Berkeley Engineering | BerkeleyHaas

PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Office Hour #9 with Matilde D'Amelio

May 12, 2022 at 9 pm UTC

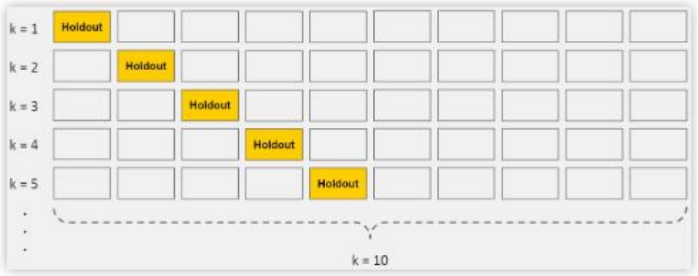
Prevent Overfitting

Get More Training Data

Cross-Validation & HoldOut: Use your initial training data to generate multiple mini train-test splits. Use these splits to tune your mode

Augmentation: If you can't get more data, you can try augmentation to add variation in your data. Augmentation means artificially modifying your existing data by means of transforms that resemble the variation you might expect in the real data.

Feature Selection: it is a process by which you automatically search for the best subset of attributes in your dataset.



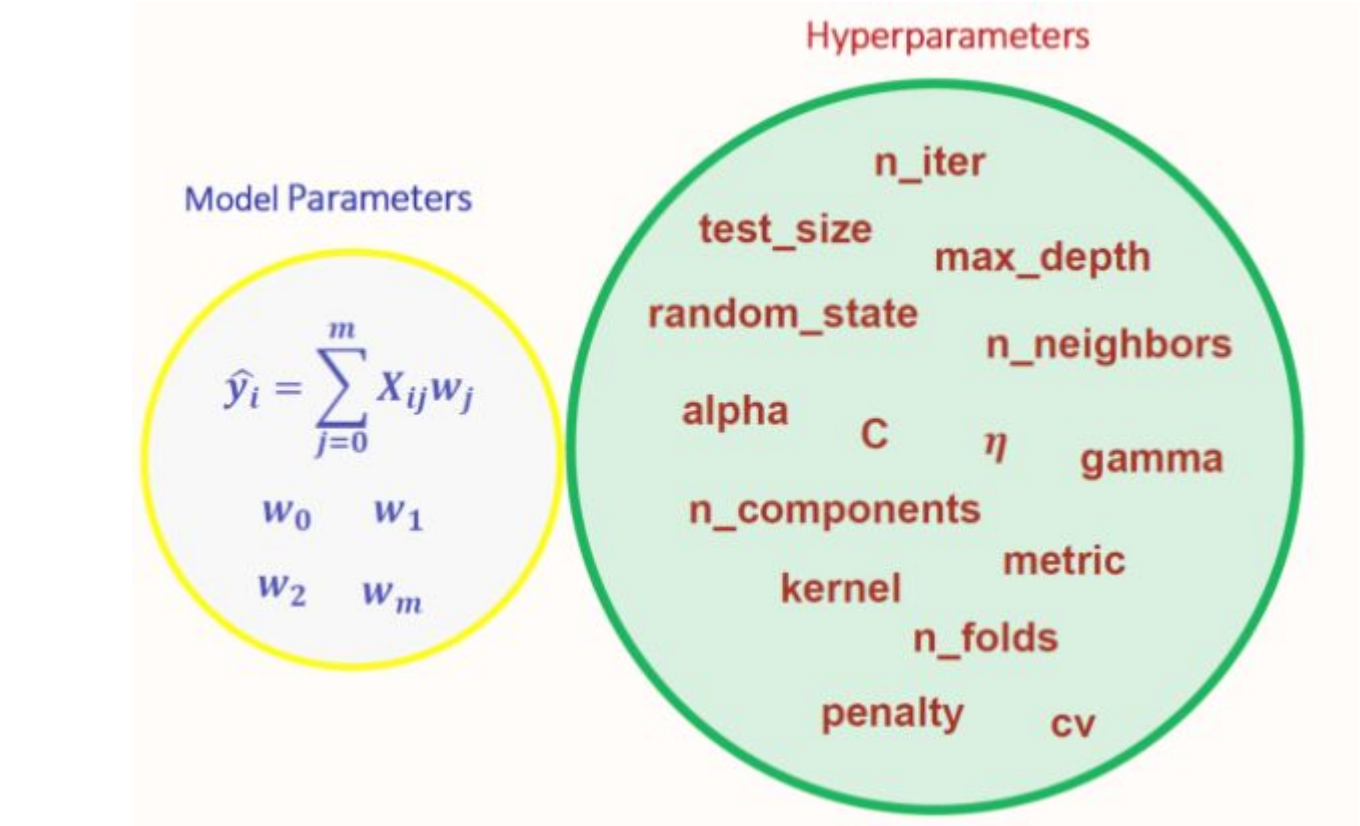
Overfitting

Overfitting refers to a model that models the training data too well.

