1 Refresher: MAP inference

The maximum a posteriori (MAP) estimate is given by:

$$\hat{w}_{\text{map}} = \arg \max_{\vec{w}} \log p(\vec{w} | Y, X, \theta) = \arg \max_{\vec{w}} \left[ \log p(Y | X, \vec{w}) + \log p(\vec{w} | \theta) \right]. \quad (1)$$

The priors we have considered so far:

- **ridge prior**: $\vec{w} \sim \mathcal{N}(0, \frac{1}{\lambda} I)$, which has log-prior (or penalty) equal to
  $$\log p(\vec{w} | \lambda) = -\frac{1}{2} \lambda \vec{w}^T \vec{w} + \text{const}, \quad (2)$$

- **L2 smoothing prior, or Graph Laplacian**, which has log:
  $$\log p(\vec{w} | \lambda) = -\frac{1}{2} \lambda \vec{w}^T L \vec{w} + \text{const}, \quad (3)$$

  where $L$ is the graph Laplacian.

2 The key question: how to set $\lambda$?

In both of the above priors, we used $\lambda$ to denote the inverse variance of the prior. (So large $\lambda \implies$ large penalty, or narrow prior distribution; small $\lambda \implies$ small penalty, or broad prior distribution).

The question we have (so far) neglected is: how do we choose the strength of the regularizer, or in Bayesian terms, the width of the prior?

We will consider two possible approaches: (1) Cross-validation; (2) maximum marginal likelihood / evidence optimization.

3 Cross-validation

The simplest and arguably most popular approach for determining the amount of regularization is cross-validation.
The idea is to divide the data \((X, Y)\) into a training set \((X_{\text{train}}, Y_{\text{train}})\) and a test set \((X_{\text{test}}, Y_{\text{test}})\). To perform cross-validation, we consider a grid of hyperparameter values \(\{\lambda_1, \lambda_2, \ldots, \lambda_m\}\). For each value \(\lambda_j\), we will compute the MAP estimate using the training data:

\[
\hat{w}_{\text{map},j} = \arg \max_{\tilde{w}} \log p(\tilde{w}|Y_{\text{train}}, X_{\text{train}}, \lambda_j),
\]

then compute the test log-likelihood

\[
L_{j}^{\text{test}} = \log p(Y_{\text{test}}|X_{\text{test}}, \hat{w}_{\text{map},j}).
\]

The test log-likelihood \(L_{\text{test}}\) as a function of \(\lambda\) will typically have an inverted-U shape with a maximum at some intermediate value of \(\lambda\). By contrast, the training log-likelihood, equal to

\[
L_{j}^{\text{train}} = \log p(Y_{\text{train}}|X_{\text{train}}, \hat{w}_{\text{map},j}).
\]

is always a decreasing function of \(\lambda\). Values to the left of the maximum of the test log-likelihood correspond to overfitting, where the model parameters have too many degrees of freedom (i.e., the prior is too wide / the penalty is too weak), causing the fitted parameters to fit features of the training set that are not useful for predicting the test set.

Values to the right of the maximum correspond to underfitting, where the weights are too close to zero or too smooth because prior is too narrow / penalty is too strong, so the model does not have enough flexibility to capture enough of the structure of the training data.

Other topics discussed:

- **n-fold cross-validation** - instead of dividing the data into a single training and test set, n-fold cross-validation involves dividing the data into equal subsets called “folds”. We then consider \(n\) different train-test splits, where each fold is held out as test set while the remaining folds are used for training. Average the test-loglikelihood across folds in order to determine the “best” optimum for selecting \(\lambda\).

- **test vs. validation set** - in practice it is often necessary to divide data into three different sets: a training set (used to fit the parameters for each value of \(\lambda\)), a validation set (used for selecting the optimal hyperparameters \(\lambda\)), and test set (which is kept aside for reporting “true” test performance). This approach scrupulously avoids overfitting because the test set is not used for fitting hyperparameters.

- **test log-likelihood vs. test error** - in cases with Gaussian noise it is common to report test mean-squared-error (MSE) instead of test log-likelihood. Thus, we look for a minimum of the test (or validation) error, which is equivalent to a maximum of the test (or validation) log-likelihood.

4 Maximum marginal likelihood & Empirical Bayes

An alternate approach for setting hyperparameters, which has the advantage that it does not require splitting the data into training and test sets, is to maximize the marginal likelihood:
\[ \hat{\lambda} = \arg \max P(Y|X, \lambda) \]  

where recall that the marginal likelihood or evidence, which is the denominator in Bayes’ rule, is given by:

\[ P(Y|X, \lambda) = \int P(Y|X, \vec{w})P(\vec{w}|\lambda)d\vec{w}. \]  

The marginal likelihood typically has an upside-down-U shape as a function of \( \lambda \), which in many cases has maximum that is close to the maximum of the test-loglikelihood curve obtained in cross-validation.

This approach to setting hyperparameters forms the first step of an inference procedure known as Empirical Bayes. Empirical Bayes is a two-step inference method consisting of:

1. Set hyperparameters by maximum marginal likelihood
   \[ \hat{\lambda} = \arg \max P(Y|X, \lambda) \]

2. Perform MAP inference for weights given the hyperparameter estimate
   \[ \hat{\vec{w}} = \arg \max_{\vec{w}} P(\vec{w}|Y, X, \hat{\lambda}) \]