Math & Machine Learning

What is cross validation?

How do you do it correctly?

Math & Machine Learning

Is it better to design robust or accurate algorithms?

It’s a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. Mainly used in settings where the goal is prediction and one wants to estimate how accurately a model will perform in practice. The goal of cross-validation is to define a data set to test the model in the training phase (i.e. validation data set) in order to limit problems like overfitting, and get an insight on how the model will generalize to an independent data set.

Cross-validation is also used to select model hyperparameters by finding the set of hyperparameters that give a minimum or maximum value of interest in cross-validation tests.

Examples: leave-one-out cross validation, K-fold cross validation

How to do it right?

* The training and validation data sets have to be drawn from the same population.
* Any dependence of the target variable on other target variables significantly complicates cross-validation. This can occur in time-series and spatial data.

* The ultimate goal is to design systems with good generalization capacity, that is, systems that correctly identify patterns in data instances not seen before.
* The generalization performance of a learning system strongly depends on the complexity of the model assumed.
* If the model is too simple, the system can only capture the actual data regularities in a rough manner. In this case, the system has poor generalization properties and is said to suffer from underfitting. Can interpret this as a high bias model.
* By contrast, when the model is too complex, the system can identify accidental patterns in the training data that need not be present in the test set. These spurious patterns can be the result of random fluctuations or of measurement errors during the data collection process. In this case, the generalization capacity is also poor. The learning system is said to be affected by overfitting. Can interpret this as high variance.
* Spurious patterns, which are only present by accident in the data, tend to have complex forms. This is the idea behind the principle of Occam’s razor for avoiding overfitting: simpler models are preferred if more complex models do not significantly improve the quality of the description for the observations
* Quick response: Occam’s Razor. It depends on the learning task. Choose the right balance
* Ensemble learning can help balancing bias/variance (several weak learners together = strong learner).

Math & Machine Learning

How are model metrics defined and selected?

Math & Machine Learning

Explain what regularization is and why it’s useful.
Name and describe several specific methods.

The selection of model predictive metrics depends on whether it's a regression or classification problem.

For regression problems, common metrics are:
RSS (residual sum of squares), RMSE (root mean squared error), MAE (mean absolute error), WMAE(weighted mean absolute error), RMSLE (root mean squared logarithmic error). Try to write out each one mathematically.

For classification problems, common metrics are:
Accuracy: TP + TN / (TP + TN + FP + FN)
Recall / Sensitivity / True Positive Rate: TP / (TP + FN)
Precision / Positive Predictive Rate: TP / (TP + FP)
Specificity / True Negative Rate: TN / (TN + FP)

ROC & AUC: For a binary classification problem, the ROC plots the True Positive Rate vs the False Positive Rate (1- Specificity). Ideally, at a FPR of 0 the TPR will be 1, and that would yield an AUC (area under the curve) of 1. The ROC illustrates how the probability threshold at which the classification occurs affects the classification.

Regularization adds a term to the least squares loss function that penalizes the magnitude of the model coefficients. This term is called a penalty, or shrinkage term. It is used to prevent overfitting and thereby improve the generalization of a model (increase bias and decrease variance). We have covered L1 (Lasso) and L2 (Ridge) regularization techniques.

In both cases the penalty term is a function of the model coefficients. This term includes a value, lambda, that affects how sensitive the total cost function is to the penalty term.

In the case of L1, the penalty term is lambda multiplied by the sum of the absolute value of the model coefficients. Lasso (L1) zeros out some coefficients entirely. A disadvantage of this method is that this selection can be arbitrary.

In the case of L2, the shrinkage term is lambda multiplied by the sum of the square value of the coefficients. Ridge (L2) maintains all features in the data set. Ridge regression tends to perform better than Lasso when the coefficients are correlated.

Both L1 and L2 help deal with collinearity issues. A combination of the two is termed an Elastic Net that usually combines the advantages of both.

Math & Machine Learning

Explain what a local optimum is and why it is important in a specific context, such as K-means clustering.

What are specific ways of determining if you have a local optimum problem? What can be done to avoid local optima?

Math & Machine Learning

Assume you need to generate a predictive model using multiple regression. Explain how you intend to validate this model

A local optimum is a solution that is optimal within a neighboring set of candidate solutions. This is in contrast with a global optimum that is the optimal solution among all solutions. Often in machine learning the model coefficients are fit in such a way that the cost function associated with the predictive error is minimized. The question of whether the cost function is at a local minimum or a global minimum arises in this context.

In K-means clustering an objective cost function will always decrease until a local optimum is reached. However, cluster results will depend on the initial random cluster assignment.

Differing initializations resulting in different clusters is evidence of a local minimum problem.

This problem can be addressed for K-Means by repeating the clusters for many different intialization values and then taking the solution that has the lowest cost.

