k|\alpha}} + (1 - q_{k|\alpha}) \log \frac{1 - q_{k|\alpha}}{1 - p_{k|\alpha}} \right]$$

where p_{α} is the a priori probability of occurrence of example α at the input.

To perform supervised training of the multilayer perceptron, we use gradient descent on $D_{p\|q}$ in the weight space. First, we use the chain rule to express the partial derivative of $D_{p\|q}$ with respect to the synaptic weight w_{kj} of the output neuron k as follows:

$$\frac{\partial D_{p\|q}}{\partial w_{kj}} = \frac{\partial D_{p\|q}}{\partial p_{k|\alpha}} \frac{\partial p_{k|\alpha}}{\partial y_{k|\alpha}} \frac{\partial y_{k|\alpha}}{\partial v_{k|\alpha}} \frac{\partial v_{k|\alpha}}{\partial w_{kj}} = -\sum_{\alpha} p_{\alpha} (q_{k|\alpha} - p_{k|\alpha}) y_{j|\alpha}$$

Next, we express the partial derivative of $D_{p\|q}$ with respect to the synaptic weight w_{ji} of hidden neuron j by writing

$$\frac{\partial D_{p\|q}}{\partial w_{ji}} = -\sum_{\alpha} p_{\alpha} \sum_{\alpha} \left(\frac{q_{k|\alpha}}{p_{k|\alpha}} - \frac{1 - q_{k|\alpha}}{1 - p_{k|\alpha}} \right) \frac{\partial p_{k|\alpha}}{\partial w_{ji}} \tag{10}$$

Via the chain rule, we go on to write

$$\frac{\partial p_{k|\alpha}}{\partial w_{ji}} = \frac{\partial p_{k|\alpha}}{\partial y_{k|\alpha}} \frac{\partial y_{k|\alpha}}{\partial v_{k|\alpha}} \frac{\partial v_{k|\alpha}}{\partial y_{j|\alpha}} \frac{\partial y_{j|\alpha}}{\partial v_{j|\alpha}} \frac{\partial v_{j|\alpha}}{\partial w_{ji}} = \varphi'(v_{k|\alpha}) w_{kj} \varphi'(v_{j|\alpha}) x_{i|\alpha} \tag{11}$$

But

$$\varphi'(v_{k|\alpha}) = y_{k|\alpha}(1 - y_{k|\alpha}) = p_{k|\alpha}(1 - p_{k|\alpha}) \tag{12}$$

Hence, using (11) and (12) we may simplify (10) as

$$\frac{\partial D_{p\|q}}{\partial w_{kj}} = -\sum_{\alpha} p_{\alpha} x_{i|\alpha} \varphi' \left(\sum_{i} w_{ji} x_{i|\alpha} \right) \sum_{k} (p_{k|\alpha} - q_{k|\alpha}) w_{kj}$$

where $\varphi'(\cdot)$ is the derivative of the logistic function $\varphi(\cdot)$ with respect to its argument.

Assuming the use of the learning-rate parameter for all weight changes applied to the network, we may use the method of steepest descent to write the following two-step probabilistic algorithm:

1. For output neuron k, compute

$$\begin{aligned}\Delta w_{kj} &= -\eta \frac{\partial D_{p\|q}}{\partial w_{kj}} \\ &= \eta \sum_{\alpha} p_{\alpha}(q_{k|\alpha} - p_{k|\alpha}) y_{j|\alpha}\end{aligned}$$

2. For hidden neuron j, compute

$$\begin{aligned}\Delta w_{ji} &= -\eta \frac{\partial D_{p\|q}}{\partial w_{ji}} \\ &= \eta \sum_{\alpha} p_{\alpha} x_{i|\alpha} \varphi'\Big(\sum_{i} w_{ji} x_{i|\alpha}\Big) \sum_{k} (p_{k|\alpha} - q_{k|\alpha}) w_{kj}\end{aligned}$$

