Kevin R Murphy,
"Machine Learning - A Probabilistic Perspective",
The MIT Press, 2012.

Secs:

- 7.5 (Ridge Regression vs PCA);

- 9.5 (multi-task learning) in brief;

- 10.1; 10.3 – Bayes Net (DGM)

Ridge Regression vs. PCA

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \mathbf{w}^T\mathbf{x}_i))^2 + \lambda||\mathbf{w}||_2^2 \tag{7.32}$$

where $\lambda \triangleq \sigma^2/\tau^2$ and $||\mathbf{w}||_2^2 = \sum_j w_j^2 = \mathbf{w}^T\mathbf{w}$ is the squared two-norm. Here the first term is the MSE/ NLL as usual, and the second term, $\lambda \geq 0$, is a complexity penalty. The corresponding solution is given by

$$\hat{\mathbf{w}}_{ridge} = (\lambda\mathbf{I}_D + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{7.33}$$

This technique is known as **ridge regression**, or **penalized least squares**. In general, adding a Gaussian prior to the parameters of a model to encourage them to be small is called ℓ_2 **regularization** or **weight decay**. Note that the offset term w_0 is not regularized, since this just

Now let

$$\tilde{\mathbf{X}} = \mathbf{Q}\mathbf{R} \tag{7.41}$$

be the QR decomposition of $\mathbf{X}$, where $\mathbf{Q}$ is orthonormal (meaning $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$), and $\mathbf{R}$ is upper triangular. Then

$$(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1} = (\mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R})^{-1} = (\mathbf{R}^T\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-T} \tag{7.42}$$

Hence

$$\hat{\mathbf{w}}_{ridge} = \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{R}^T\mathbf{Q}^T\tilde{\mathbf{y}} = \mathbf{R}^{-1}\mathbf{Q}\tilde{\mathbf{y}} \tag{7.43}$$

$$\hat{\mathbf{w}}_{ridge} = (\lambda \mathbf{I}_D + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

If $D \gg N$, we should first perform an SVD decomposition. In particular, let $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ be the SVD of $\mathbf{X}$, where $\mathbf{V}^T\mathbf{V} = \mathbf{I}_N$, $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_N$, and $\mathbf{S}$ is a diagonal $N \times N$ matrix. Now let $\mathbf{Z} = \mathbf{U}\mathbf{D}$ be an $N \times N$ matrix. Then we can rewrite the ridge estimate thus:

$$\hat{\mathbf{w}}_{ridge} = \mathbf{V}(\mathbf{Z}^T\mathbf{Z} + \lambda \mathbf{I}_N)^{-1}\mathbf{Z}^T\mathbf{y} \quad (7.44)$$

$$\widehat{w}_{ridge} = [\lambda \boldsymbol{I}_D + \boldsymbol{X}^T\boldsymbol{X}]^{-1}\boldsymbol{X}^T\boldsymbol{y} \quad \ldots\ldots\ldots\ldots\ldots\ldots (7.33)$$

Substituting $\boldsymbol{X} = \boldsymbol{USV}^T$, the SVD decomposition

where $\boldsymbol{U}: N \times N, \quad \boldsymbol{S}: N \times N, \quad \boldsymbol{V}: D \times N$ and

$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_N, \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_N, \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_D$

$$\widehat{w}_{ridge} = [\lambda \boldsymbol{I}_D + (\boldsymbol{USV}^T)^T(\boldsymbol{USV}^T)]^{-1}(\boldsymbol{USV}^T)^T\boldsymbol{y}$$

$$= [\lambda \boldsymbol{I}_D + \boldsymbol{VSU}^T\boldsymbol{USV}^T]^{-1}\boldsymbol{VSU}^T\boldsymbol{y}$$

$$= [\lambda \boldsymbol{V}\boldsymbol{I}_N\boldsymbol{V}^T + \boldsymbol{VS}^2\boldsymbol{V}^T]^{-1}\boldsymbol{VSU}^T\boldsymbol{y}$$

$$= [\boldsymbol{V}(\lambda \boldsymbol{I}_N + \boldsymbol{S}^2)\boldsymbol{V}^T]^{-1}\boldsymbol{VSU}^T\boldsymbol{y}$$

$$= \boldsymbol{V}(\lambda \boldsymbol{I}_N + \boldsymbol{S}^2)^{-1}\boldsymbol{V}^T\boldsymbol{VSU}^T\boldsymbol{y}$$

$$= \boldsymbol{V}(\lambda \boldsymbol{I}_N + \boldsymbol{S}^2)^{-1}\boldsymbol{SU}^T\boldsymbol{y}$$

