LECTURE 14: Data-mining, Neural Networks and the OVER- FITTING problem

HBR.ORG

Spotlight

Big Data: The Management Revolution 60
by Andrew McAfee and Erik Brynjolfsson

Data Scientist: The Sexiest Job of the 21st Century 70
by Thomas H. Davenport and D.J. Patil

Making Advanced Analytics Work for You 78
by Dominic Barton and David Court

Big Data

Businesses are collecting more data than they know what to do with. To turn all this information into competitive gold, they'll need new skills and a new management style.

Data, data everywhere

He Who Rules The Data, Rules The World: A Brief History Of Data Governance https://www.forbes.com/sites/ciocentral/2017/11/16/he-who-rules-the-data-rules-the-world-a-brief-history-of-data-governance/#7dc5872639b5

KDD – Knowledge Discovery in Data Bases – uses large data bases to predict outcomes with algorithms that search for patterns in data sets – millions of customers, Census respondents, minute by minute price changes, information on transactions in real time or maybe huge numbers of observations on one person. Since 1997 KDD-Cup Competition announced a problem with data, groups enter the competition, using different algorithms:

2018 competition was: KDD Cup of Fresh Air, solicits machine learning solutions to accurately forecast air quality indices (AQIs) of the future 48 hours. Accurate predictions of AQIs can bring enormous value to governments, enterprises, and the general public - and help them make informed decisions. Participants were asked to forecast the AQIs of Beijing, China and London, UK. Over 4,000 teams from 49 countries participated in the competition, and made over 20,000 submissions. The total prize of $36,500 was awarded to the top 10 teams in the regular track and top 3 teams in each of two special tracks.

KDD Cup 2018 Winners: Main Prize; First prize ($10,000): Haoran Jiang and Binli Luo from Central South University, Jindong Han, Juan Liu, and Qianqian Zhang from Beijing University of Posts and Telecommunications
Second prize ($5,000): Zhipeng Luo.f Microsoft, Jianqiang Huang Peking University, and Ke Hu from Alibaba
Third Prize ($3,000): Jie Zhou from East China Normal University, Hengxing Cai from Sun Yat-Sen University and Cortex Labs, and Xiaozhou Liu from Sun Yat-Sen University

2009 competition was: Customer Relationship Management (CRM) … using large marketing databases from French Telecom Orange to predict propensity of customers to switch provider, buy new products or services, or buy upgrades or add-ons proposed to them to make the sale more profitable betterthan in-house system. Winning Team: IBM T.J. Watson Research Center: Members: Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Prem Melville, Yan Liu, Jianying Hu, Moninder Singh; IBM China Research Laboratory: Dong Wang, Jing Xiao, Wei Xiong Shang, Yan Feng Zhu:

1) Lots of Work pre-processing and feature construction: **We normalized the numerical variables by range,** keeping the sparsity... coded categorical variables into at most 11 binary columns. For each categorical variable, we generated a binary feature for ten most common values, encoding whether the instance had this value or not.... We replaced missing values by mean for numerical attributes, and coded them as a separate value for discrete attributes. *Base classifier:* Decision tree, stub, or Random Forest, Linear classifier, Non-linear kernel method. We used all the models as candidates for inclusion in the final ensemble.

The largest advantage was training a large variety of classifiers. Different classifiers were best on the three different problems. Logistic regression was better on churn, random forests were the winner on appetency **Ensemble Selection** significantly boosted the performance over using the best single model ... (recoding) some of the numerical variables there were not linearly correlated with the target,led to significant gains. LOTS OF ENGINEERING here.

Cup2017 Alibaba Cloud, the cloud computing arm of Alibaba Group organized the KDD Cup titled "Highway Tollgates Traffic Flow Prediction" to empower traffic management authorities with data-driven preemptive measures and to pave the way towards holistic and realistic solution to traffic bottleneck (and) provide real-time traffic prediction and recommendations on travel routes in China. A total of 3547 teams from around the world participated . The winners are, Ke Hu, Microsoft(China) Co., Ltd. (Team Leader) Huan Chen, Beihang University Pan Huang, Microsoft(China) Co., Ltd. Peng Yan, Meituan-Dianping Co., Ltd. Presentation (pdf) (http://www.kdd.org/kdd2017/files/Task1_1stPlace.pdf);http://www.kdd.org/kdd2017/files/Task2_1stPlace.pdf)

What will data mining do for you? (www.kdnuggets.com/ as portal; Andrew Moore of CMU, Googletutorials http://www.autonlab.org/tutorials/) . In huge data the problem is MODEL SELECTION NOT ESTIMATION. Approach pushed by business is to predict consumer behavior not test models. Statistics is also moving away from the do your "tests" as if you had a preset model.