10.29

(a) The functional F, in its expanded form, is defined by (see Eq. (10.86) of the textbook)

$$F(\gamma(\mathbf{t}), q(\mathbf{t}|\mathbf{x})) =$$
$$\int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} \Big\{ p(\mathbf{x}) q(\mathbf{t}|\mathbf{x}) \|\mathbf{x} - \gamma(\mathbf{t})\|^2 + \lambda p(\mathbf{x}) q(\mathbf{t}|\mathbf{x}) \log \frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + \beta(\mathbf{x}) q(\mathbf{t}|\mathbf{x}) \Big\} \mathrm{d}\mathbf{x} \right\} \mathrm{d}\mathbf{t} \tag{13}$$

For this first part of the problem, the $\gamma(\mathbf{t})$ is fixed and attention is focused on $q(\mathbf{t}|\mathbf{x})$. However, an important point to keep in mind in optimizing the functional F with respect to $q(\mathbf{t}|\mathbf{x})$ is the fact that $q(\mathbf{t})$ is dependent on $q(\mathbf{t}|\mathbf{x})$ as shown by the formula (see Eqn. (10.185))

$$q(\mathbf{t}) = \int_{-\infty}^{\infty} p(\mathbf{x}) q(\mathbf{t}|\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{14}$$

To differentiate the integral inside the braces in (13) with respect to $q(\mathbf{t}|\mathbf{x})$, it is the second term that needs special attention as it depends on $q(\mathbf{t})$. To this

end, we write

$$\begin{aligned}
&\frac{\partial}{\partial q(\mathbf{t}|\mathbf{x})}\left[q(\mathbf{t}|\mathbf{x})\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\right] \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + q(\mathbf{t}|\mathbf{x})\frac{\partial}{\partial q(\mathbf{t}|\mathbf{x})}\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + q(\mathbf{t}|\mathbf{x})\frac{\partial}{\partial q(\mathbf{t}|\mathbf{x})}(\log q(\mathbf{t}|\mathbf{x}) - \log q(\mathbf{t})) \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + 1 - q(\mathbf{t}|\mathbf{x})\frac{\partial q(\mathbf{t})}{\partial q(\mathbf{t}|\mathbf{x})}\frac{\partial}{\partial q(\mathbf{t})}\log q(\mathbf{t}) \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + 1 - \frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\frac{\partial q(\mathbf{t})}{\partial q(\mathbf{t}|\mathbf{x})} \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + 1 - \frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\frac{\partial}{\partial q(\mathbf{t}|\mathbf{x})}\int_{-\infty}^{\infty} p(\mathbf{x})q(\mathbf{t}|\mathbf{x})\mathrm{d}\mathbf{x} \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + 1 - \frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\int_{-\infty}^{\infty} p(\mathbf{x})\mathrm{d}\mathbf{x} \\
&\quad= \log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + 1 - \frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}
\end{aligned} \tag{15}$$

where, for the last term, we used the pdf property

$$\int_{-\infty}^{\infty} p(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \tag{16}$$

Next, using (15) in (13) we may express the inner integral (after the differentiation with respect to $q(\mathbf{t}|\mathbf{x})$) as follows:

$$\int_{-\infty}^{\infty}\Big\{p(\mathbf{x})\|\mathbf{x}-\gamma(\mathbf{t})\|^2 + \lambda\Big(p(\mathbf{x})\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})} + p(\mathbf{x}) - \frac{1}{q(\mathbf{t})}p(\mathbf{x})q(\mathbf{t}|\mathbf{x})\Big) + \beta(\mathbf{x})\Big\}\mathrm{d}\mathbf{x} \tag{17}$$