You can validate the model using:

$R^2$: this coefficient of determination quantifies the fraction of the variance of the predicted (dependent) variable that can be explained by the independent variable. However having a large $R^2$ is not enough. In fact, you can always increase $R^2$ by adding more variables but that doesn't mean your model is better or "validated"

Analysis of residuals:
- Check homoskedasticity (is the variance from the regression line the same for all values of the predictor variable? It shouldn't increase or decrease as the predictor variable changes.)
- The residuals should be normally distributed.
- The target variables, or anything from one row of data to the next, should not be dependent on each other (time series and spatial data can violate this.) No multicollinearity.

Out-of-sample evaluation:
Cross-validation (then checking with $R^2$)

Math & Machine Learning

What is latent semantic indexing?
What is it used for?

Math & Machine Learning

Explain what resampling methods are and why they are useful.

Latent semantic indexing and retrieval is a method that uses singular value decomposition to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. It is based on the principle that words that are used in the same contexts tend to have similar meanings. For example: two synonyms may never occur in the same passage but should nonetheless have highly associated representations

Latent semantic indexing is used for:

- Learning correct word meanings
- Subject matter comprehension
- Information retrieval
- Sentiment analysis (social network analysis)

Resampling methods repeatedly draw samples from a given sample in order to provide alternative representations of that sample. Resampling is commonly used to generate alternative training sets so that a model of interest can be refit on each set in order to obtain additional information about the fitted model.

For example: repeatedly draw different samples from training data, fit a linear regression to each new sample, and then examine the extent to which the resulting fit differs.

Cross-validation and bootstraping both utilize resampling.

In cross-validation the the data set is randomly split into training and test data sets (using random sampling with no replactement) and then the model is trained on the training set and evaluated on the test set to evaluate model performance.

In bootstrapping data is drawn with replacement from the dataset. Bootstrapping is mostly used to quantify the uncertainty associated with a given estimator or statistical learning method, but it can also be used to provide alternative datasets for a model to train on (such as in random forests).

Math & Machine Learning

What is principal component analysis?
Explain the sort of problems you would use PCA for.
Also explain its limitations as a method.

Math & Machine Learning

Explain what false positives and false negatives are.
Provide examples when
false positives are more important than false negatives, and
false negatives are more important than false positives.

PCA is a statistical method that uses an orthogonal transformation to convert a set of observations of correlated variables into a set of values of linearly uncorrelated variables called principal components. It's a form of dimensionality reduction.

In PCA data is reduced from m to k dimensions (k <= m) where the goals is to find the k vectors onto which to project the data so as to minimize the projection error.

Algorithm:
1) Preprocessing (standardization)
2) Compute covariance matrix Σ
3) Compute eigenvectors and eigenvalues of Σ
4) Choose k principal components so as to retain x% of the variance (typically x=99) by summing the eigenvalues.

PCA is used in compression (reducing disk/memory needed to store data), predictive models, data visualization, PCR (principal component regression), and survey and polling analyses.

Limitations:
- PCA is not scale invariant
- The directions with largest variance are assumed to be of most interest
- Only considers orthogonal transformations (rotations) of the original variables
- If the variables are correlated, PCA can achieve dimension reduction. If not, PCA just orders them according to their variances.

False positive
* Improperly reporting the presence of a condition when it's not in reality. Example: HIV positive test when the patient is actually HIV negative

False negative
* Improperly reporting the absence of a condition when in reality it's the case. Example: not detecting a disease when the patient has this disease.

When false positives are more important than false negatives:
- In a non-contagious disease, where treatment delay doesn't have any long-term consequences but the treatment itself is grueling.
- HIV test: psychological impact

When false negatives are more important than false positives:
- If early treatment is important for good outcomes
- In quality control: a defective item passes through the cracks!
- Software testing: a test to catch a virus has failed

Math & Machine Learning

What is the difference between supervised and unsupervised learning? Give concrete examples.

Math & Machine Learning

When would you use random forests instead of Support-Vector-Machines and why?

Supervised learning: predictors (or features) are associated with a response (target); we wish to fit a model that relates features to targets for better understanding the relation between them (inference) or with the aim to accurately predicting the target for future observations (prediction).
Unsupervised learning: there isn't a response (target) that can supervise the analysis. Unsupervised learning instead tends to aim towards organizing for finding structure in the data.

Supervised learning machine learning techniques: support vector machines, neural networks, linear regression, logistic regression, extreme gradient boosting
Unsupervised learning machine learning techniques: clustering (hierarchecal, k-means, density-based), principal component analysis, singular value decomposition; identify group of customers, non-negative matrix factorization (NMF), self-organizing maps.

