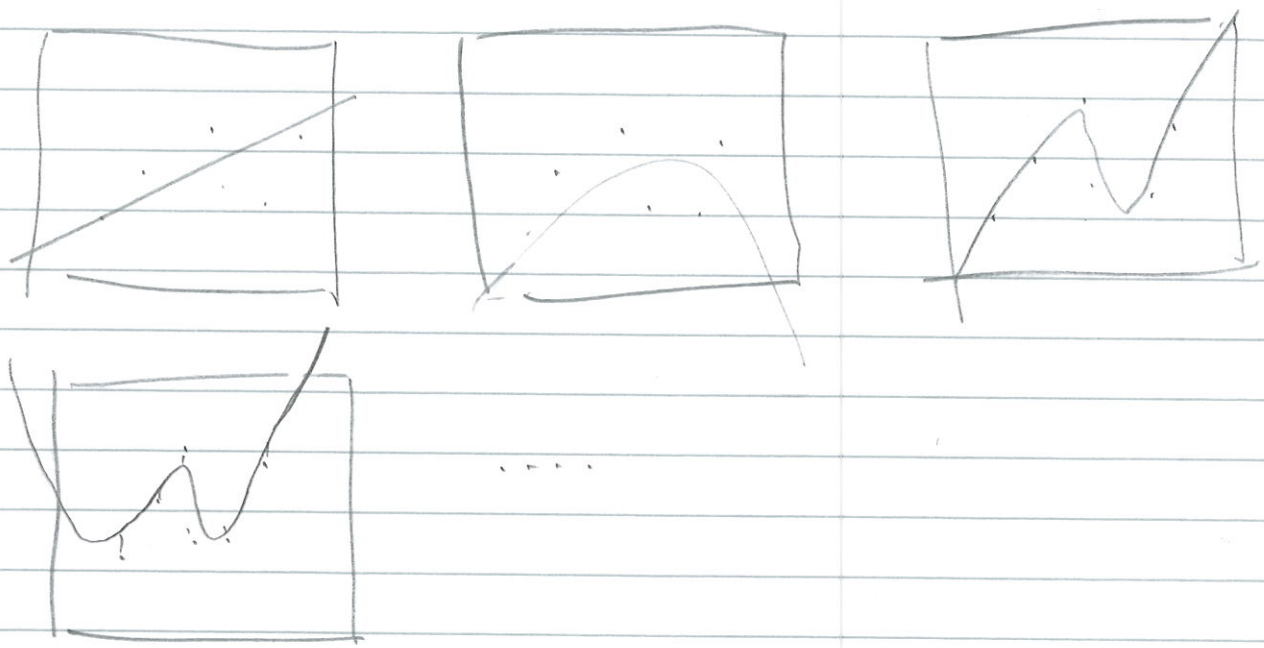
Regularization & Bayesian Linear Regression (1)

As model complexity increases the ability for a model to fit the data becomes higher.

Polynomial regression serves as an excellent pedogogical tool for explaining what happens when a model begins to overfit:



As the polynomial order increases the ability of the model to exactly reproduce the target values increases; however, out-of-sample performance typically degrades.

To combat this we typically regularize or bias models towards reasonable solutions. In this case we might like the polynomial weights to be small.

One way to do this is to directly penalize large weights, ie. modifying our ML objective

$$E_D(\vec{w}) = \frac{1}{2} \sum_{n=1}^{N} \{ t_n - \vec{w}^T \phi(x_n) \}^2$$

$$= \frac{1}{2} \left(\vec{t} - \underline{\Phi} \vec{w} \right)^T \left(\vec{t} - \underline{\Phi} \vec{w} \right)$$

where E is "energy", ie. log. prob. and

$$\underline{\Phi} = \begin{matrix} \\ n \end{matrix} \overset{k}{\begin{bmatrix} \phi_1(x_1) & \cdots & \phi_k(x_1) \\ \vdots & & \vdots \\ \phi_1(x_n) & \cdots & \phi_k(x_n) \end{bmatrix}}$$

to include a term like

$$E_w(\vec{w}) = \frac{\lambda}{2} \vec{w}^T \vec{w}$$

where λ is a smoothing tuning parameter. Maximum probability is minimum energy so we might wish to find