In light of formulas in (14) and (16), we see that the integral over the last two terms in the middle line of (17) reduces to zero, as shown by

$$\begin{aligned}
\int_{-\infty}^{\infty}\Big\{p(\mathbf{x}) - \frac{1}{q(\mathbf{t})}p(\mathbf{x})q(\mathbf{t}|\mathbf{x})\Big\}\mathrm{d}\mathbf{x} &= \int_{-\infty}^{\infty} p(\mathbf{x})\mathrm{d}\mathbf{x} - \frac{1}{q(\mathbf{t})}\int_{-\infty}^{\infty} p(\mathbf{x})q(\mathbf{t}|\mathbf{x})\mathrm{d}\mathbf{x} \\
&= 1 - \frac{q(\mathbf{t})}{q(\mathbf{t})} = 0
\end{aligned}$$

Accordingly, we may simplify the integral in (17) as follows:

$$\int_{\infty}^{\infty}\Big\{p(\mathbf{x})\Big(\|\mathbf{x}-\gamma(\mathbf{t})\|^2 + \lambda\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\Big) + \beta(\mathbf{x})\Big\}\mathrm{d}\mathbf{x}. \tag{18}$$

We are now ready to state the necessary condition for optimality of the functional $F(\gamma(\mathbf{t}), q(\mathbf{t}|\mathbf{x}))$ with respect to $q(\mathbf{t}|\mathbf{x})$ given that $\gamma(\mathbf{t})$ is fixed. Specifically,

this condition is satisfied when the integrand in (18) is zero; that is,

$$p(\mathbf{x})\Big(\|\mathbf{x}-\gamma(\mathbf{t})\|^2+\lambda\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\Big)+\beta(\mathbf{x})=0$$

or, equivalently,

$$\frac{1}{\lambda}\|\mathbf{x}-\gamma(\mathbf{t})\|^2+\log\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}+\frac{1}{\lambda}\frac{\beta(\mathbf{x})}{p(\mathbf{x})}=0, \tag{19}$$

which proves the validity of the relation preceding Eqn. (10.187) in the textbook.

(b) Let

$$\frac{1}{\lambda}\frac{\beta(\mathbf{x})}{p(\mathbf{x})}=\log Z(\mathbf{x}).$$

Then we may rewrite (19) as follows:

$$\frac{1}{\lambda}\|\mathbf{x}-\gamma(\mathbf{t})\|^2+\log\left(Z(\mathbf{x})\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}\right)=0$$

Equivalently, we may rewrite

$$Z(\mathbf{x})\frac{q(\mathbf{t}|\mathbf{x})}{q(\mathbf{t})}=\exp\left(-\frac{1}{\lambda}\|\mathbf{x}-\gamma(\mathbf{t})\|^2\right) \tag{20}$$

We already know $q(\mathbf{t})$ in terms of $q(\mathbf{t}|\mathbf{x})$ by (14), we therefore may solve (20) for the unknown $q(\mathbf{t}|\mathbf{x})$ that optimizes the functional $F(\gamma(\mathbf{t}),q(\mathbf{t}|bx))$ for fixed $\gamma(\mathbf{t})$, obtaining

$$q(\mathbf{t}|\mathbf{x})=\frac{q(\mathbf{t})}{Z(\mathbf{x})}\exp\left(-\frac{1}{\gamma}\|\mathbf{x}-\gamma(\mathbf{t})\|^2\right) \tag{21}$$

For $q(\mathbf{t}|\mathbf{x})$ to qualify as a conditional pdf, for a given $\mathbf{x}$, its integral with respect to $\mathbf{t}$ must be unity. This property is satisfied by having $Z(\mathbf{x})$ assume the following value

$$Z(\mathbf{x})=\int_{-\infty}^{\infty}q(\mathbf{t})\exp\left(-\frac{1}{\lambda}\|\mathbf{x}-\gamma(\mathbf{x})\|^2\right)\mathrm{d}\mathbf{t} \tag{22}$$

Eqn. (21) and Eqn. (22) complete the solution. Note also that $Z(\mathbf{x})$ performs the role of a normalizing function.