feature selection

determining which predictors should be included in a model is becoming one of the most critical questions as data are becoming increasingly high-dimensional

from a practical point of view, a model with less predictors may be more interpretable and less costly especially if there is a cost to measuring the predictors

statistically, it is often more attractive to estimate fewer parameters — also some models may be negatively affected by non-informative predictors.

feature selection

some models are naturally resistant to non-informative predictors.

tree- and rule-based models, MARS and the lasso, for example, intrinsically conduct feature selection.

for example, if a predictor is not used in any split during the construction of a tree, the prediction equation is functionally independent of the predictor

feature selection

an important distinction to be made in feature selection is that of supervised and unsupervised methods

when the outcome is ignored during the elimination of predictors, the technique is unsupervised.

removing predictors that have high correlations with other predictors or near-zero variance predictors — outcome is independent of the filtering calculations.

for supervised methods, predictors are specifically selected for the purpose of increasing accuracy or to find a subset of predictors to reduce the complexity of the model — outcome is typically used to quantify the importance of the predictors

consequences of using non-informative predictors

feature selection is primarily focused on removing non-informative or redundant predictors from the model.

the importance of feature selection depends on which model is being used. many models, especially those based on regression slopes and intercepts, will estimate parameters for every term in the model

the presence of non-informative variables can add uncertainty to the predictions and reduce the overall effectiveness of the model

consequences of using non-informative predictors

random forests show a moderate degradation in performance with the issue being that the random selection of predictors for splitting can coerce the model into including some unimportant predictors. Generally their inclusion does not seriously impact the overall model.

parametrically structured models (such as linear regression), partial least squares, and neural networks are most affected

neural networks have issues, often due to the excess number of parameters added to the model

consequences of using non-informative predictors

given the potential negative impact, there is the need to find a smaller subset of predictors.

the basic goal is to reduce their number in a way that maximizes performance — this is similar to the previous discussions:

how can we reduce complexity without negatively affecting model effectiveness?

approaches for reducing the number of predictors

apart from models with built-in feature selection, most approaches for reducing the number of predictors can be placed into two main categories

wrapper methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance

filter methods evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion

both approaches have advantages and drawbacks

filter methods are usually more computationally efficient than wrapper methods, but the selection criterion is not directly related to the effectiveness of the model

most filter methods evaluate each predictor separately, and, redundant predictors may be selected while variable interactions will not be quantified

the downside of the wrapper method is that many models are evaluated (which may also require parameter tuning) and thus an increase in computation time

there is also an increased risk of over-fitting with wrappers

wrapper methods — forward selection linear regression classical example

the predictors are evaluated (one at a time) in the current linear regression model

a statistical hypothesis test can be conducted to see if each of the newly added predictors is statistically significant (at some predefined threshold)

if at least one predictor has a p-value below the threshold, the predictor associated with the smallest value is added to the model and the process starts again

the algorithm stops when none of the p-values for the remaining predictors are statistically significant

classical forward selection for linear regression

```
Create an initial model containing only an intercept term.
repeat
    for each predictor not in the current model do
        Create a candidate model by adding the predictor to the current model
        Use a hypothesis test to estimate the statistical significance of the new model term
    end
    if the smallest p-value is less than the inclusion threshold then
        Update the current model to include a term corresponding to the most statistically significant predictor
    else
        Stop
    end
until no statistically significant predictors remain outside the model
```

in this scheme, linear regression is the base learner and forward selection is the search procedure.

the objective function is the quantity being optimized which, in this case, is statistical significance as represented by the p-value.

1. forward search is greedy — it does not reevaluate past solutions
2. the use of repeated hypothesis tests in this manner invalidates many statistical properties since the same data are being evaluated numerous times
3. maximizing statistical significance may not be the same as maximizing more relevant accuracy-based quantities

wrapper methods

suppose that the RMSE was the objective function instead of statistical significance.

the algorithm would be the same but would add predictors to the model that results in the smallest model RMSE continuing until some predefined number of predictors has been reached or the full model is used

the RMSE can be monitored to determine a point where the error began to increase and the subset size associated with the smallest RMSE is chosen

forward, backward, and stepwise selection

stepwise selection is a popular modification where, after each candidate variable is added to the model, each term is reevaluated for removal from the model

in backward selection, the initial model contains all P predictors which are then iteratively removed to determine which are not significantly contributing to the model

these procedures can be improved using non-inferential criteria, such as the AIC statistic, to add or remove predictors from the model.

recursive feature elimination is a backward selection algorithm that avoids refitting many models at each step of the search

a measure of variable importance is computed that ranks the predictors from most important to least through a model based approach (e.g., the random forest importance criterion) or using a more general approach that is independent of the full model.

the least important predictors are iteratively eliminated prior to rebuilding the model

the process continues for some predefined sequence, and the subset size corresponding to the best value of the objective function is used as the final model

while it is easy to treat the RFE algorithm as a black box, there are some considerations that should be made.

when the outcome has more than two classes, some classes may have a large degree of separation from the rest of the training set.

it may be easier to achieve smaller error rates for these classes than the others. When the predictors are ranked for selection, the predictors associated with the “easy” classes may saturate the positions for the highest ranks.

