Lecture 13, Feb 26, 2024

Linear Regression

- Consider a linear model $Y = \boldsymbol{w}^T\boldsymbol{X} + Z$ where we have n noisy measurements y_i from n inputs $\boldsymbol{x}_i$
  - Assume Z is some IID Gaussian random noise
  - Given these measurements, our goal is to find the best set of weights $\boldsymbol{w}^T = \begin{bmatrix} w_1 & \dots & w_D \end{bmatrix}$
  - Each weight w_j corresponds to the jth coefficient of $\boldsymbol{x}$, which has dimension D
- Form the *design matrix* $\begin{bmatrix} \boldsymbol{x}_1^T & y_1 \\ \vdots & \vdots \\ \boldsymbol{x}_n^T & y_n \end{bmatrix}$
- Consider the MLE

$$\begin{aligned}
\hat{\boldsymbol{w}}_{\text{ML}} &= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} \log p((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)|\boldsymbol{w}) \\
&= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} \log \prod_{i=1}^{n} p(\boldsymbol{x}_i, y_i|\boldsymbol{w}) \\
&= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2}\right) \\
&= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} -\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2 \\
&= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}}\, \boldsymbol{e}^T(\boldsymbol{w})\boldsymbol{e}(\boldsymbol{w})
\end{aligned}$$

  - Where the error vector is $\boldsymbol{e}(\boldsymbol{w}) = \begin{bmatrix} y_1 - \boldsymbol{w}^T\boldsymbol{x}_1 \\ \dots \\ y_n - \boldsymbol{w}^T\boldsymbol{x}_n \end{bmatrix} = \boldsymbol{y} - \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}\boldsymbol{w}$
  - This is now a *least squares regression problem*
  - Let $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$ then we have $\underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$
  - Expand: $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{X}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}\boldsymbol{X}^T\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}$ (note a factor of $\frac{1}{2}$ was added)
  - Derivative: $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y} = 0 \implies \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$
  - Therefore $\hat{\boldsymbol{w}}_{\text{ML}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- Another way to write this is $\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = 0$, meaning we can interpret this as making the error vector orthogonal to all the input data
  - This is the *normal equation*
- $\boldsymbol{X}^T\boldsymbol{X}$ is the *scatter matrix*
  - This is an estimate of the covariance/correlation matrix of the data
- Regression can be performed in any general vector space, so our model can be nonlinear in $\boldsymbol{x}$ (but still linear in $\boldsymbol{w}$)
  - In general given any basis function $\boldsymbol{\phi}(\boldsymbol{x}_i)$ we can try to fit $y_i = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) + z_i$
  - Let $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \end{bmatrix}$ then $\hat{\boldsymbol{w}}_{\text{ML}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  - e.g. we can work in the vector space of polynomials to perform polynomial regression, or the space of sinusoids for a Fourier series
    - For d-degree polynomial regression we'd have $\boldsymbol{\phi}^T(x_i) = \begin{bmatrix} 1 & x_i & x_i^2 & \dots & x_i^d \end{bmatrix}$
  - Example: measuring the height of a cannonball h_i vs. time t_i for $i = 1, \dots, n$
    - Use the model $h_i = w_1 t_i + w_2 t_i^2 + z_i = \boldsymbol{w}^T\boldsymbol{x}_i + z_i$ where $\boldsymbol{x}_i = \begin{bmatrix} t_i \\ t_i^2 \end{bmatrix}$

