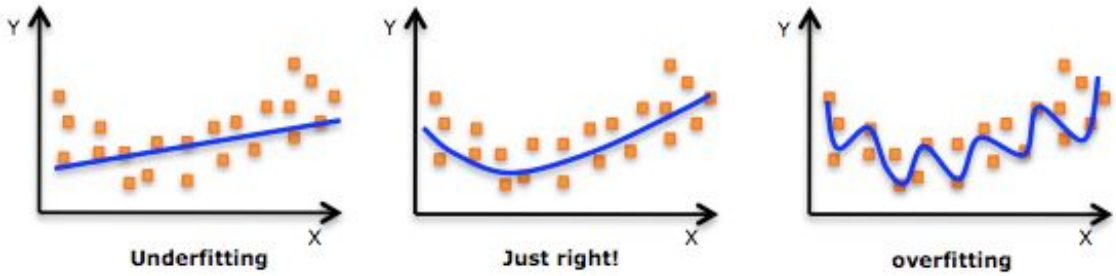
Data Scientist 高频面试题

1. Machine Learning

1. What is overfitting?(过拟合) / Please briefly describe what is bias(偏差) vs. variance(方差).

1) Overfitting

Overfitting is a **modeling error** which occurs when a **function** is too **closely fit** to **a limited set of data points**. The essence of overfitting is to have unknowingly **extracted** some of the **residual variation** (i.e. the noise) as if that variation represented underlying model structure.



2) bias VS variance

Error due to Biasexpected (or average) prediction

Error due to Variance: The error due to variance is taken as the **variability** of **a model prediction** for a given data point.

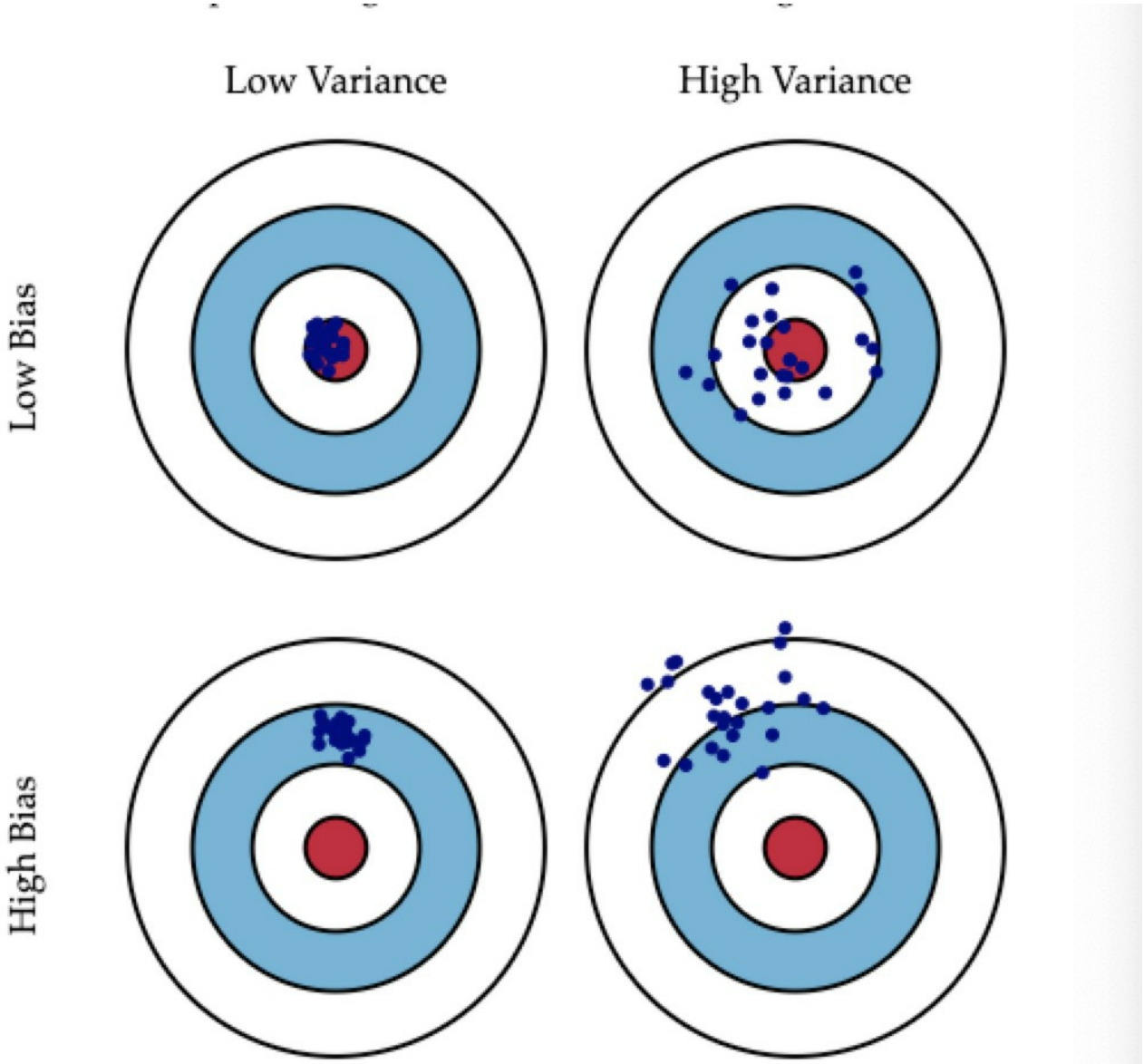


Fig. 1 Graphical illustration of bias and variance.

