Lecture 17

Goodness-of-fit measures

(a) Root-mean-square error:

$$RMSE(\vec{w}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \vec{w}'\cdot\vec{x}_n)^2}$$

($\vec{w}$: D+1; $\vec{x}_n$: D+1)

(b) Coeffs. of determination:

$$R^2 = 1 - \frac{\sum_n (y_n - \hat{y}_n)^2}{\sum_n (y_n - \bar{y})^2}$$

← var in our model

← var in 'null' model $\bar{y}$

$$\hat{y}_n = \vec{w}'\cdot\vec{x}_n \,, \quad \bar{y} = \frac{1}{N}\sum_n y_n .$$

$0 \le R^2 \le 1$

↑ model does just as well as the $\bar{y}$ model

↑ model reproduces y_n exactly

larger R^2 ⇒ better fit

Ridge regression

Introduce prior $p(\vec{w}) = \mathcal{N}(\vec{w}|\vec{0}, \tau^2 \mathbb{I})$ to avoid overfitting.

($\vec{w}$: D-dim)

$$\vec{w}'_{MAP} = \underset{\vec{w}}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma^2}(\vec{y} - X\vec{w}')^T\cdot(\vec{y} - X\vec{w}') + \frac{1}{2\tau^2}\underbrace{\vec{w}^T\cdot\vec{w}}_{=|\vec{w}|^2} \right\}$$

← D-dim

b is found as before and is not constrained

Consider $J(\vec{w}) = (\vec{y} - X\vec{w})^T \cdot (\vec{y} - X\vec{w}) + \lambda |\vec{w}|^2$.
($J(\vec{w})$: D-dim; $\lambda = \frac{\sigma^2}{\tau^2}$)

Then $\underbrace{\frac{\partial}{\partial \vec{w}} J(\vec{w})}_{\text{D-dim}} = 2\left(X^T X \vec{w} - X^T \vec{y} + \lambda \vec{w}\right) = 0$, so that

[$X_{N\times D}$ is the centered data matrix, $\vec{y}$ is a vector of centered observations]

$$(X^T X + \lambda \mathbb{I}_D)\, \vec{w}_{MAP} = X^T \vec{y},$$

$$\vec{w}_{MAP} = (X^T X + \lambda \mathbb{I}_D)^{-1} X^T \vec{y} =$$

$$= \Big(\underbrace{\sum_n (\vec{x}_n \cdot \vec{x}_n^T + \lambda \mathbb{I}_D)}_{D\times D} \Big)^{-1} \times \sum_n y_n \vec{x}_n .$$

$X^T X + \lambda \mathbb{I}_D$ should be more stable to invert than $X^T X$.

SVD solution

Consider $X_{N\times D} = \mathcal{U}_{N\times N} S_{N\times D} V^T_{D\times D}$, then

$$\vec{w}_{MAP} = ((V S^T \mathcal{U}^T)(\mathcal{U} S V^T) + \lambda \mathbb{I}_D)^{-1} (V S^T \mathcal{U}^T)\, \vec{y} \quad (=)$$

Note that in general,

$$(+) \quad \underbrace{(\mathbb{I}_M + A_{M\times N} B_{N\times M})^{-1}}_{M\times M} A_{M\times N} = A_{M\times N} \underbrace{(\mathbb{I}_N + B_{N\times M} A_{M\times N})^{-1}}_{N\times N}$$

Using Eq. (+) with $\begin{cases} A \to X^T \\ B \to X \end{cases}$, we obtain:

$$\Leftrightarrow X^T \lambda^{-1} \left[\mathbb{I}_N + \lambda^{-1} \underbrace{X X^T}_{N \times N} \right]^{-1} \vec{y} \Leftrightarrow$$

$$X X^T = (U S V^T)(V S^T U^T) = \underbrace{(U S)}_{\| R_{N \times D}} (S^T U^T) = R R^T$$

$$\Leftrightarrow (V R^T) \lambda^{-1} \left[\mathbb{I}_N + \lambda^{-1} R R^T \right]^{-1} \vec{y} =$$

$$= (V R^T) \left[\lambda \mathbb{I}_N + R R^T \right]^{-1} \vec{y} =$$

('A' = R^T; 'B' = R; 'A' = R^T; $\vec{y}$: 'N-space')

$$= V \left[\lambda \mathbb{I}_D + \underbrace{R^T R}_{D \times D} \right] R^T \vec{y}.$$

('D-space')

It may be more numerically stable to compute $\vec{w}_{MAP}$ in N-space.