Data-mining KDD methods are:

1. Multivariate statistics. Linear regression (often with nonlinear transforms of variables) describes data.

2.Classification into groups. Algorithm finds rules to classify into groups that you or machine specifies according to some optimizing criterion. Clustering algorithms and K-nearest neighbor algorithms – **Tree models**, which use hierarchical path breaks to classify/predict, using splits of different types: more use of local information.

3.Nonlinear likelihood-based models assume that outcomes depend on some combination of nonlinear "basis functions" (sigmoids, splines, polynomials) of input variables. **Neural nets fit here.** You say: search for the best model using these 100 variables and the algorithm reports results to you.

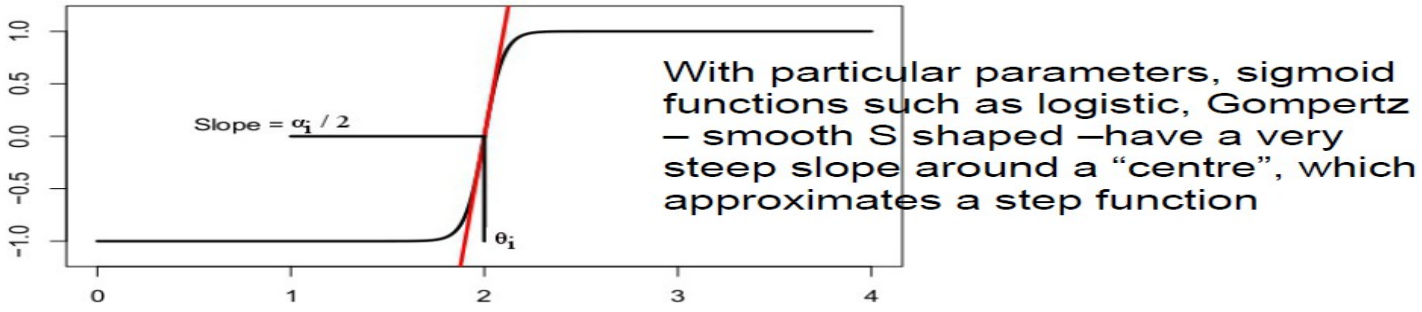
KEY: No single algorithm works best … must determine **which algorithm works best on which problem.** Some tools stress understanding what goes on. Others are "black box" of interactions that you don't understand.

Global search via Neural Nets

Neural net algorithm uses all the data to derive model. Mimicking biological learning, it creates "artificial neurons" that transform inputs into outputs in nonlinear ways using "squasher functions". NNs developed by Rosenblatt, denounced in 1969 by Minsky & Papert, resuscitated in the 1980s with Hopfield and McLelland and Rumelhart using backprop algorithm and McCulloch-Pitts learning rules now triumphing as "DEEP LEARNING".

You obtain data, divide data into **training set and test set**. The NN "discovers" non-linearities in the training set to predict. The test set makes sure the model does not "over-fit" the data.

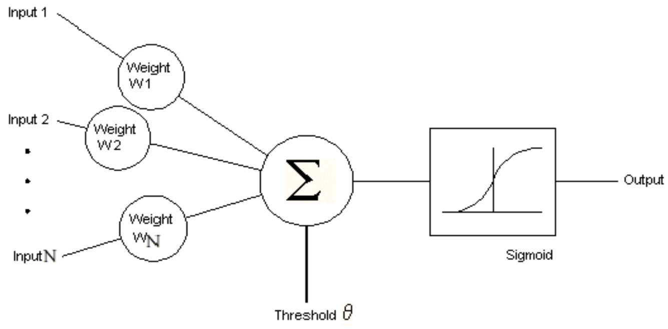
Single-layer Feed-forward Perceptron takes a weighted average of input variables and uses the equation Oi = g(∑wij Xj) with g being **sigmoid-type** (logistic) function that approximates the 0/1 step function. Trains model to recognize patterns – for instance, if someone is likely to be criminal or not. By relating outcome to many terms, errors of measurement in particular variables should have little effect on prediction –> robust mode of prediction.