Substituting $\boldsymbol{Z} = \boldsymbol{US}, \quad \boldsymbol{Z}^T = \boldsymbol{SU}^T, \quad \boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{SU}^T\boldsymbol{US} = \boldsymbol{S}^2$

Therefore,

$$\widehat{w}_{ridge} = \boldsymbol{V}(\lambda \boldsymbol{I}_N + \boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{y} \ldots\ldots\ldots\ldots\ldots\ldots (7.44)$$

Ridge Regression vs. PCA

- We discuss an interesting connection between ridge regression and PCA, which gives further insight into why ridge regression works well. From (7.44), as below:

$$\hat{\mathbf{w}}_{ridge} = \mathbf{V}(\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I}_N)^{-1}\mathbf{Z}^T\mathbf{y}$$

- Let $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$ be the SVD of $\mathbf{X}$ (data). From Equation 7.44 (Murphy), we have

$$\hat{\mathbf{w}}_{ridge} = \mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{S}\mathbf{U}^{\mathrm{T}}\,\mathbf{y} \qquad (7.45)$$

- Hence the ridge predictions on the training set are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}_{ridge} = \mathbf{U}[\mathbf{S}\mathbf{V}^{\mathrm{T}}\mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{S}]\mathbf{U}^{\mathrm{T}}\,\mathbf{y} \qquad (7.46)$$

$$= \mathbf{U}\tilde{\mathbf{S}}\mathbf{U}^{\mathrm{T}}\mathbf{y} = \sum_{j=1}^{D}\mathbf{u}_{\mathrm{j}}\tilde{S}_{jj}\mathbf{u}_{\mathrm{j}}^{\mathrm{T}}\mathbf{y} \qquad (7.47)$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}_{ridge} = \mathbf{USV}^{\mathrm{T}}\mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{SU}^{\mathrm{T}}\,\mathbf{y} \qquad (7.46)$$

$$= \mathbf{U}\tilde{\mathbf{S}}\mathbf{U}^{\mathrm{T}}\,\mathbf{y} = \sum_{j=1}^{D} \mathbf{u}_{\mathrm{j}}\,\tilde{S}_{jj}\mathbf{u}_{\mathrm{j}}^{\mathrm{T}}\mathbf{y} \qquad (7.47)$$

where

$$\tilde{S}_{jj} \triangleq [S(S^2 + \lambda I)^{-1}S]_{jj} = \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \qquad (7.48)$$

- And σ_j are the singular values of $\mathbf{X}$. Hence

$$\hat{\mathrm{y}} = \mathbf{X}\hat{\mathbf{w}}_{ridge} = \sum_{j=1}^{D} \mathbf{u}_{\mathrm{j}}\,\frac{\sigma_j^2}{\sigma_j^2 + \lambda}\,\mathbf{u}_{\mathrm{j}}^{\mathrm{T}}\mathbf{y} \qquad (7.49)$$

- In contrast, the least squares prediction is

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}_{ls} = \left(\mathbf{USV}^{\mathrm{T}}\right)\left(\mathbf{V}\mathrm{S}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{y}\right) = \mathbf{UU}^{\mathrm{T}}\mathbf{y} = \sum_{j=1}^{D} \mathbf{u}_{\mathrm{j}}\mathbf{u}_{\mathrm{j}}^{\mathrm{T}}\,\mathbf{y} \qquad (7.50)$$

- If σ_j^2 is small compared to λ, then direction $\mathbf{u}_j$ will not have much effect on the prediction. In view of this, we define the effective number of ***degrees of freedom*** of the model as follows:

$$\mathrm{dof}(\lambda) = \sum_{j=1}^{D} \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \qquad (7.51)$$

- When $\lambda = 0, \mathrm{dof}(\lambda) = D$, and as $\lambda \to \infty, \mathrm{dof}(\lambda) \to 0$.