2. How do you overcome overfitting? Please list 3-5 practical experience. / What is 'Dimension Curse'? How to prevent?

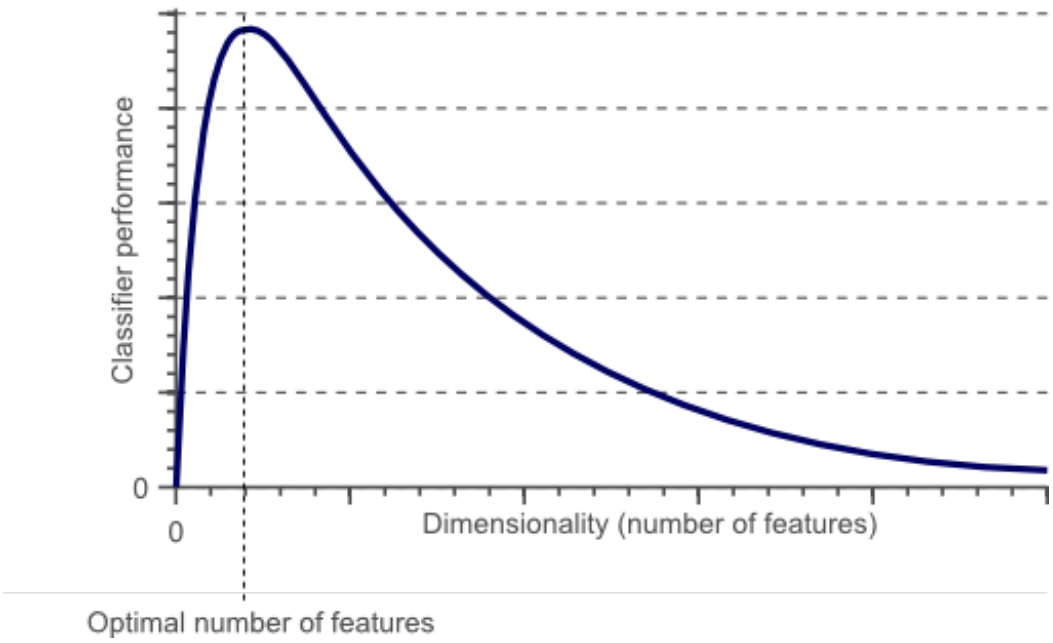
1) Overcome Overfitting

1. Add more data
2. Use data augmentation (image rotation(旋转), translations(移位), shift(平移) , 尺度变化(scale), 噪声扰动(noise))
3. Use architectures that generalize well
4. Add regularization (mostly dropout, L1/L2 regularization are also possible)

5. Reduce architecture complexity

2) Dimension Curse(维度灾难)

In fact, after **a certain point**, increasing **the dimensionality** of the problem by **adding new features** would actually **degrade the performance of our classifier**. This is illustrated by figure 1, and is often referred to as **'The Curse of Dimensionality'** .



Solution

1. Dimension reduction

1) PCA

2) Feature Selection

3. Please briefly describe the Random Forest classifier. How did it work? Any pros and cons in practical implementation?

Random Forest Classifier is **ensemble algorithm**. Ensembled algorithms are those which **combines more than one algorithms** of same or different kind for **classifying objects**. Random forest classifier creates **a set of decision trees** from **randomly selected subset** of **training set**. It then **aggregates the votes** from different decision trees to decide the **final class** of the test object.

Advantage

1) As we mentioned earlier a single decision tree tends to overfit the data. The

process of averaging or combining the results of different decision trees helps to overcome the problem of overfitting.
2) Random forests also have less variance than a single decision tree. It means that it works correctly for a large range of data items than single decision trees.
3) Random forests are extremely flexible and have very high accuracy.
3) They also do not require preparation of the input data. You do not have to scale the data.
4) It also maintains accuracy even when a large proportion of the data are missing.

Disadvantage
1) The main disadvantage of Random forests is their **complexity**. They are much harder and time-consuming to construct than decision trees.
2) They also require **more computational resources** and are also **less intuitive**. When you have a large collection of decision trees it is hard to have an intuitive grasp of the relationship existing in the input data.
3) In addition, the **prediction process** using random forests is **time-consuming** than other algorithms.

