Supervised Learning :: CHEAT SHEET

Regularization

Regularization is an effective technique to reduce overfitting.

$$\mathcal{R}_{\text{reg}}(f) = \mathcal{R}_{\text{emp}}(f) + \lambda \cdot J(f) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right) + \lambda \cdot J(f)$$

- $J(f)$: **complexity penalty**, **roughness penalty** or **regularizer**
- $\lambda \geq 0$: **complexity control** parameter
- The higher λ, the more we penalize complexity
- $\lambda = 0$: We just do simple ERM; $\lambda \to \infty$: we don't care about loss, models become as "simple" as possible
- λ is hard to set manually and is usually selected via CV
- As for $\mathcal{R}_{\text{emp}}$, $\mathcal{R}_{\text{reg}}$ and J are often defined in terms of $\boldsymbol{\theta}$:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \cdot J(\boldsymbol{\theta})$$

Ridge Regression

Use L2 penalty in linear regression:

$$\begin{aligned}\hat{\theta}_{\text{ridge}} &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2 \\ &= \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2\end{aligned}$$

Can still analytically solve this:

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

Equivalent to solving the following constrained optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)^2 \\ \text{s.t.} \quad & \|\boldsymbol{\theta}\|_2^2 \leq t\end{aligned}$$

For special case of orthonormal design $\mathbf{X}^\top\mathbf{X} = \boldsymbol{I}$, $\hat{\theta}_{\text{OLS}} = \mathbf{X}^\top\mathbf{y}$:

$$\hat{\theta}_{\text{Ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda \boldsymbol{I})^{-1}\mathbf{X}^T\mathbf{y} = ((1+\lambda)\boldsymbol{I})^{-1}\hat{\theta}_{\text{OLS}} = \frac{\hat{\theta}_{\text{OLS}}}{1+\lambda} \quad \text{(no sparsity)}$$

Geometric Analysis

Quadratic Taylor approx of unregularized $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ around its minimizer $\hat{\theta}$, where $\boldsymbol{H}$ is the Hessian of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ at $\hat{\theta}$:

$$\tilde{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\theta}) + \nabla_{\boldsymbol{\theta}}\mathcal{R}_{\text{emp}}(\hat{\theta}) \cdot (\boldsymbol{\theta} - \hat{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\theta})^T \boldsymbol{H}(\boldsymbol{\theta} - \hat{\theta})$$

Since we want a minimizer, first-order term is 0 and $\boldsymbol{H}$ is positive semidefinite:

$$\tilde{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\theta})^T \boldsymbol{H}(\boldsymbol{\theta} - \hat{\theta})$$

$$\nabla_{\boldsymbol{\theta}}\tilde{\mathcal{R}}_{\text{reg}}(\boldsymbol{\theta}) = 0 \to \hat{\boldsymbol{\theta}}_{\text{ridge}} = (\boldsymbol{H} + \lambda \boldsymbol{I})^{-1}\boldsymbol{H}\hat{\theta}$$

$\boldsymbol{H}$ is a real symmetric matrix, it can be decomposed as $\boldsymbol{H} = \boldsymbol{Q}\Sigma\boldsymbol{Q}^\top$:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{ridge}} &= \left(\boldsymbol{Q}\Sigma\boldsymbol{Q}^\top + \lambda \boldsymbol{I}\right)^{-1}\boldsymbol{Q}\Sigma\boldsymbol{Q}^\top\hat{\theta} \\ &= \left[\boldsymbol{Q}(\Sigma + \lambda \boldsymbol{I})\boldsymbol{Q}^\top\right]^{-1}\boldsymbol{Q}\Sigma\boldsymbol{Q}^\top\hat{\theta} \\ &= \boldsymbol{Q}(\Sigma + \lambda \boldsymbol{I})^{-1}\Sigma\boldsymbol{Q}^\top\hat{\theta}\end{aligned}$$

Lasso Regression

Use L1 penalty in linear regression:

$$\begin{aligned}\hat{\theta}_{\text{lasso}} &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} |\theta_j| \\ &= \arg\min_{\boldsymbol{\theta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1\end{aligned}$$

Lasso can shrink some coeffs to zero, which gives sparse solutions. However, it has difficulties handling correlated predictors.

Equivalent to solving the following constrained optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)^2 \\ \text{s.t.} \quad & \|\boldsymbol{\theta}\|_1 \leq t\end{aligned}$$

For special case of orthonormal design $\mathbf{X}^\top\mathbf{X} = \boldsymbol{I}$, $\hat{\theta}_{\text{OLS}} = \mathbf{X}^\top\mathbf{y}$:

$$\hat{\theta}_{\text{lasso}} = \text{sign}(\hat{\theta}_{\text{OLS}})(|\hat{\theta}_{\text{OLS}}| - \lambda)_+ \quad \text{(sparsity)}.$$

Function $S(\theta, \lambda) := \text{sign}(\theta)(|\theta| - \lambda)_+$ is called **soft thresholding** operator: for $|\theta| \leq \lambda$ it returns 0, whereas params $|\theta| > \lambda$ are shrunken toward 0 by λ.

