Existing Approaches

In this chapter, we briefly discuss some wrapper methods for interaction discovery and an embedded method. We explain why we choose some of these methods in our computational experiments.

Wrapper Methods

As mentioned in Chapter **??**, wrapper methods are applied on the expanded set of main-effect and interaction features. As a result, the strong hierarchy might not hold. To address this shortcoming, a solution is to first identify both relevant main and interaction effects, and then include the omitted main-effect features from the identified interactions into selection. Wrapper methods which can incorporate this approach are SFS, SFFS and GA. SBS or SBFS are ruled out because they require $p < n$ so that the backward elimination can work.

SFFS performs better than SFS since it validate the possibility of improvement of the performance criterion when some feature is excluded [@SFFS]. However, as we shall show in Chapter **??**, we include SFS as an alternative to SFFS since the latter runs into stall in some experiments. We also do not include SODA [@yangli] in comparison since this method is restricted with EBIC whereas our experiements require other evaluation measures. As a result, we choose SFS, SFFS and GA as our competitor wrapper methods.

Embedded Method

@glinternet develop **group-lasso interaction network (‘glinternet‘) by imposing a penalty function $p(\beta)$ on the coefficients when minimising the log function, $\mathcal{L}(Y;\beta)$ as follows:**

$$\arg\min_{\beta} \mathcal{L}(Y;\beta) + \lambda p(\beta)$$

λ is the tuning parameter for $\lambda > 0$. λ can be determined via a grid search with cross validations. If Y is continuous, the loss function is a squared-error loss:

$$\mathcal{L}(Y;\beta) = \frac{1}{2}\left[Y - \left(\beta_0 + \sum_{j=1}^{p}\beta_j X_j + \sum_{j<k}\beta_{j:k}X_{j:k}\right)\right]^2$$

For a binary response, the loss function is given by:

$$\mathcal{L}(Y;\beta) = -\left[Y^T\left(\beta_0 + \sum_{j=1}^{p}\beta_j X_j + \sum_{j<k}\beta_{j:k}X_{j:k}\right) - \mathbf{1}^T \log\left(\mathbf{1} + \exp(\beta_0 + \sum_{j=1}^{p}\beta_j X_j + \sum_{j<k}\beta_{j:k}X_{j:k})\right)\right]$$

The penalty function specification depends on the feature type of X. For example, if all explanatory features are continouous, $p(\beta)$ becomes:

$$p(\beta) = \sum_{j=1}^{p}|\beta_j| + \sum_{j<k}||\beta_{j:k}||_2$$

where $|\cdot|$ and $||\cdot||_2$ are absolute value and Euclidean norms. For more details, @glinternet offer more theoretical explanations, including interactions between categorical and continuous explanatory features, and interactions among categorical explanatory features as well. The tuning parameter, `glinternet` can be implemented using the R package of the same name developed by the authors [-@glinternet].