Exercise 4.3
Part a:
It will cause deterministic noise to go up, because it will fit the training data too closely and capture more noise. From the book we know the increase of noise will cause a higher tendency to overfitting. Therefore in this case, it will cause a higher tendency of over fitting
Part b:
It will cause deterministic noise to go up, which causes an increased tendency to overfitting. On the other hand our model is simpler now, which can cause a decrease tendency to overfitting. However, since the model now is less complex, which means it is more likely to ignore the noise. Thus, I think the tendency to overfitting is going down.

Exercise 4.5
Part a:
Γ = I, I is a identify matrix with size Q+1 by Q+1:

$w^T\Gamma^T\Gamma w$ = $w^T I^T I w$ = $w^T w$ = $\sum_{q=0}^{Q} w^2$

Since $w^T\Gamma^T\Gamma w$ <= C, $\sum_{q=0}^{Q} w^2$ <=C

Part b:
Γ = matrix with 1 by Q+1, and each element in the matrix is equal to 1. Let's set the matrix called a(sorry can't create a matrix on google docs)

$w^T\Gamma^T\Gamma w$ =$w^T a^T w a$ = $\sum_{q=0}^{Q} w$ * $\sum_{q=0}^{Q} w$ = $(\sum_{q=0}^{Q} w)^2$

Exercise 4.6
For hard-order constraints, we can limit the length of the vector to avoid noise. However, for soft-order constraint, sign($w^T x$) = sign($\alpha w^T x$) if α > 0, which means soft-order constraint didn't do anything in classification. Thus, hard-order constraint is more useful.

Exercise 4.7
Part a:

$\sigma^2_{val} = VarD_{val}[E_{val}(g^-)]$

$E_{val}(g^-)$ = $\frac{1}{K}\sum_{x_n \in D_{val}}$ [e($g^-(xn)$, yn]

σ^2_{val} = $VarD_{val}$[$\frac{1}{K}\sum_{x_n \in D_{val}}$ [e($g^-(xn)$, yn]

σ^2_{val} = $\frac{1}{K^2}\sum_{x_n \in D_{val}} Var_{xn}$[e($g^-(xn)$, yn]

σ^2_{val} = $\frac{1}{K^2}$ * K * Var_x[e($g^-(x)$, y]

σ^2_{val} = $\frac{1}{K}$ *σ^2 (g^-)

Part b:

e($g^-(x)$,y) = [$g^-(x)$!= y]

E(e($g^-(x)$,y)) = P[$g^-(x)$!= y]

σ^2_{val} = $\frac{1}{K}Var_x$[e($g^-(x)$, y]

σ^2_{val} = $\frac{1}{K}$(E[(e($g^-(x)$, y)^2] - E[(e($g^-(x)$, y)]^2)

E(e($g^-(x)$,y)) = P[$g^-(x)$!= y], and P[$g^-(x)$!= y] can only be 0 or 1, 0^2 = 0, 1^2=1, so (e($g^-(x)$, y)^2 = (e($g^-(x)$, y)

σ^2_{val} = $\frac{1}{K}$(P[$g^-(x)$!= y] - P[$g^-(x)$!= y]^2)

Part c:

From part c:

σ^2_{val} = $\frac{1}{K}$(P[$g^-(x)$!= y] - P[$g^-(x)$!= y]^2)

Set P = P[$g^-(x)$!= y]

σ^2_{val} = $\frac{1}{K}$(P - P^2)

σ^2_{val} = $\frac{1}{K}$(- $(P-\frac{1}{2})^2 + \frac{1}{4}$)

σ^2_{val} = $\frac{1}{4K}$ - $\frac{1}{K}(P-\frac{1}{2})^2$

Since $\frac{1}{K}(P-\frac{1}{2})^2$ always >= 0

$\sigma^2_{val} \leq \frac{1}{4K}$

Part d:

Based on variation formula Var_x[e($g^-(x)$, y] = E[(e($g^-(x)$, y)^2] - E[(e($g^-(x)$, y)]^2, we know squared error e($g^-(x)$, y is unbonded, so Var_x[e($g^-(x)$, y] is unbonded.

Part e:

Higher, Because training with less data points will be harder to predict the target function, so σ^2 will increase, and $\sigma^2 g^-$ will be higher.

Part f:

From part a we know σ^2_{val} = $\frac{1}{K}$ *σ^2 (g^-).

If we increase the size of validation of set means the size of training set will became smaller, from part e we know that decrease the training set size will cause $\sigma^2\ (g^-)$ become larger, but on the other hand $\frac{1}{K}$ become smaller. Also when K is small add a bit to K can make $\frac{1}{K}$ become much smaller than K is big. Thus, I think if we add data to a small size of validation set the Eout will become better, but if we add data to a large size of validation set the Eout will become worse.

Exercise 4.8
Yes, Em is an unbiased estimate for Eout, because validation set is not involving in any decision that affect any process.