As a result, the difficult classes are neglected and maintain high error rates. In this case, class-specific importance scores can aid in selecting a more balanced set of predictors in an effort to balance the error rates across all the classes.

backward selection via the RFE algorithm

1 Tune/train the model on the training set using all P predictors
2 Calculate model performance
3 Calculate variable importance or rankings
4 **for** *each subset size* S_i, $i = 1 \ldots S$ **do**
5 | Keep the S_i most important variables
6 | [Optional] Pre-process the data
7 | Tune/train the model on the training set using S_i predictors
8 | Calculate model performance
9 | [Optional] Recalculate the rankings for each predictor
10 **end**
11 Calculate the performance profile over the S_i
12 Determine the appropriate number of predictors (i.e. the S_i associated with the best performance)
13 Fit the final model based on the optimal S_i

simulated annealing (again) for feature selection

an initial subset of predictors is selected and is used to estimate performance of the model (denoted here as E_1, for the initial error rate)

the current predictor subset is slightly changed, and another model is created with an estimated error rate of E_2

if the new model is an improvement over the previous one (i.e., $E_2 < E_1$), the new feature set is accepted

if it is worse, it may still be accepted based on some probability p_{a_i}, where i is the iteration of the process

simulated annealing (again) for feature selection

this probability is configured to decrease over time so that, as i becomes large, it becomes very unlikely that a suboptimal configuration will be accepted

the process continues for some pre-specified number of iterations and the best variable subset across all the iterations is used

the idea is to avoid a local optimum (a solution that is currently best but is not best overall).

by accepting “bad” solutions, the algorithm is able to continue the search in other spaces and therefore is less greedy

simulated annealing (again)

Generate an initial random subset of predictors
for *iterations* $i = 1 \ldots t$ **do**
  Randomly perturb the current best predictor set
  [Optional] Pre-process the data
  Tune/train the model using this predictor set
  Calculate model performance (E_i)
  if $E_i < E_{best}$ **then**
    Accept current predictor set as best
    Set $E_{best} = E_i$
  else
    Calculate the probability of accepting the current predictor set

$$p_i^a = \exp\left[(E_{best} - E_i)/T\right]$$

    Generate a random number U between $[0, 1]$
    if $p_i^a \leq U$ **then**
      Accept current predictor set as best
      Set $E_{best} = E_i$
    else
      Keep current best predictor set
    end
  end
end
Determine the predictor set associated with the smallest E_i across all iterations
Finalize the model with this predictor set

genetic algorithm (again) for feature selection

the problem of feature selection is inherently a complex optimization problem, where we seek the combination of features that provides an optimal prediction of the response

to employ GAs towards this end, we must frame the feature selection problem in terms of the GA machinery.

genetic algorithm (again)

in the context of feature selection, the chromosome is a binary vector that has the same length as the number of predictors in the data set

each binary entry of the chromosome, or gene, represents the presence or absence of each predictor in the data

the fitness of the chromosome is determined by the model using the predictors indicated by the binary vector

GAs are therefore tasked with finding optimal solutions from the 2^n possible combinations of predictor sets.

genetic algorithm (again)

to begin the search process, GAs are often initiated with a random selection of chromosomes from the population of all possible chromosomes.

each chromosome’s fitness is computed, which determines the likelihood of the chromosome’s selection for the process of reproduction

two chromosomes from the current population are then selected based on the fitness criterion and are allowed to reproduce.

genetic algorithm (again)

in the reproduction phase, the two parent chromosomes are split at a random position (also called loci), and the head of one chromosome is combined with the tail of the other chromosome and vice versa

after crossover, the individual entries of the new chromosomes can be randomly selected for mutation in which the current binary value is changed to the other value

genetic algorithm (again)

the crossover phase drives subsequent generations towards optimums in subspaces of similar genetic material

the search subspace is narrowed to the space defined by the most fit chromosomes

the algorithm could become trapped in a local optimum— in the context of feature selection, this means that the selected features may produce an optimal model, but other more optimal feature subsets may exist

genetic algorithm (again)

the mutation phase enables the algorithm to escape local optimums by randomly perturbing the genetic material

usually the probability of mutation is kept low (say, $p_m < 0.05$).

If their are concerns about local optimums, then the mutation probability can be raised

the effect of raising the mutation probability is a slowing of the convergence to an optimal solution

genetic algorithm (again)

```
1  Define the stopping criteria, number of children for each generation
   (GenSize), and probability of mutation (p_m)
2  Generate an initial random set of m binary chromosomes, each of
   length p
3  repeat
4      for each chromosome do
5          Tune and train a model and compute each chromosome's
           fitness
6      end
7      for reproduction k = 1 ... GenSize/2 do
8          Select two chromosomes based on the fitness criterion
9          Crossover: Randomly select a loci and exchange each
           chromosome's genes beyond the loci
10         Mutation: Randomly change binary values of each gene in
           each new child chromosome with probability, p_m
11     end
12 until stopping criteria is met
```

filter methods

filter methods evaluate the predictors prior to training the model, and, based on this evaluation, a subset of predictors are entered into the model.

Most of these techniques are uni- variate, meaning that they evaluate each predictor in isolation. In this case, the existence of correlated predictors makes it possible to select important, but redundant, predictors.

The obvious consequences of this issue are that too many predictors are chosen and, as a result, collinearity problems arise.

filter methods

if hypothesis tests are used to determine which predictors have statistically significant relationships with the outcome (such as the t-test), the problem of multiplicity can occur.

For example, if a confidence level of $\alpha = 0.05$ is used as a p-value threshold for significance, each individual test has a theoretical false-positive rate of 5%. However, when a large number of simultaneous statistical tests are conducted, the overall false-positive probability increases exponentially.