Supervised learning examples: predict the price of a house based on the area, size; churn prediction; predict the relevance of search engine results.
Unsupervised learning examples: find customer segments; image segmentation; classify senators by their voting.

In a case of a multi-class classification problem: SVM will require one-against-all method (memory intensive).
If one needs to know the feature importances.
If one needs to get a model fast (SVM is long to tune, need to choose the appropriate kernel and its parameters, for instance sigma and epsilon)
In a semi-supervised learning context (random forest and dissimilarity measure): SVM can work only in a supervised learning mode.

JB comments: "Never use SVMs is a pretty safe bet..."
"Regression based SVMs are dumb…"

Math & Machine Learning

What's collaborative filtering and
how is it used in a machine learning context?

Math & Machine Learning

Provide two or more ways of determining how similar data points are.

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue *x* than to have the opinion on *x* of a person chosen randomly.

This is investigated mathematically by looking for correlations between users or between items. Then these correlations are used to weight the ratings of users on items that are most highly correlated.

Some recommendation systems use collaborative filtering.

Cosine similarity quantifies how much two multidimensional vectors point in the same direction on a scale from 1 (pointing in the same direction) to -1 (in the same line but pointing in opposite directions). Cosine similarity of 0 means the vectors are orthogonal. Cosine similarity is calculated from the dot product of the two vectors divided by the product of their magnitudes. Cosine similarity does not differentiate based on the magnitude of vectors, only where they are pointing.

Euclidean distance quantifies the straight line distance between two vectors using the root sum of the squared differences. Euclidean distance depends on both vector magnitudes and the directions the vectors are pointing.

Math & Machine Learning

What's the difference between the coefficient of determination $R^2$ and an adjusted $R^2$?
When might you use the adjusted value?

Math & Machine Learning

Do you think the ensemble average of 50 small decision trees will outperform a single deep decision tree?
Why or why not?

$R^2$ measures how close data are to a fitted regression line. Another way to think of it is the fraction of variance in the target variable that can be explained by the model. It is defined mathematically as 1 minus the sum of the squared residuals (SSR) divided by the total sum of the squares (SST), where SST is the sum, over all observations, of the squared differences of each observation from the overall mean.

$R^2 = 1 – SSR/SST$

$R^2$ will vary from 0 (the model explains none of the variance) to 1 (the model explains all the variance). $R^2$ will not decrease as more coefficients are added to a regression model, even if the coefficients are non-sensical. Adjusted $R^2$ modifies the $R^2$ value by penalizing it by the number of coefficients.

It's calculated from: Adjusted $R^2 = ((1 – n) R^2 – k) / (n – k – 1)$
where n is the number of data points and k is the number of coefficients.

Used adjusted $R^2$ when you wish to punish model complexity.

A single deep decision tree is likely to suffer from overfitting. It will exhibit very little bias on the training data, but likely show large variance on the test data.

The idea of using many small trees is an example of the approach of combining many weak learners into a strong learner. Each tree of the 50 trees will likely show a good degree of bias, but as the ensemble average is taken both bias and variance will be decrease.

So the model using multiple small trees will likely be better in practice.

Math & Machine Learning

For a regression model, why might the mean square error be a poor measure of model performance?
What would you suggest instead?

Math & Machine Learning

What is a major shortcoming of the Naive Bayes approach?
How might you address it?

In MSE, the residuals are squared. Therefore large outliers, which will result in large residuals (errors) and even larger squared residuals, may unduly affect the regression to minimize the effect of the outliers rather than the rest of the data.

An alternate approach would be to use the mean absolute error (MAE) which would prevent squaring an already possibly large value.

Naive Bayes assumes that all features in the dataset are uncorrelated and independent. In large datasets this is rarely the case.

A way to deal with correlation between features is to use Principal Component Analysis (PCA). PCA performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. These eigenvectors are less correlated than the original higher dimension feature vectors.

Math & Machine Learning

What are a couple of reasons to use an intercept term in a linear regression?

Math & Machine Learning

What assumptions are required for linear regression?

Associate the assumptions with the ability to:
1) Predict target $y$ from features $x$
2) Estimate the standard error of the coefficients
3) Predict an unbiased target $y$ from features $x$
4) Make probability statements, perform hypothesis testing involving slope and correlation, estimate confidence intervals

1) The intercept will gaurantee that the residuals have zero mean.

2) It guarantees the least squares slopes estimates (the coefficients) are unbiased.

Assumptions:
1) The data used in fitting the model is representative of the population.
2) The true underlying relation between *x* and *y* is linear.
3) Variance of the residuals is constant (homoscedastic, not heteroscedastic).
4) The residuals are independent (time series and spacial data can complicate this).
5) The residuals are normally distributed.