- Let us try to understand why this behavior is desirable. It can be shown that $cov\,[\mathbf{w}|\mathcal{D}] = \sigma^2(\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}$ (Sec. 7.6, Murphy), if we use a *uniform prior for* ***w.***

Sec. 7.6: Posterior, of Bayesian linear regression

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) &\propto \mathcal{N}(\mathbf{w}|\mathbf{w}_0, \mathbf{V}_0)\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}_N) = \mathcal{N}(\mathbf{w}|\mathbf{w}_N, \mathbf{V}_N) \\
\mathbf{w}_N &= \mathbf{V}_N\mathbf{V}_0^{-1}\mathbf{w}_0 + \frac{1}{\sigma^2}\mathbf{V}_N\mathbf{X}^T\mathbf{y} \\
\mathbf{V}_N^{-1} &= \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} \\
\mathbf{V}_N &= \sigma^2(\sigma^2\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X})^{-1}
\end{aligned}
$$

$$cov\,[\mathbf{w}|\mathcal{D}] = \sigma^2\left(\mathbf{X^T X}\right)^{-1}$$

- Thus the directions in which we are most uncertain about $\mathbf{w}$ are determined by the eigenvectors of this matrix with the smallest eigenvalues.
- Furthermore, it is known that the squared singular values σ_j^2 are equal to the eigenvalues of $X^T X$. Hence small singular values σ_j correspond to directions with high posterior variance.
- It is these directions which ridge shrinks the most.

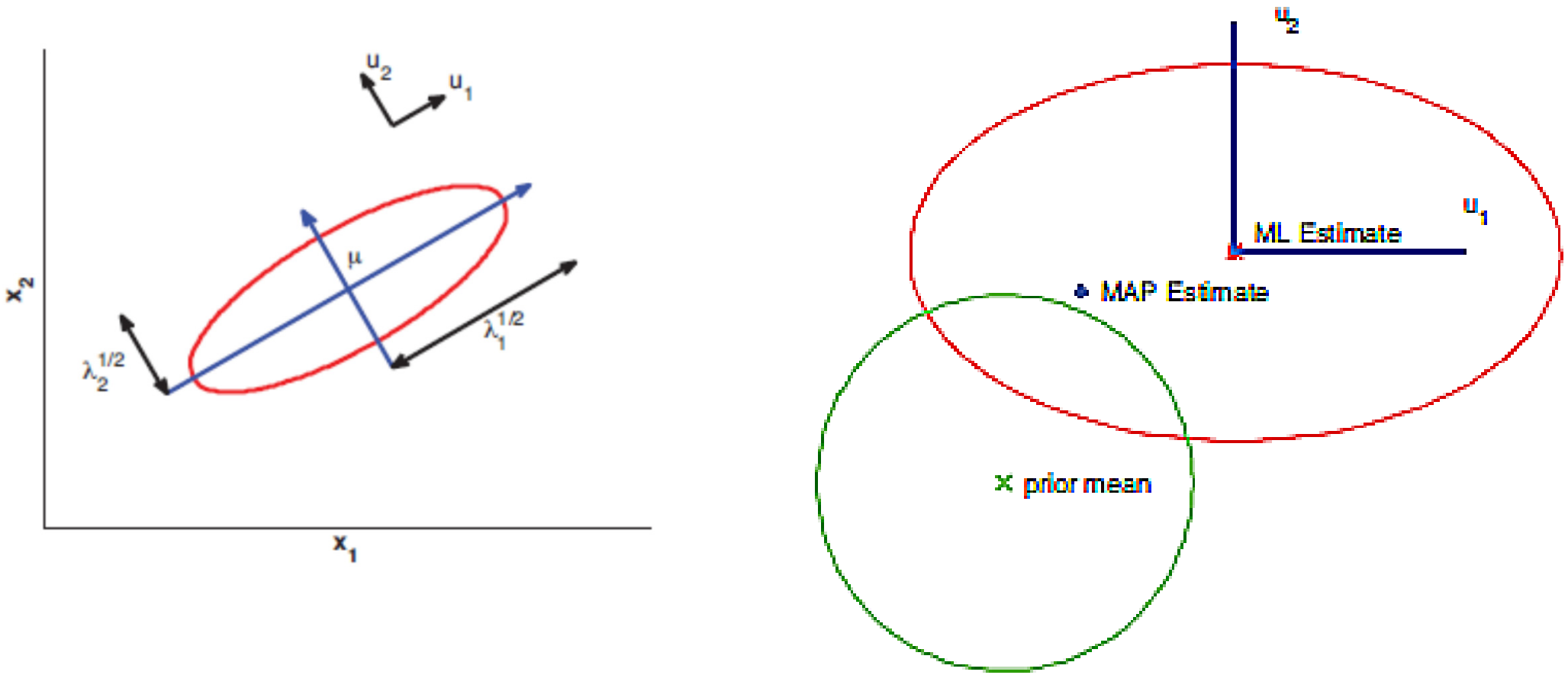

Figure 7.9 Geometry of ridge regression. The likelihood is shown as an ellipse, and the prior is shown as a circle centered on the origin.