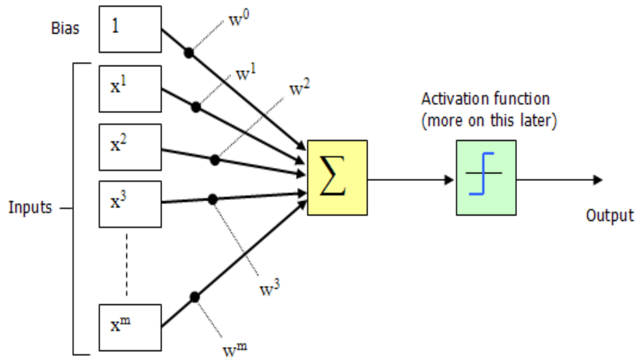


With particular parameters, sigmoid functions such as logistic, Gompertz — smooth S shaped —have a very steep slope around a "centre", which approximates a step function

To estimate neural net, randomly choose weights w. Then change weight vector in direction that improves prediction: **Δw_{ik} = a (Yi-Oi) Xk, where Yi is the actual value; and Oi is the predicted output.**

If Yi > Oi, short of the actual value so you want to increase prediction. You do this by raising the weight on predicting the ith term Oi. If Xk is 1, you will raise the parameters proportionately on all units that "fire" in the prediction equation. Doing this one term at a time might yield cycles, but it doesn't.





Example: If Oi = g(∑wij Xj > 0) then the prediction is 1.

Observation	Y	X	Z
1	1	1	1
2	0	1	0
3	1	0	1

What are the best weights for fitting these data? Weight of 1 on Z and 0 on X.
But say you start with ½ on X and Z, giving Y = ½ X + ½ Z. Obvtnn 1: ½ weight works. Fails on observation 2.

LOWER the weight on X which is "on" in the second observation. Let a = ½ in the model **Δw_{ik} = a (Yi-Oi) Xk.** Then lower it to ¼. Go to observation 3 and must raise weight on Z ... to 3/4ths. After enough iterations the weights will be close to weight of 1 on Z and 0 on X. If the model included a constant "bias" of -1/4 , we would get correct predictions in one more iteration.

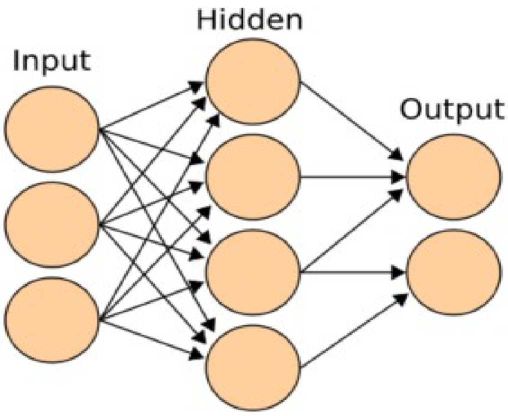
BUT MODEL CANNOT DEAL WITH XOR problem – determining if one of two conditions is present:

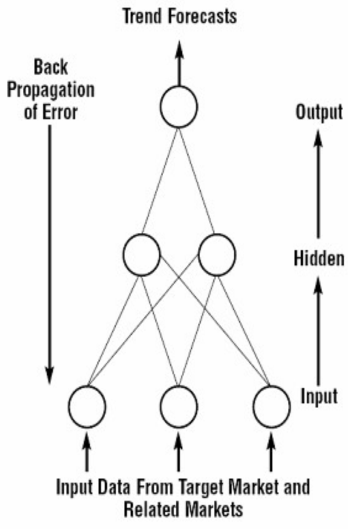
X1	X2	OUTPUT	WHAT YOU NEED
0	0	0	w_1 X1 + w_2 X2 <0 (or some other specified c)
0	1	1	w_1 X1 + w_2 X2 >0
1	0	1	w_1 X1 + w_2 X2 >0
1	1	0	w_1 X1 + w_2 X2 <0

