Machine Learning

Machine learning is a field overlapping with statistics and computer science and strongly associated with *big data*. One important point, however, is that machine learning techniques often provide their largest advantages in medium-sized data sets (say, a few hundred to a few thousand cases); in truly large data sets, different techniques converge. Machine learning grew out of artificial intelligence research. It has been characterized as methods that allow a "machine" (typically an algorithm) to optimize some error function (a *loss* or *objective function*) in order to *learn* to do a specific task without being explicitly programmed to do that task (e.g., making predictions, classifying cases). The keys to the machine learning approach involve the discipline's approach to the data. Machine learning techniques are usually evaluated using cross-validation. The algorithm *learns* on a *training set* of data, and its performance is evaluated on different data, the *test set*.

Machine learning techniques come in two major categories: *supervised* and *unsupervised*. Supervised techniques include two classes of problems: *regression problems*, or quantitative prediction, and *classification problems*, or qualitative prediction. Supervised techniques expect that "correct answers" exist in the training data. Ordinary least squares regression is an example of supervised learning. The outcome variable is quantitative (is indexed by the real numbers) and the technique produces predicted values for the outcome variable that are optimal relative to the squared error loss function. Any problem with these features –predicting known quantitative data from other data – is a regression problem. Logistic regression is an example of a *classification problem*: the known outcome is nominal. Cluster analysis is an example of an *unsupervised learning* technique: there are no "correct" answers provided in the dataset. For example, cluster analysis methods attempt to find groups in the training data, but there is no explicit labeling of the correct grouping included. Exploratory factor analysis is another example of an unsupervised technique.

A difficulty with learning algorithms is the *curse of dimensionality*. Think of a dataset as points in space; in low dimension, say one predictor and one response, relatively few data points can effectively fill the space. Fairly little data can fully characterize the shape of the response function of the outcome on the predictor. As the dimension of the problem increases (e.g., including more predictors), more points are needed to fill the space. In sufficiently high dimension, the points needed to fill the space become too numerous, and the problem, intractable. Tools like linear regression can still work in these situations, because assumptions, like additivity and linearity of the partial responses of the outcome on each predictor, help to simplify the problem. The common learning algorithms described next can help answer a variety of problems in traditional data analysis.

Smoothing and Nearest Neighbor Methods

Smoothing techniques are important for machine learning: imagine a rough piece of wood that needs to be sanded down in order to apply varnish. A raw dataset is like wood before sanding, individual data points stick out like rough edges. A fitted model is *smoothed*-"sanded" like the prepared wood, the rough edges sanded away mathematically, leaving only the line, curve, or other shape represented by the model. In a regression problem, *k*-nearest neighbor methods (*k*-NN) are *smoothers* that create *neighborhoods* of *k* data points that sweep along the *X*-axis, essentially tracing out the plot of the local average. If *k* = 1, then the size of the neighborhood is an individual datum; the k-NN smoother does no smoothing. That is, a *k*=1 nearest neighbors model simply reproduces the data. As *k* increases, the neighborhood gets bigger, until *k* = *N*, and we just have *Y* = E[Y|X], the standard regression function. This approach can also be used for classification and clustering.

Other smoothing techniques use *kernel methods*. Kernel methods are conceptually similar to k-NN, but instead of defining a neighborhood size, a *kernel function* is used. The kernel function is essentially a weighting scheme that determines how much influence a data point has in calculating the local average at the current focal point. Another approach is *smoothing splines*, which are piecewise functions (often linear or cubic) lined up at various breakpoints, called *knots*, along the *X*-axis. Splines (or other smoothers) can be used to construct additive models, which are generalizations of multiple regression models, with the restriction to linear partial response functions relaxed. Since they retain the additivity assumption (that the partial responses sum up to produce the output), additive models reduce to iterative fitting in two dimensions (the outcome on an individual predictor variable), so they are not subject to the curse of dimensionality. Other smoothers, like kernel methods, can fail by smoothing over the whole space.

Neural networks

Artificial neural networks (ANNs) are essentially a form of non-parametric regression analysis. Most neural networks, used in this fashion, take the form of *multilayer perceptrons*, such that there are two "visible" layers – one for inputs (*predictor* or X-side variables) and one for outputs (*criteria* or Y-side variables) – and at least one hidden layer. Each of these layers is made up of *nodes*, the simple artificial neurons, with one node corresponding to each input variable and one node for each outcome. Input nodes are connected to hidden nodes in a way that can represent essentially arbitrary transformations of the inputs. The nodes in the hidden layer are wired to the output to produce a *response*. Think of a polynomial regression of the form $Y = B_1X_1 + B_2X_2 + B_3X_1^2 + B_4X_2^2 + B_5XY$. The "inputs" to this model are the variables X_1 and X_2. Their squares and their interaction are higher-order terms constructed as predictors for the regression model. Neural networks work analogously, except the transformations are arbitrary and not explicitly represented. Instead, the transformations are implicit in the pattern and strength of connections between the nodes in the input layer and those in the hidden layer (a similar approach in statistics is called *projection pursuit regression*). These transformations allow the ANN to capture non-specific nonlinearities in the response surface of the output on the field of inputs.