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns.

Automated Hyperparameter Selection



Hyperparameters: adjustable parameters that must be tuned in order to obtain a model with optimal performance (they can be also polynomial)

Sequential Feature Selection

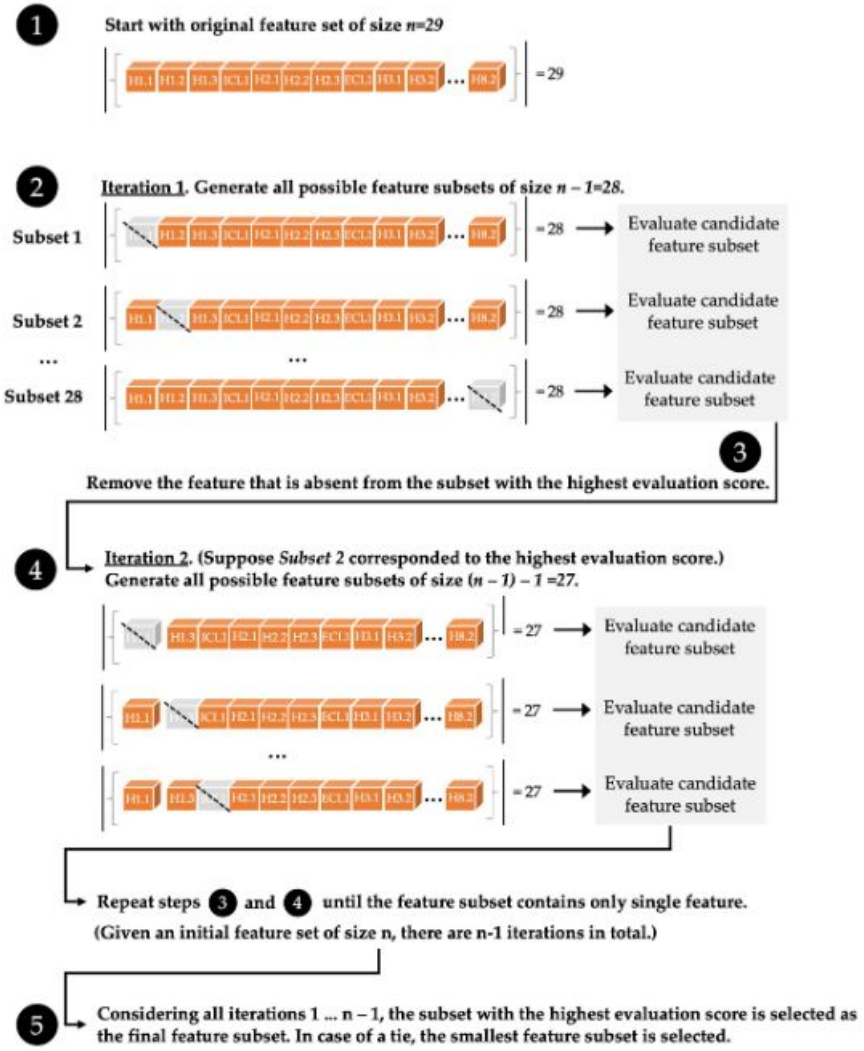
Sequential feature selection algorithms are algorithms that are used to reduce an initial *d*-dimensional feature space to a *k*-dimensional feature subspace where $k < d$. The motivation behind feature selection algorithms is to **automatically select a subset of features that is most relevant to the problem**. The goal of feature selection is two-fold: We want to improve the computational efficiency and reduce the generalization error of the model by removing irrelevant features or noise.