- This process is illustrated in Figure 7.9. The horizontal w_1 parameter is not-well determined by the data (has high posterior variance), but the vertical w_2 parameter is well-determined. Hence w_2^{map} is close to $\hat{w}_2^{mle}$, but w_1^{map} is shifted strongly towards the prior mean, which is 0.

- In this way, ill-determined parameters are reduced in size towards 0. This is called shrinkage.

- There is a related, but different, technique called **principal components regression**.

- The idea is this: first use PCA to reduce the dimensionality to K dimensions, and then use these low dimensional features as input to regression.
- However, this technique does not work as well as ridge in terms of predictive accuracy (Hastie et al. $2001, p70$).
- The reason is that in PC regression, only the first K (derived) dimensions are retained, and the remaining $D - K$ dimensions are entirely ignored. By contrast, ridge regression uses a "soft" weighting of all the dimensions.

Multi-task learning

- Sometimes we want to fit many related classification or regression models. It is often reasonable to assume the input-output mapping is similar across these different models, so we can get better performance by fitting all the parameters at the same time.

- In machine learning, this setup is often called ***multi-task learning*** (Caruana 1998), ***transfer learning*** (e.g., (Raina et al.2005)), or ***learning to learn*** (Thrun and Pratt 1997).

- In statistics, this is usually tackled using hierarchical Bayesian models (Bakker and Heskes 2003), although there are other possible methods (see e.g., (Chai 2010)).

Hierarchical Bayes for multi-task learning

- Let y_{ij} be the response of the i^{th} item in group j, for $i = 1 : N_j$ and $j = 1 : J$. For example, j might index schools, i might index students within a school, and y_{ij} might be the test score.
- Or j might index people, and i might index purchase, and y_{ij}might be the identity of the item that was purchased.
- Let x_{ij}be a feature vector associated with y_{ij}. The goal is to fit the models $p(y_j|x_j)$ for all j.
- Although some groups may have lots of data, there is often a long tail, where the majority of groups have little data.

- Thus we can't reliably fit each model separately, but we don't want to use the same model for all groups.
- As a compromise, we can fit a separate model for each group, but encourage the model parameters to be similar across groups.
- More precisely, suppose $\mathbb{E}[y_{ij}|\mathrm{x}_{ij}] = g(\mathrm{x}_{ij}^T\beta_j)$, where g is the link function for the Generalized Linear Model.
- Furthermore, suppose $\beta_j \sim \mathcal{N}(\beta_*, \sigma_j^2\boldsymbol{I})$, and that $\beta_* \sim \mathcal{N}(\mu, \sigma_*^2\boldsymbol{I})$.
- In this model, groups with small sample size borrow statistical strength from the groups with larger sample size, because the β_j 's are correlated via the latent common parents β_*.

- The term σ_j^2 controls how much group j depends on the common parents and the σ_*^2 term controls the strength of the overall prior.
- Suppose, for simplicity, that $\boldsymbol{\mu} = \mathbf{0}$, and that σ_j^2 and σ_*^2 are all known (e.g., they could be set by cross validation). The overall log probability has the form:

$$\log p(\mathcal{D}|\beta) + \log p(\beta) = \sum_j [\log p\left(\mathcal{D}_j \middle| \beta_j\right) - \frac{\left|\left|\beta_j - \beta_*\right|\right|^2}{2\sigma_j^2}] - \frac{||\beta_*||^2}{2\sigma_j^2} \quad (9.110)$$

- We can perform MAP estimation of $\beta = (\beta_{1:J}, \beta_*)$ using standard gradient methods.
- Alternatively, we can perform an iterative optimization scheme, alternating between optimizing the β_j and the β_*; since the likelihood and prior are convex, this is guaranteed to converge to the global optimum.
- Note that once the models are trained, we can discard β_*, and use each model separately.