Geometric Analysis

$$\tilde{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\theta})^T \boldsymbol{H}(\boldsymbol{\theta} - \hat{\theta})$$

We assume the $\boldsymbol{H}$ is diagonal, with $H_{j,j} \geq 0$

$$\tilde{\mathcal{R}}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\theta}) + \sum_j \left[\frac{1}{2}H_{j,j}(\theta_j - \hat{\theta}_j)^2\right] + \sum_j \lambda|\theta_j|$$

Minimize analytically:

$$\begin{aligned}\hat{\theta}_{\text{lasso},j} &= \text{sign}(\hat{\theta}_j)\max\left\{|\hat{\theta}_j| - \frac{\lambda}{H_{j,j}}, 0\right\} \\ &= \begin{cases}\hat{\theta}_j + \frac{\lambda}{H_{j,j}} & \text{, if } \hat{\theta}_j < -\frac{\lambda}{H_{j,j}} \\ 0 & \text{, if } \hat{\theta}_j \in [-\frac{\lambda}{H_{j,j}}, \frac{\lambda}{H_{j,j}}] \\ \hat{\theta}_j - \frac{\lambda}{H_{j,j}} & \text{, if } \hat{\theta}_j > \frac{\lambda}{H_{j,j}}\end{cases}\end{aligned}$$

If $H_{j,j} = 0$ exactly, $\hat{\theta}_{\text{lasso},j} = 0$

More Regularization Methods

Elastic Net Regression

$$\begin{aligned}\mathcal{R}_{\text{elnet}}(\boldsymbol{\theta}) &= \sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\theta}^\top\mathbf{x}^{(i)})^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{\theta}\|_2^2 \\ &= \sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\theta}^\top\mathbf{x}^{(i)})^2 + \lambda\left((1-\alpha)\|\boldsymbol{\theta}\|_1 + \alpha\|\boldsymbol{\theta}\|_2^2\right),\end{aligned}$$

$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}, \lambda = \lambda_1 + \lambda_2$

Supervised Learning :: CHEAT SHEET

Other Examples

- **L0**: not continuous or convex, NP-hard
$$\lambda \|\boldsymbol{\theta}\|_0 = \lambda \sum_j |\theta_j|^0$$
- Smoothly Clipped Absolute Deviations (SCAD): non-convex, $\gamma > 2$ controlls how fast penalty "tapers off"
$$\text{SCAD}(\theta \mid \lambda, \gamma) = \begin{cases} \lambda|\theta| & \text{if } |\theta| \leq \lambda \\ \frac{2\gamma\lambda|\theta| - \theta^2 - \lambda^2}{2(\gamma-1)} & \text{if } \lambda < |\theta| < \gamma\lambda \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |\theta| \geq \gamma\lambda \end{cases}$$
- Minimax Concave Penalty (MCP): non-convex, $\gamma > 1$ controlls how fast penalty "tapers off"
$$MCP(\theta|\lambda, \gamma) = \begin{cases} \lambda|\theta| - \frac{\theta^2}{2\gamma}, & \text{if } |\theta| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |\theta| > \gamma\lambda \end{cases}$$

Equivalence of Regularization

RRM vs MAP

Regularized risk minimization (RRM) is the same as a maximum a posteriori (MAP) estimate in Bayes.

From Bayes theorem:
$$p(\boldsymbol{\theta}|\mathbf{x}, y) = \frac{p(y|\boldsymbol{\theta}, \mathbf{x})q(\boldsymbol{\theta})}{p(y|\mathbf{x})} \propto p(y|\boldsymbol{\theta}, \mathbf{x})q(\boldsymbol{\theta})$$
The maximum a posteriori (MAP) estimator of $\boldsymbol{\theta}$ is now the minimizer of
$$-\log p\left(y \mid \boldsymbol{\theta}, \mathbf{x}\right) - \log q(\boldsymbol{\theta}).$$
Identify the loss $L\left(y, f(\mathbf{x} \mid \boldsymbol{\theta})\right)$ with $-\log(p(y|\boldsymbol{\theta}, \mathbf{x}))$:

- If $q(\boldsymbol{\theta})$ is constant (i.e., we used a uniform, non-informative prior), the second term is irrelevant and we arrive at ERM.
- If not, we can identify $J(\boldsymbol{\theta}) \propto -\log(q(\boldsymbol{\theta}))$, i.e., the log-prior corresponds to the regularizer, and the additional λ, which controls the strength of our penalty, usually influences the peakedness / inverse variance / strength of our prior.

L2 vs Weight Decay

$L2$ regularization with GD is equivalent to weight decay.

Optimize $L2$-regularized risk of a model $f(\mathbf{x} \mid \boldsymbol{\theta})$ by GD:
$$\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$
The gradient is
$$\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}$$
We iteratively update $\boldsymbol{\theta}$ by step size α times the negative gradient:
$$\begin{aligned} \boldsymbol{\theta}^{[\text{new}]} &= \boldsymbol{\theta}^{[\text{old}]} - \alpha\left(\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]}) + \lambda\boldsymbol{\theta}^{[\text{old}]}\right) \\ &= \boldsymbol{\theta}^{[\text{old}]}(1 - \alpha\lambda) - \alpha\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]}) \end{aligned}$$
We see how $\boldsymbol{\theta}^{[old]}$ decays in magnitude – for small α and λ.

Early stopping

Early stopping is another technique to aboid overfitting, which makes traning process stop when validation error stops decreasing.

1. Split training data $\mathcal{D}_{\text{train}}$ into $\mathcal{D}_{\text{subtrain}}$ and $\mathcal{D}_{\text{val}}$.
2. Train on $\mathcal{D}_{\text{subtrain}}$ and evaluate model using the validation set $\mathcal{D}_{\text{val}}$.
3. Stop training when validation error stops decreasing.
4. Use parameters of the previous step for the actual model.