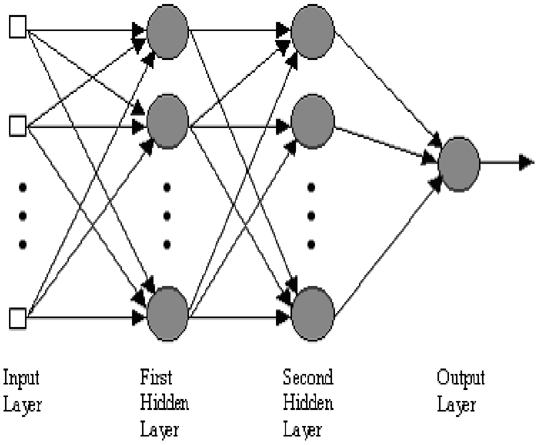
Cannot get weights to make this distinction, bcs to predict 1 on the second observation need $w_2 > 0$ and to predict 1 on the third observation need $w_1 > 0$. But then $w_2 + w_1 > 0$, so cannot predict the fourth observation.

To the rescue Multi-layer Model, with hidden layers

Use 3 layers. Each input contributes to each hidden layer, which in turn contribute to each final outcome. The hidden layers are latent variables that give flexibility to fit the data: # of hidden "neurons" can >=# inputs. Also can have more than one hidden layer.




Trend Forecasts
Back Propagation of Error
Output
Hidden
Input
Input Data From Target Market and Related Markets



So, how do we find weights that link inputs to hidden layer and hidden layer to outputs? There are many weights in these models. 3 x 4 + 5 = 17 weights in the first model, 8 in the second model; 21 in the 3rd model. Each input affects the output through all of the hidden layer nodes. So the effect of a given input is the sum of its effects on each hidden layer multiplied by the effect of each hidden layer on the outcome.

Nonlinearity comes from a function that takes 0/1 inputs (on/off) and sends output of 1 if weighted sum of the inputs > critical value; otherwise the node sends output of 0; **logistic $Y = 1/[1+\exp(-K \sum wx)]$.** Logistic has values between 0 and 1, going to 1 as x--> ∞, going to 0 and → 0 as x--> -∞. Logisic has S growth shape; is steeper the larger is K; Can also use tanh(x) = (exp x - exp -x)/ (exp x + exp-x), with values between -1 and 1.

Finding Weights

To find 17 or 8 or 21 weights, neural nets use iterative “Learning rules”. Here is a simple learning rule that will get the neural net to tell you if both A and B are present – the logical “and”: Start with random weight of 0.5: **0.5 X + 0.5 Z - 0.4 >0** is your initial rule for declaring yes, both A and B are present. The weights on X and Z are 0.5 and the “bias” is -0.4. Then change weights observation by observation as more data appears:

				Transformed		
obs	X	Z	output	by squasher	desired output	Result
1	0	0	-.4	0	0	good
2	1	0	.1	1	0	bad
3	0	1	.1	1	0	bad
4	1	1	.4	1	1	good

Since prediction of 1 in line 2 is wrong, change weights to predict 0 instead of 1, say by some proportion of the error term. Change the weight on X to 0.4. This creates equation .4 X + .5 Z - 0.4 = 0, which would bring the 2d observation into line. Now go to third observation. The change in the weight on X does you no good since X has no effect on the third observation. Change the weight on Z to 0.4. This works. The fourth observation still gets >0. And the first observation <0 with new weights. Adjusting weights on A and B down by 20% worked

Widrow-Hoff (delta) learning rule is: **ΔWeight = a (TRUE - OUTPUT) X,** where a = learning rate; X is 1/0

Extra layers solves the nonlinear exclusive OR, where X1 OR X2 are present you get a 1 but 0 when neither are present. To solve the XOR, need weights so that when X1 =1 and X2 = 0 the neural net gives output 1 and when X2 =1 and X1 = 0 it gives output 1. Get 0 otherwise.

X1	X2	HL1	HL2	OUTPUT
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

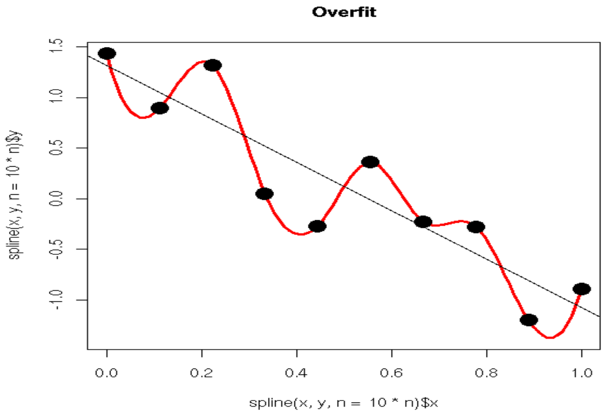
Set HL1: = w_{11} X1 + w_{21} X2 - 0.5 = X1 + X2- 0.5. HL1 tells you if either is on since if X1 or X2 =1, HL1>0
Set HL2: = ww_{12}X1 + w_{22}X2 - 1.5 = X1 + X2- 1.5. HL2 tells you if both are on.
Then let Output = HL1 - 2HL2. So O > 0 if HL1=1 and HL2 =0 or if HL1=0 and HL2 =1
But O<=0 if HL1 and HL2 =0 or if HL1 and HL2 = 1. This does it. Note these are NOT the only set of weights that will accomplish this: many others will work. Do not need two hidden layers. One hidden layer will do it.