Support vector machines

Another method strongly associated with machine learning is support vector machines (SVMs). The inputs are represented as points in a high dimensional space. Traditionally, SVMs were used for classification problems, but have been adapted for regression problems. In the classification problem setup, the SVM finds the hyperplane that optimally separates the two classes of data points.

Tree methods

Tree methods divide the data into partitions based on the variable (or set of variables) that predicts best at the current level of partitioning. This partitioning is represented in a tree, and each terminal node or *leaf node* contains only cases with the same predicted value of the outcome. This approach can be used for both metric variables (*regression tree*) or for nominal outcomes (*classification tree*). The overall approach is known as *classification and regression trees* (CARTs). These methods can substantially improve on predictive results over traditional methods, especially in the presence of nonlinearities or interactions.

Ensembles

"Ensemble" methods blend together the results of various models and algorithms. This is helpful, since a source of uncertainty in statistical models that is *uncertainty due to model choice.* That is, a different statistical model than the one the researcher has specified might better represent a phenomenon: additional control variables should be included, an interaction was not modeled, the form of the response function is nonlinear, etc. Ensemble methods help by fitting a large variety of models and using predictions that take them all into account, say by averaging predictions over models (this can work well when each individual model is itself poor). This can be done using model averaging or more elaborate methods. One approach is *random forests*. Tree methods can be very sensitive to outlying observations and their structure can be affected strongly by the predictors available. One option is to grow a variety of trees fitted to random subsets (both cases and variables) of the data. This forest often predicts better than any single tree. Another option is *bagging* (bootstrap aggregation), which only samples from cases, using all variables.

Cross-validation and regularization

One of the defining aspects of machine learning is its emphasis on cross-validation. While cross-validation is used in the organizational sciences, it is essentially a *sine qua non* in machine learning. A technique is generally not considered to "work" unless it produces good results in new data, not just the data that it was fitted in originally. That is, a machine learns in a training set, where the fit of the model is assessed by some sort of error measure (e.g., squared residuals), and then validated in a test set, using the same error measure. If the predictive residuals in the test set are too large, the technique is probably capitalizing on idiosyncrasies in the training data.

Such "capitalizing on chance" – or over-fitting – is one of the primary motivations for *regularization*, also referred to as *shrinkage*, methods. These methods penalize the loss function that the statistical method is designed to optimize. For example, *ridge regression* works by minimizing the least squares error subject to constraints on the size of the sum of the squared regression coefficients. Similarly, the *least absolute shrinkage and selection operator* (LASSO) minimizes the least squares error subject to a constraint on the sum of the *absolute values* of the regression coefficients. These penalties "shrink" regression coefficients, relative to the unconstrained least squares estimates; the coefficients produced are smaller, on average, than those produced by traditional methods. This helps prevent over-fitting. Regularized estimators usually cross-validate better than the unconstrained estimator.

Specifically, they produce smaller mean-square residuals in test set data than do unconstrained coefficients. This helps ensure greater transportability of results between research settings.

Other techniques

Genetic algorithms have proven useful in reducing the scale length without sacrificing construct coverage. Genetic algorithms represent their solutions as a binary sequence, called *chromosomes*, which are distributed in a population (around 100 to 200 chromosomes). In each generation, an optimization function determines the *fitness* of the chromosomes, and the fittest (around 20%) are retained; random mutations are introduced and the retained chromosomes recombine with each other. This process is repeated for around 100 generations, and the fittest chromosome of the last generation is taken as the answer. This rarely produces the *most optimal* solution, but usually produces good solutions.

In addition, like regularization methods, standard statistical models can often be improved by changing their objective functions. For instance, robust regression uses the least squares error function for points "near" the current estimated regression function and down-weights more distal points.

Implications

Smoothers, ANNs, SVMs, and CARTs are especially appropriate when theory is only strong enough to specify that a relationship is monotonic but necessarily linear. ANNs and SVMs can be "black box" solutions whose internal workings can be difficult to understand. Tree methods and additive models are comparatively easily interpretable. Ensemble methods and robust methods can help to specify and control uncertainty in results. Cross-validation and regularization can help to improve stability of results across research settings. In sum, machine-learning techniques can diversify organizational researchers' statistical methods.

Seth M. Spain, Binghamton University

Cross-references: Regression techniques, big data techniques/analytics and IO Psychology, big data techniques, factor analysis

Bibliography

Berk, R.A. (2009). *Statistical learning from a regression perspective.* New York, NY: Springer.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science, 16,* 199 –215. DOI: 10.1214/ss/1009213725

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd Ed.). New York, NY: Springer.

Hindman, M. (2015). Building better models: Prediction, replication, and machine learning in the social sciences. *The Annals of the American Academy of Political and Social Science, 659*, 48 -62. DOI: 10.1177/0002716215570279

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in R.* New York, NY: Springer.

Yarkoni, T. (2010). The abbreviation of personality, or how to measure 200 personality scales with 200 items. *Journal of Research in Personality, 44,* 180 – 198. DOI: 10.1016/j.jrp.2010.01.002