Next, consider

$$\hat{\vec{y}} = X \vec{w}_{MAP} = (U S V^T) V \left[\lambda \mathbb{I}_D + \underbrace{R^T R}_{\| \underbrace{(S^T U^T)(U S)}_{S^T_{D \times N} S_{N \times D}} \equiv \underbrace{S^2}_{\text{diag. } D \times D}} \right]^{-1} (S^T U^T) \vec{y} \Leftrightarrow$$

$$\Leftrightarrow U \underbrace{S \left[\lambda \mathbb{I}_D + S^2 \right]^{-1} S^T}_{N \times N \text{ matrix with D non-zero elements on the diagonal: } \frac{\sigma_j^2}{\sigma_j^2 + \lambda}, \; j = 1, \ldots, D} U^T \vec{y}.$$

Finally,

$$\underbrace{\hat{\vec{y}}}_{N\text{-dim}} = \sum_{j=1}^{D} \vec{u}_j \frac{\sigma_j^2}{\sigma_j^2+\lambda} \vec{u}_j^T \vec{y} \,.$$

If $\sigma_j^2 \ll \lambda$, $\frac{\sigma_j^2}{\sigma_j^2+\lambda} \simeq \frac{\sigma_j^2}{\lambda} \ll 1$, and the corresponding $\vec{u}_j$ does not contribute to $\hat{\vec{y}}$.

One can define the effective # of degrees of freedom (DoF):

$$D(\lambda) = \sum_{j=1}^{D} \frac{\sigma_j^2}{\sigma_j^2+\lambda} \,.$$

If $\lambda \to 0$, $D(\lambda) \to D$;
If $\lambda \to \infty$, $D(\lambda) \to 0$.

Now recall that

$$\hat{\vec{y}}_{MLE} = X \vec{w}_{MLE} = X(X^TX)^{-1}X^T\vec{y} =$$

$$= (USV^T)(VS^T\underbrace{U^TU}_{\mathbb{I}}SV^T)^{-1}(VS^TU^T)\vec{y} =$$

$$= (USV^T)(V\underbrace{S^2}_{D\times D}V^T)^{-1}(VS^TU^T)\vec{y} =$$

$$= (USV^T)\underbrace{(V^T)^{-1}}_{V}(S^2)^{-1}\underbrace{V^{-1}V}_{\mathbb{I}}S^TU^T\vec{y} \quad (=)$$

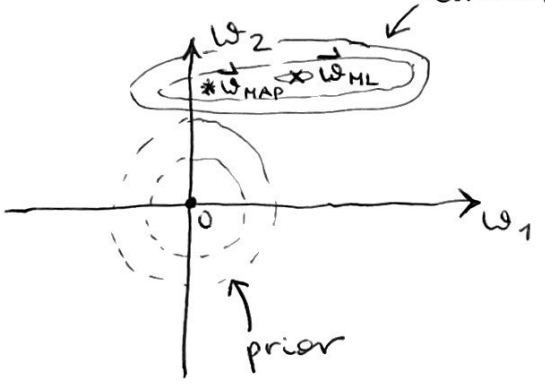
$\mathbb{I}$

(=) $\underbrace{U S (S^2)^{-1} S^T}_{N \times N \text{ matrix}} U^T \vec{y} = \sum_{j=1}^{D} \vec{u}_j \vec{u}_j^T \vec{y}$.

with D 1's on the diagonal

all $\vec{u}_j$'s are equally important, there's no regularization.

Qualitatively, the 'isobt' directions that are not strongly constrained by the data will be pulled towards $\emptyset$ by the prior, whereas the 'hard' directions will stay much closer to the ML solution:



$\vec{w} = (w_1, w_2)$

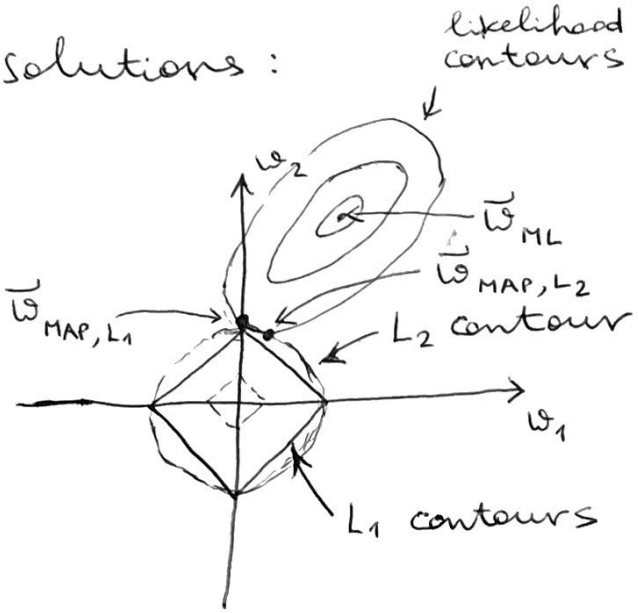
λ in the MAP procedure can be chosen using standard methods such as 5-fold cross-validation.