Bayesian Regression – Regularization

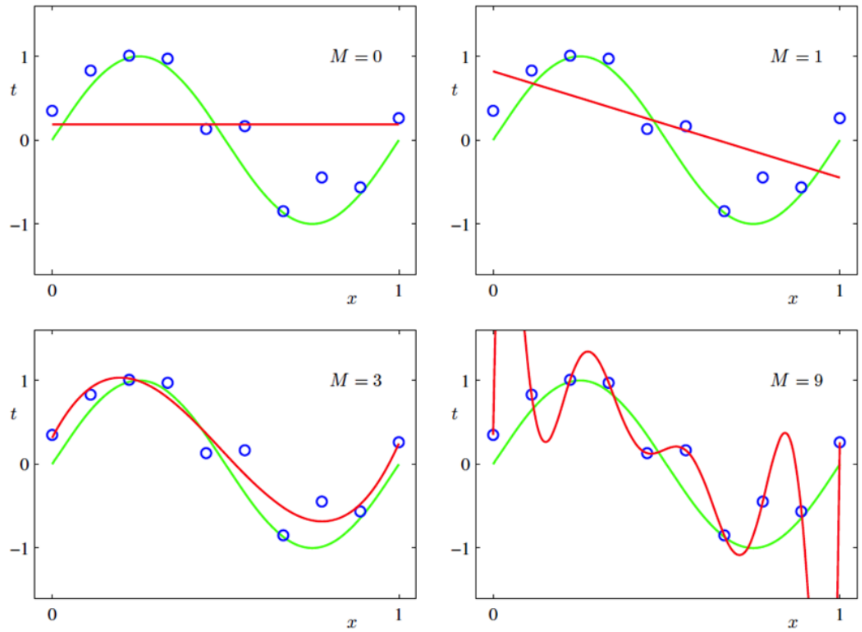

Figure 1: Polynomial regression for different degrees. Green is the underlying function we're trying to approximate.

- If we make the model too complex, i.e. too high of a dimension for $\boldsymbol{\phi}$, we will get overfitting
- Typically when the model overfits, we get very large weights that are not physically realistic for our system
  - To keep the weights down, we can use regularization
  - Here we show a way to derive the same result by instead assuming a prior on $\boldsymbol{w}$
- Assume that each weight has a prior $w_i \sim \mathcal{N}(0, \tau^2)$; now can find the MAP estimate
- $\hat{\boldsymbol{w}}_{\text{MAP}} = \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}}\, p((\boldsymbol{\phi}(x_1), y_1), \ldots, (\boldsymbol{\phi}(x_n), y_n))p(\boldsymbol{w})$

$$= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2}\right) + \sum_{j=1}^{D} \log \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{w_j^2}{2\tau^2}} \right]$$

$$= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2 + \frac{\sigma^2}{\tau^2}\|\boldsymbol{w}\|^2$$

$$= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2 + \lambda\|\boldsymbol{w}\|^2$$

  - The first term is the same least squares term as before, but now we have an additional term that penalizes the norm of $\boldsymbol{w}$, effectively keeping the weights small
  - Let $\boldsymbol{e}(\boldsymbol{w}) = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \\ -\sqrt{\lambda}\boldsymbol{1} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix}$ and $\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \\ -\sqrt{\lambda}\boldsymbol{1} \end{bmatrix}$
    - The error can again be written as $\boldsymbol{e}^T(\boldsymbol{w})\boldsymbol{e}(\boldsymbol{w})$
  - Using the same derivation as before, $\hat{\boldsymbol{w}}_{\text{MAP}} = (\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^T \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{bmatrix} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{1})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  - Notice that this result is almost the same as the MLE solution, except with the addition of $\lambda\boldsymbol{1}$
  - This is known as *ridge regression*

- We can also solve this by writing it as a Gaussian system
  - $\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} \begin{bmatrix} W_1 \\ \vdots \\ W_D \end{bmatrix} + \begin{bmatrix} Z_1 \\ Z_n \end{bmatrix} \iff \boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{z}$
  - $\hat{\boldsymbol{w}}_{\text{MAP}}(\boldsymbol{y}) = (\boldsymbol{\Sigma}_W^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_Z^{-1}\boldsymbol{X})^{-1}\left(\boldsymbol{X}^T\frac{1}{\sigma^2}\boldsymbol{y} + \frac{1}{\tau^2}\cdot 0\right) = (\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  - This gives us the conditional precision $\boldsymbol{\Sigma}_{X|Y}^{-1} = \frac{1}{\tau^2}\mathbf{1} + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}$
  - So $\boldsymbol{\Sigma}_{X|Y} = \sigma^2\left(\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1}\right)^{-1}$
    * Notice that $\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1}$ is related to the covariance
    * As we collect more data the added term becomes negligible
    * $\boldsymbol{X}^T\boldsymbol{X}$ becomes bigger so the covariance shrinks