To predict y from x: assumptions 1, 2
To estimate standard error of the coefficients: assumptions 1, 2, 3
To get an unbiased estimation of y from x: 1, 2, 3, 4
To make probability statements, perform hypothesis testing involving slope and correlation, estimate confidence intervals: 1, 2, 3, 4, 5

Note:
Linear regression doesn't assume anything about the distributions of *x* and *y*, it only makes assumptions about the distribution of the residuals, and this is all that's needed for the statistical tests to be valid.

Math & Machine Learning

How might one find the local minimums or maximums of a function?
How do you know if they are mininums or maximums?

Math & Machine Learning

What is a major difference between linear and logistic regression?
Provide a scenario in which each would be used.

How would you interpret coefficients obtained from each?

Minimums or maximums in a function are found by finding where the first derivative (taken analytically or numerically) is equal to zero.

The second derivative is the derivative of the first derivative.
Where the first derivative is zero:

- if the second derivative is positive, the point is a local minimum.
- if the second derivative is negative, the point is a local maximum.

In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In logistic regression, the outcome (dependent variable) has only a limited number of possible values.

For instance, if X contains the area in square feet of houses, and *y* contains the corresponding sale price of those houses, you could use linear regression to predict selling price as a function of house size. However, it you wanted to predict whether or not a house would sell for more than $200k based on size, you would use logistic regression. The possible outputs are either Yes, the house will sell for more than $200K, or No, the house will not.

In linear regression, the interpretation of each of the coefficients (besides the intercept) is the difference in the predicted value *y* for each one-unit difference in the feature associated with that coefficient, if all other features remain constant.

In logistic regression, the coefficients (besides the intercept) can be interpreted in terms of an increase or decrease in the odds ratio, which is the ratio of the probability of something happening to the probability of something not happening. A one unit increase in a feature will either increase or decrease the odds ratio by $e^{\text{Coefficent\_for\_that\_feature}}$

Math & Machine Learning

What is multicollinearity in your data set and why is it a potential problem?

How do you detect it?

How might you remove it?

Math & Machine Learning

Describe how a decision tree works.

Additionally, provide some detail on how the feature to split on is determined at some point in the tree.

Collinearity occurs in a dataset when two or more features (a.k.a X variables) are highly correlated. These features provide redundant information. The issue with collinear featues is that the design matrix becomes singular and can't be inverted so the best fit coefficients can't be determined.

Collinearity can be detected from:
- Opposing signs for the coefficients of the affected variables, where it's expected that both would be positive or negative.
- The standard errors of the regression coefficients of the affected variables tend to be large.
- Large changes in the individual coefficients when a predictor variable is added or deleted.
- Rule of thumb: a variance inflation factor (VIF) > 5 indicates a multicollinearity problem, where:
  tolerance = $1 - R^2_j$ and VIF = 1 / tolerance
  $R^2_j$ is the coefficient of determination of a regression of feature j on all the other features.

Collinearity can be addessed by:
- Regularization (Ridge and Lasso)
- Principal component analysis (PCA)
- Engineering a feature that combines the affected features
- Simple dropping one of the features (worst, but viable, option)

1) Take the entire data set as input
2) Feature by feature, search for a split that maximizes the "separation" of the classes. This split will divide the data in two.
3) Apply the best split (the one that decreases impurity the most) to the input data. There are different ways to do this (see below).
4) Re-apply steps 1 – 3, where in each case the divided data is the new data set
5) Stop at a stopping criteria. For example, this could be a maximum depth, or minimum number of samples per leaf, or all the data has been perfectly classified (and the model is probably overfit!).
6) Optional – to decrease overfitting, you could go back and "prune" the trees to some maximum depth.

Algorithms for constructing decision trees usually work top-down, by choosing a feature at each step that "best" splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets after the split. These metrics are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split. Algorithms for determining this split are:
Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.
Information gain (entropy) is based on the concept of entropy from information theory, where the goal is to mimimize heterogeneity in the resulting subsets.
Variance reduction is often employed in cases where the target is continuous (regression tree), meaning that use of other metrics would first require discretization before being applied. The variance reduction is defined as the total reduction of the variance of the target variable due to the split.

Math & Machine Learning

What is the curse of dimensionality?
How does it affect distance and similarity measures?

Math & Machine Learning

What do you think about the idea of injecting noise in your data set to test the sensitivity of your models?

* The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high dimensional spaces.

* Common theme: when number of dimensions increases, the volume of the space increases so fast that the available data becomes sparse

* Issue with any method that requires statistical significance: the amount of data needed to support the result grows exponentially with the dimensionality.

* Everything becomes far and difficult to organize, so everything is nearly equally far and dissimilar.

It’s not a bad idea – it should help avoid overfitting, so you could use this to increase the generality of the model. Regularization and ensemble methods address this, too.