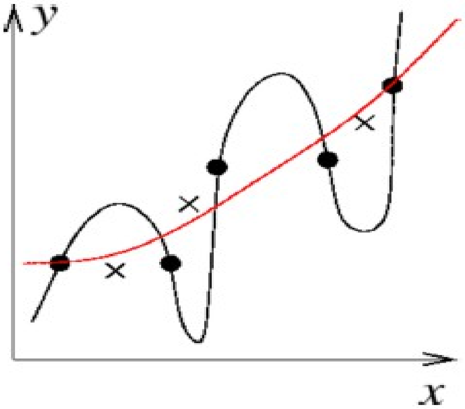
NB: A **hidden layer helps only if node functions are non-linear**. Linear relations between the inputs and the hidden layer gives linear averages, which replace the hidden layer:

HL1 = a1 X1 + b1 X2
HL2 = a2 X1 + b2 X2 –> O = c1HL1 + dHL2 = (ca1+ da2) X1 + (cb1+ db2) X2

That many different vectors of weights can fit the same data creates problems. Say you set up some weights to predict perfectly 5 data points. Will the weights fit a 6th data point? Likely not, because you may have overfit the broader data set. By changing weights to fit observed data a data-mining model that uses lots of parameters/ weights to replicate existing data may generalize poorly to new data.

There is no general theory about how many layers or nodes are best, any more than for the standard regression model where you do not know how many variables in what form to enter an equation
Data-mining deals with the over-fitting problem by dividing data into a training data set on which you fit the model and a test data set to rule out excessive nodes/parameters or weights. This is what overfitting looks like:





There is a theorem just one hidden layer and appropriate learning rule can approximate or fit ANY function (Hornik, Stincombe, White) but if you overfit, your model will work on training data but fail with test data.

The actual procedure for a neural net model:

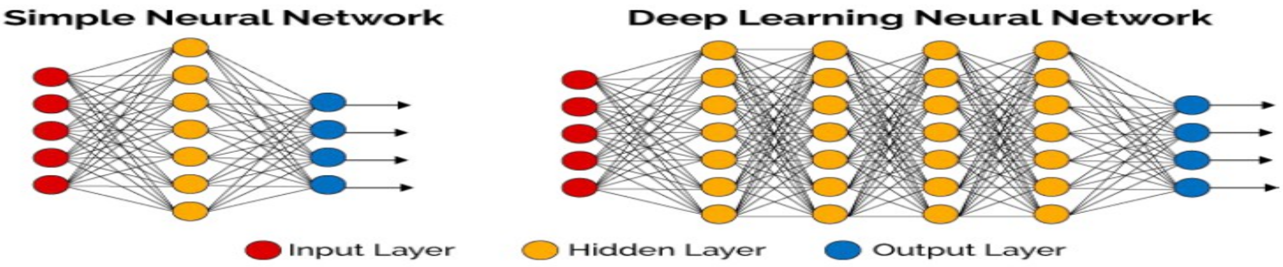
1. Get some data; normalise the data in some fashion
2 Decide on which type of model to use (or use many models).
3 Choose NN "architecture": how many nodes and layers, learning rates, etc.
4. Train your model by adjusting weights to get the best prediction
5.Assess the model on the training or validation set.

If you have 3 NN models and use the second data set to decide which is best, you will need a third one to estimate potential outcomes on new data. Can also add costs to penalize model for too much complexity. Say you have an NN with 6 variables, 3 hidden layers and 1 output --> 21weignts to estimate. Comparable regression model would have 21 regressors, which allows you to square every term; and have other non-linearities.