In a nutshell, SFAs remove or add one feature at the time based on the classifier performance until a feature subset of the desired size *k* is reached. There are 4 different flavors of SFAs available:

1. Sequential Forward Selection (SFS)
2. Sequential Backward Selection (SBS)
3. Sequential Forward Floating Selection (SFFS)
4. Sequential Backward Floating Selection (SBFS)

The ***floating*** variants, SFFS and SBFS, can be considered as extensions to the simpler SFS and SBS algorithms. The floating algorithms have an additional exclusion or inclusion step to remove features once they were included (or excluded), so that a larger number of feature subset combinations can be sampled. It is important to emphasize that this step is conditional and only occurs if the resulting feature subset is assessed as "better" by the criterion function after removal (or addition) of a particular feature.

Sequential Backward Selection



Regularization

This technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

A simple relation for linear regression looks like this.

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

The fitting procedure involves a loss function, known as residual sum of squares or RSS. The coefficients are chosen, such that they minimize this loss function.

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

If there is noise in the training data, then the estimated coefficients won't generalize well to the future data. This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.

Ridge Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

RSS is modified by adding the shrinkage quantity. λ is the tuning parameter that decides how much we want to penalize the flexibility of our model. The increase in flexibility of a model is represented by increase in its coefficients, and if we want to minimize the above function, then these coefficients need to be small. This is how the Ridge regression technique prevents coefficients from rising too high. Also, notice that we shrink the estimated association of each variable with the response, except the intercept β0.

Selecting a good value of λ is critical. Cross validation comes in handy for this purpose. ***W***

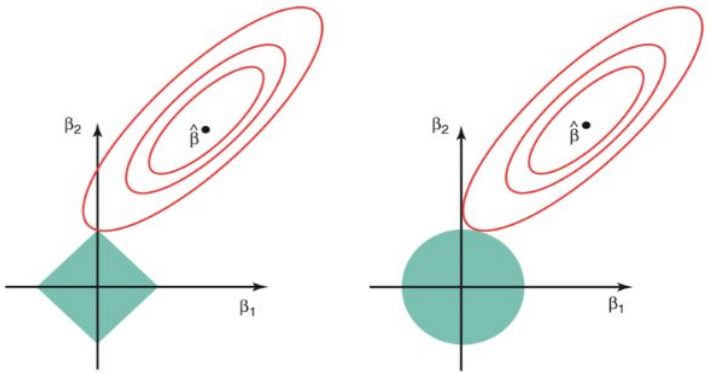
ed to standardize the predictors or bring the predictors to the same scale before performing ridge regression. The formula used to do this is given below.

$$\tilde{x}_{ij}=\frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\overline{x}_j)^2}}.$$

Lasso Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

This variation differs from ridge regression only in penalizing the high coefficients. It uses |βj|(modulus)instead of squares of β, as its penalty.



Constraint functions(green areas), for lasso(left) and ridge regression(right), along with contours for RSS(red ellipse). Points on the ellipse share the value of RSS. The lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region. ***Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coefficient estimates will be exclusively non-zero. However, the lasso constraint has corners at each of the axes, and so the ellipse will often intersect the constraint region at an axis. When this occurs, one of the coefficients will equal zero.*** **This sheds light on the obvious disadvantage of ridge regression, which is model interpretability.** It will shrink the coefficients for least important predictors, very close to zero. However, in the case of the lasso, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. **Therefore, the lasso method also performs variable selection and is said to yield sparse models**

QUESTIONS?