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} \; E_D(\vec{w}) + E_w(\vec{w})$$

$$= \underset{\vec{w}}{\operatorname{argmin}} \; \frac{1}{2} \left(\vec{t} - \underline{\Phi} \vec{w} \right)^T \left(\vec{t} - \underline{\Phi} \vec{w} \right) + \frac{\lambda}{2} \vec{w}^T \vec{w}$$

which arises at $\frac{\partial}{\partial \vec{w}} \left(E_D(\vec{w}) + E_w(\vec{w}) \right) = 0$

$$0 = \frac{\partial}{\partial \vec{w}} \left(\frac{1}{2} \left(\vec{t} - \underline{\Phi} \vec{w} \right)^T \left(\vec{t} - \underline{\Phi} \vec{w} \right) + \frac{\lambda}{2} \vec{w}^T \vec{w} \right)$$

$$- \underline{\Phi}^T \left(\vec{t} - \underline{\Phi} \vec{w} \right) + \lambda \vec{w}$$

$$= - \underline{\Phi}^T \vec{t} + \left(\underline{\Phi}^T \underline{\Phi} + \lambda \underline{I} \right) \vec{w}$$

$$\Rightarrow \vec{w} = \left(\underline{\Phi}^T \underline{\Phi} + \lambda \underline{I} \right)^{-1} \underline{\Phi}^T \vec{t}$$

What does this mean and do?

1) Inverting $\Phi^T\Phi$ becomes unstable or impossible if Φ is rank deficient. Adding positive elements to the diagonal ensures that $(\Phi^T\Phi + \lambda I)$ is full rank and therefore invertible.

2) $\frac{\lambda}{2}\vec{w}^T\vec{w}$ looks a lot like a mumormal, ie. $\frac{1}{2}(\vec{w}-\vec{0})^T \lambda I (\vec{w}-\vec{0})$ which in energy terms will be minimized when $\vec{w}$ is close to $\vec{0}$.

3) This is a form of BIAS that is helpful!

To interpret this further we appeal to Bayesian reasoning, e.g. Bayesian linear regression.

Let's consider Bayes rule in the context of linear regression.

$$p(\vec{w} \mid \vec{t}, \Phi, \beta, \lambda) = \frac{p(\vec{t} \mid \Phi, \vec{w}, \beta)\, p(\vec{w} \mid \lambda)}{p(\vec{t} \mid \Phi, \beta, \lambda)}$$

this is a little confusing so let's drop all the constants

$$p(\vec{w} \mid \vec{t}) = \frac{p(\vec{t} \mid \vec{w})\, p(\vec{w})}{p(\vec{t})}$$

$$\propto p(\vec{t} \mid \vec{w})\, p(\vec{w})$$

$$= \mathcal{N}(\vec{t};\ \Phi\vec{w}, \beta I)\, \mathcal{N}(\vec{w};\ \vec{0}, \lambda I)$$

This is quite interesting. It says that a Bayesian approach to linear regression introduces bias. and, simultaneously, that solving for the MAP (maximum a posteriori) $\vec{w}$ is clearly equivalent to solving the regularized least squares problem.

What is more, we can, here, analytically derive the full posterior distribution. If we know (and understand) the following about the MVN

