Notes on Feature Selection

Shasha Liao
Georgia Tech

October 26, 2020

1 Motivation

- Simplify model
- More data efficient
- Better interpretation
- Increase accuracy by eliminating noise features
- Enhance generalization by reducing overfitting

2 Major approaches for feature selection

- Combination: evaluation metrics and search technique
- Evaluation metrics: Heuristics, information theoretic metirc, bias-variance tradeoffs, error probability
- Search technique: subset selection, l1-regularization

2.1 Information Theoretic Metric: Quantify the Uncertainty

- Select the variables that contain the most important information for the response. How to measure the importance of a variable/feature?
- We are uncertain about the response Y before having any feature. Quantify the uncertainty using **entropy** $H(Y)$
- Given a particular feature X_i, the uncertainty of Y reduces. Quantify the new uncertainty using **conditional entropy** $H(Y|X_i)$
- Reduction in uncertainty is the informativeness of feature X_i. Quantify the reduction in uncertainty using **mutual information** $I(X_i, Y) = H(Y) - H(X_i|Y)$.

2.2 Entropy: Quantify the Uncertainty

- Entropy $H(Y)$ of a discrete random variable Y:

$$H(Y) = -\sum_{k=1}^{K} P(y = k) \log_2 P(y = k).$$

  Ex: Binary case: if $P(y = 1) = p$, then

$$H(Y) = -p \log_2 p - (1 - p) \log_2 (1 - p).$$

- Conditional entropy $H(Y|X_i)$ of a random variable given a continuous feature X_i:

$$\begin{aligned} H(Y|X_i) &= \int H(Y|X_i = x_i) p(x_i) dx_i \\ &= -\int \left(\sum_{k=1}^{K} P(y = k|X_i = x_i) \log_2 P(y = k|X_i = x_i) \right) p(x_i) dx_i. \end{aligned}$$

  $H(Y|X_i)$ integrated the information from X_i and quantifies the remaining uncertainty in Y after seeing X_i.

- Mutual information:

$$I(X_i, Y) = H(Y) - H(X_i|Y)$$

  Quantify the reduction in uncertainty in Y after seeing X_i. The more the reduction in entropy, the more informative the feature X_i is.

- Mutual information can capture nonlinear dependence. F-test captures linear capture.

2.3 A feature selection algorithm

- Given a dataset $S = \{(x^1, y^1), (x^2, y^2), ..., (x^m, y^m)\}$, $x \in R^n$, $y \in \{1, 2, ..., K\}$.
- 1, For each feature x_i, estimate density $p(x_i)$
- 2, For each class $y = c$, estimate density $p(y = c)$
- 3, For each class $y = c$ and each feature x_i, estimate joint density $p(y = c, x_i)$ which equals to $p(x_i|y = c)p(y = c)$.
  Score feature x_i on class $y = c$ using MI (Mutual Information):

$$I_{c,i} = \int \sum_{c=1}^{K} p(x_i, y = c) \log_2 \frac{p(x_i, y = c)}{p(x_i)p(y = c)} dx_i.$$

  Choose those feature x_i for class c with high $I_{c,i}$ score.

See the slides for an interesting example of using MI to extract key words for documents of different classes.

3 Search technique

- Linear Regression Model

$$y = \theta^T x + \epsilon.$$

Use the Least-Square method to fit the linear regression model. That is to find θ to minimize the mean square error, which is

$$\theta = (XX^T)^{-1}Xy.$$

What if $X^T X$ is not invertible when our variables are highly correlated? If this happens, our θ will be very large. We can use regression to put a penalty on large values of θ.

- Ridge Regularization

$$\theta^r = argmin_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \theta^T x^i)^2 + \lambda\|\theta\|_2^2,$$

where λ is called the **regularization parameter**, which can be tuned by cross validation.
Now take the gradient of $L(\theta)$ to be zero and we obtain

$$\theta^r = (\frac{1}{m}XX^T + \lambda I)^{-1}(\frac{1}{m}Xy).$$

Note that here $\frac{1}{m}XX^T + \lambda I$ is for sure invertible (if $\lambda > 0$) as it has the smallest eigenvalue λ. And if we choose a different λ, we will have a different solution.

Ridge Regularization will not set coefficients directly to zero. Instead, as we increase the value of λ, Ridge Regularization sends the coefficients of unimportant features to be very close to zero.

It is an NP-hard problem to find a subset of variables that are most "important". When there are n variables, there are 2^n ways to select a subset of variable. This is related to solve

$$argmin_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \theta^T x^i)^2 + \lambda\|\theta\|_0,$$

where $\|\theta\|_0$ is the number of nonzero entries in θ. It is hard to solve because $\|\cdot\|_0$ is non-convex. LASSO address this issue by *convex relaxation.*

- LASSO Regularization: Least absolute shrinkage and selection operator

$$argmin_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \theta^T x^i)^2 + \lambda\|\theta\|_1.$$

1, Regularizer λ controls the model complexity. Large λ gives simpler model!
2, With the L1 norm, we have a convex problem, which can be solved efficiently!

3, L1 penalty can also be used for other types of algorithms to encourage sparsity in solution.

- Elastic Net
  * Ridge regression: helps when variables are correlated but cannot perform variable selection
  * Lasso: helps with variable selection, not stable when variables are correlated
  * Elastic Net: combines two approaches by choosing $\alpha \in [0, 1]$ and solve

$$\min \frac{1}{m} \sum_{i=1}^{m} (y_i - \theta^T x^i)^2 + \lambda(\alpha \|\theta\|_2^2 + (1 - \alpha)\|\theta\|_1).$$

4

-