4. Please describe the difference between Gradient Boosting Tree model and Random Forest.

error = bias + variance

Boostingweak learners (high bias, low variance)shallow treesdecision stumpsBoostingerrorreducing bias

Random Forest uses as you said **fully grown decision trees** (low bias, high variance). It tackles the **error reduction** task in the **opposite** way: by **reducing variance**. The trees are made uncorrelated to **maximize the decrease in variance**, but the algorithm cannot **reduce bias** (which is **slightly higher** than the bias of an individual tree in the forest). Hence the need for large, unpruned trees, so that the bias is initially **as low as possible**.

5. What is SVM? what parameters you will need to tune during model training? How is different kernel changing the classification result?

1) **SVM(Support Vector Machine)**
A **Support Vector Machine (SVM)** is a **discriminative classifier** formally defined by a **separating hyperplane**. In other words, given **labeled training data** (supervised learning), the algorithm outputs an **optimal hyperplane** which **categorizes new examples**. In **two dimentional space** this **hyperplane** is **a line dividing a plane** in two parts where in each class lay in either side.

2) **Tuning parameters**: Kernel, Regularization, Gamma and Margin.
Kernel: The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

Regularization: The Regularization parameter tells the SVM optimization how much you want to avoid misclassifying each training example.

Gamma: High Gamma -> Only nearby points are considered
Low Gamma -> Far away points are alos considered

Margin: A margin is a separation of line to the closest class points. (Penalty parameter C of the error term.)

6. Briefly rephrase PCA in your own way. How does it work? And tell some goods and bads about it.

Principal Component Analysis (PCA) is a **dimension-reduction** tool that can be used to **reduce a large set of variables** to a small set that still **contains most of the information** in the large set.

Principal component analysis (PCA) is a mathematical procedure that **transforms a number of (possibly) correlated variables** into **a (smaller) number of uncorrelated variables** called **principal components**.

The **first principal component** accounts for **as much of the variability in the data** as possible, and each **succeeding component** accounts for as much of the **remaining variability** as possible.

Advantage
1. Removes Correlated Features

2. Improves Algorithm Performance
3. Reduces Overfitting
4. Improves Visualization

Disadvantage
1. Independent variables become less interpretable
2. Data standardization is must before PCA
3. Information Loss

7). Why doesn't logistic regression use R^2?

R-squared tells us how much of the difference in outcome is explained by the model.

8). When will you use L1 regularization compared to L2?

L1会趋向于产生少量的特征(sparse模型)，而其他的特征都是0。对于large-scale的问题来说这一点很重要，因为可以减少存储空间。缺点是加入L1后目标函数在原点不可导(not differentiable)，需要做特殊处理。

L2会选择更多的特征，这些特征都会接近于0。Lasso在特征选择时候非常有用，而Ridge就只是一种规则化而已。

L0范数和L1范数都能够达到使参数稀疏的目的，但L0范数更难优化求解，L1范数是L1的最优凸相似且更易求解，故得到广泛的应用。

L2范数主要作用是防止模型过拟合，提高模型的泛化能力。

9). List out at least 4 metrics you will use to evaluate model performance and tell the advantage for each of them. (F1 score, ROC curve, recall, etc...)

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$F - measure = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

10). What would you do if you have > 30% missing value in an important field before building the model?

1) **Deleting Rows**

Pros:

1. Complete removal of data with missing values results in robust and highly accurate model
2. Deleting a particular row or a column with no specific information is better, since it does not have a high weightage

Cons:

1. Loss of information and data
2. Works poorly if the percentage of missing values is high (say 30%), compared to the whole dataset

2) **Replacing With Mean/Median/Mode**

Pros:

1. This is a better approach when the **data size** is **small**
2. It can prevent data loss which results in removal of the rows and columns

Cons:

1. Imputing the approximations **add variance and bias**
2. Works poorly compared to other multiple-imputations method

3) **Assigning An Unique Category**

Pros:

1. Less possibilities with one extra category, resulting in low variance after one hot encoding — since it is categorical
2. Negates the loss of data by adding an unique category

Cons:

1. Adds less variance
2. Adds another feature to the model while encoding, which may result in poor

performance

4) **Predicting The Missing Values**

Pros:

1. Imputing the **missing variable** is an improvement as long as the bias from the same is smaller than the omitted variable bias
2. Yields **unbiased estimates** of the model parameters

Cons:

1. Bias also arises when an **incomplete conditioning set** is used for a **categorical variable**
2. Considered only as a proxy for the **true values**

5) **Using Algorithms Which Support Missing Values**

Pros:

1. Does not require creation of a predictive model for each attribute with missing data in the dataset
2. **Correlation** of the data is **neglected**

Cons:

1. Is a very **time consuming process** and it can be **critical** in data mining where **large databases are being extracted**
2. Choice of **distance functions** can be Euclidean, Manhattan etc. which is do not yield a robust result