Lasso regression

L_1 rather than L_2 norm is employed:

$\lambda|\vec{w}|^2$ is replaced by $\lambda \sum_{j=1}^{D} |w_j|$.

This turns out to yield much sparser solutions: ← feature selection



Ex.

$\vec{w} = (1, 0)$ sparse

$\vec{w}' = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ non-sparse

$|\vec{w}|^2 = |\vec{w}'|^2 = 1$, but

$\sum_{i=1}^{2} |w_i| = 1 < \sum_{i=1}^{2} |w_i'| = \sqrt{2}$

↓

sparse solution preferred under the L_1 norm

However, using L_1 penalty requires numerical optimization ⇒ there are methods adapted specifically to this problem.

The problem is that the L_1 penalty is non-smooth ⇒ use proximal grad descent

---

Consider minimizing $J(\vec{\theta}) = \underbrace{J_1(\vec{\theta})}_{\text{smooth (e.g., -likelihood)}} + \underbrace{J_2(\vec{\theta})}_{\text{non-smooth (e.g., a lasso constraint)}}$

Proximal grad method:

$$\vec{\theta}_{t+1} = Pr\left[\underbrace{\vec{\theta}_t - \eta_t \nabla J_1(\vec{\theta}_t)}_{\vec{\theta}'_{t+1}}\right],$$

where $\Pr[\vec{\theta}^*] = \underset{\vec{z}}{\operatorname{argmin}}\left[\mathcal{I}_2(\vec{z}) + \frac{1}{2\lambda}|\vec{z}-\vec{\theta}|^2\right]$. (*)

↑ proximal operator

So, the grad step produces $\vec{\theta}'_{t+1}$ & the projection step produces $\vec{\theta}_{t+1}$ using $\vec{\theta}'_{t+1}$ as an input prm into Eq. (*).

For ex., consider a 1D constraint:

$$\mathcal{I}_2(\theta) = \begin{cases} 0 & \text{if } -a \le \theta \le a \\ \infty & \text{otherwise} \end{cases}$$

Then $$\Pr[\theta] = \begin{cases} +a, & \theta > a \\ \theta, & -a \le \theta \le a \\ -a, & \theta < -a \end{cases}$$

What about the $\mathcal{I}_1$ penalty?

$\mathcal{I}_2(\vec{z}) = \sum_{i=1}^{D} |\theta_i|$, each component can be projected separately.

For a single component θ, we need to find $\Pr[\theta]$:

$$= \underset{z}{\operatorname{argmin}}\left\{|z| + \frac{1}{2\lambda}(z-\theta)^2\right\} =$$

$$= \underset{z}{\operatorname{argmin}}\left\{\underbrace{\lambda|z| + \frac{1}{2}(z-\theta)^2}_{\text{“}f(z)}\right\}.$$

It turns out that ($\lambda > 0$)

$$P_\tau[\theta] = \begin{cases} \theta - \lambda & \theta \geq \lambda \\ 0 & |\theta| \leq \lambda \\ \theta + \lambda & \theta < -\lambda \end{cases} = \begin{array}{l} \text{soft-thresholding} \\ \text{operator:} \\ \text{Soft}(\theta, \lambda) \end{array}$$

Indeed, $\frac{df(z)}{dz} = \begin{cases} \lambda + z - \theta, & z \geq 0 \\ -\lambda + z - \theta, & z < 0 \end{cases}$

If $\theta > \lambda \Rightarrow$ expect $z^* > 0$ @ $\min$ (unique) $\quad z^* = \theta - \lambda > 0$

If $\theta < -\lambda \Rightarrow$ expect $z^* < 0 \quad z^* = \theta + \lambda < 0$

If $-\lambda \leq \theta \leq \lambda \Rightarrow f(z)$ has a minimum at $z = 0 \Rightarrow f(z) = \frac{\theta^2}{2}$ in this case.

Then Lasso can be implemented using

$$\vec{w}_{t+1} = \text{Soft}\left(\vec{w}_t - \eta_t \underbrace{\vec{g}(\vec{w})}_{\text{grad of negative log-likelihood}}, \eta_t \lambda\right) \quad (+)$$

penalty depends on step-size

Soft(...) acts on $\vec{w}$ component-by-component.

Eq. (+) implements iterative soft thresholding algorithm (ISTA)