"Deep Learning"

"Deep learning allows computational models that are composed of **multiple processing layers** to learn representations of data with multiple levels of abstraction. These methods have dramatically improved ...speech recognition, visual object recognition, object detection … drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the **backpropagation algorithm** to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep **convolutional nets** have brought about breakthroughs in processing images, video, speech and audio, whereas **recurrent nets** have shone light on sequential data such as text and speech." (LeCun, Bengio & Hinton, Deep Learning Review, Nature, Vol 521 May 28, 2015)

An image, for example, comes in the form of an array of pixel values, and the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations and locations in the image. The second layer detects motifs by spotting particular arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts ... **these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure. ...** (with) ... **hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine.** "Promising results in natural language understanding, particularly topic classification, sentiment analysis, question answering and language translation."



See http://deeplearning.net/

What is backprop? A gradient descent algorithm that calculates how much loss function changes with changes in weights of network and then updates the weights to reduce the loss function It uses chain rule from calculus to go from error in network "backwards".

Take error, compute how much changes in weights at given layer affect error and alter weights to make the error smaller … use logistic or other differentiable functions. The process begins by modifying the weights at the output layer, and proceeds backwards on the hidden layers one by one until it stops at the input layer. The proceeding backwards gives the name Backward Propagation.

Neural networks designed for vision/face recognition problems exploit localization, with neurons near each other linked together so that the network first create good representations of small parts of the input, then assemble representations of larger areas from them. And they use similar weights across a layer that reduces the number of weights to be learned.

Data-Mining Meets the Law

Supreme Court Strikes Down Ban on Data-Mining (Emily P. Walker, MedPage Today, June 23, 2011

Data on which doctors are prescribing which drugs is speech that is protected by the First Amendment, and pharmaceutical companies have every right to buy that information and use is to target their marketing efforts, the Supreme Court has ruled. The nation's high court handed down a verdict in the Sorrell v. IMS Health case, striking down by a 6-3 vote a 2007 Vermont law that that bans the sale and use of prescriber-identifiable information for marketing or promoting a drug, including drug detailing -- unless a physician specifically gives his or her permission to use the information.

N.S.A. Collection of Bulk Call Data Is Ruled Illegal NY TIMES MAY 7, 2015 A federal appeals court in New York ruled on Thursday that the once secret National Security Agency program that is systematically collecting Americans' phone records in bulk is illegal.

Big pharma, big data: why drugmakers want your health records March 1, 2018 LONDON (Reuters) - Drugmakers are racing to scoop up patient health records and strike deals with technology companies as big data analytics start to unlock a trove of information about how medicines perform in the real world. Studying such real-world evidence offers manufacturers a powerful tool to prove the value of their drugs - something Roche aims to leverage, for example, with last month's $2 billion purchase of Flatiron Health. Real-world evidence involves collecting data outside traditional randomized clinical trials, the current gold standard for judging medicines, and interest in the field is ballooning. Half of the world's 1,800 clinical studies involving real-world or real-life data since 2006 have been started in the last three years, with a record 300 last year, according to a Reuters analysis of the U.S. National Institutes of Health's clinicaltrials.gov website.

Economist prediction: Profit-maximizing firms will data mine Big Data about you to offer products/prices that will "discriminating monopolist" smart, charging different prices for each person (as long as p> cost) to gain higher profits than normal monopolist or to offer different wages to workers (w>reservation wage). Both processes will produce competitive Q's with consumer/worker inframarginal consumer surplus gained by firm. Discriminating monopoly/monopsony lead to efficient outcomes with more unequal income distribution.

Law Firms Enter the Golden Age of Data Mining

By **Doug Stansfield, LexisNexis** | December 11, 2019 at 02:29 PM

Law firms generate and store incomprehensible amounts of data. Most, if not all, of that data has been digitized and many firms that recognize the untapped value of their data have begun to leverage sophisticated technologies to mine it for reusable work product and valuable insights.

As a matter of practice, law firms generate and store incomprehensible amounts of data. Most, if not all, of that data has been digitized and many firms that recognize the untapped value of their data have begun to leverage sophisticated technologies to mine it for reusable work product and valuable insights. While most large law firms with 50 or more attorneys are currently mining data, not all of them are doing it well. Data mining is a relatively new practice in the legal space and the data profiles of firms are highly variable from one organization to another, so identifying the right tools and prioritizing initiatives can be challenging.