μ, b, x, y vectors
A, L, Λ, Σ matrices

$$\Rightarrow \begin{cases} p(x) = \mathcal{N}(x \mid \mu, \Lambda^{-1}) \\ p(y|x) = \mathcal{N}(y \mid Ax+b, L^{-1}) \end{cases}$$

$$\begin{cases} p(y) = \mathcal{N}(y \mid A\mu+b, L^{-1} + A\Lambda^{-1}A^T) \\ p(x|y) = \mathcal{N}(x \mid \Sigma\{A^T L(y-b) + \Lambda\mu\}, \Sigma) \end{cases}$$

where $\Sigma = (\Lambda + A^T L A)^{-1}$

Bishop 2.113 → 2.117

This is effectively Bayes rule for Gaussians and with it we can immediately derive

$$p(\vec{w} \mid \cdots) = \mathcal{N}\left(\vec{w} \,\middle|\, \Sigma\left\{\Phi^T \tfrac{1}{\beta} I\left(\vec{t}\right)\right\}, \Sigma\right)$$
$$= \mathcal{N}\left(\vec{w} \,\middle|\, \tfrac{1}{\beta}\Sigma\Phi^T\vec{t}, \Sigma\right)$$

$k \times k \quad k \times n \quad n \times 1$
$k \times 1$

and

$$\Sigma = \left(\tfrac{1}{\lambda} I + \tfrac{1}{\beta}\Phi^T\Phi\right)^{-1}$$

which is the posterior distribution of the weight vector. We already know it's mode and mean: having the full distribution allows ease of model combination and propagation of uncertainty throughout inference and computation.

Leaving that for more advanced treatments, we usually, in practice, are interested in making predictions about an output t for new data x.

This can be written as

$$p(t \mid x, \vec{w}, \vec{\Phi}, \lambda, \beta) = \int p(t \mid \vec{w}, \beta)\, p(\vec{w} \mid \vec{t}, \Phi, \lambda, \beta)\, d\vec{w}$$

$p(t)$ — $p(t|x)$ — $p(x)$

$p(t|x)$: model of noise; $p(x)$: posterior of $\vec{w}$

$$\mathcal{N}(t; \vec{x}^T\vec{w}, \beta)\ \mathcal{N}\left(\vec{w}; \tfrac{1}{\beta}\Sigma\Phi^T\vec{t}, \Sigma\right)$$

$A x + b$, L^{-1}

$$\Sigma = \left(\tfrac{1}{\lambda}I + \tfrac{1}{\beta}\Phi\Phi^T\right)^{-1}$$

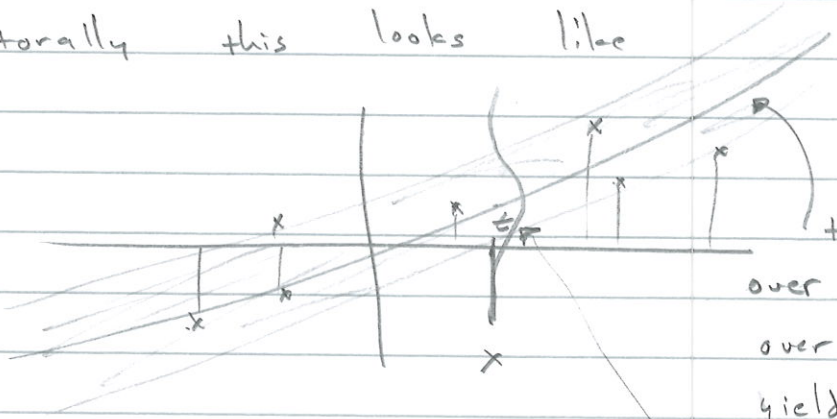
Λ^{-1}

$$\Rightarrow$$

$p(t \mid \cdots)$ Exercise for the reader ...

$$= \mathcal{N}\left(t; \vec{x}^T\Sigma\Phi^T\vec{t},\ \tfrac{1}{\beta} + \vec{x}^T\left(\tfrac{1}{\lambda}I + \tfrac{1}{\beta}\Phi^T\Phi\right)^{-1}\vec{x}\right)$$

pictorally this looks like



the integration over $\vec{w}$ averages over possible "lines" yielding a dist. over predictions

As we can see here our predictive dist. is Gaussian in the Bayesian linear regression setting. What does this mean though? By propigating uncertainty about the value of the weights throughout the computation we get error bars on our predictions. Analyses using these are constrained by the model family but otherwise are safer, in general, than using point estimates.