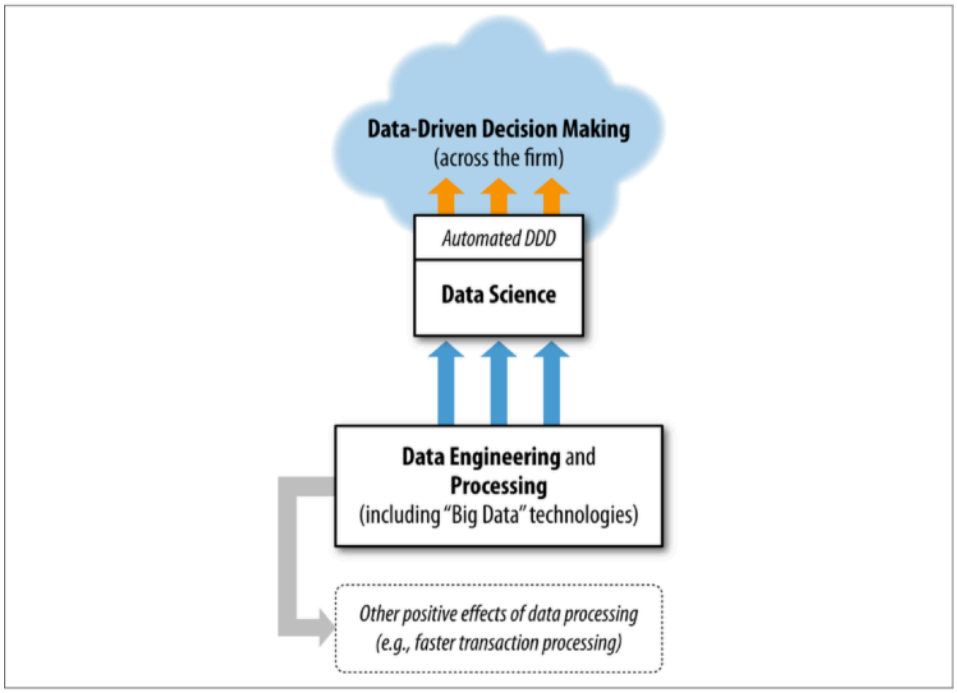
Sunday, October 14, 2018

Security Analytics



Chap1:

1. Data science: a set of fundamental principles that guide the extraction of knowledge from data.
2. Data mining: extraction of knowledge from data, via technologies that incorporate these principles
3. Data-driven decision-making: refers to the practice of basing decisions on the analysis of data, rather than purely on intuition
4. Two types of decisions: decisions for which "discoveries" need to be made within data, and (2) decisions that repeat, especially at massive scale, and so decision-making can benefit from even small increases in decision-making accuracy based on data analysis
5. Increasingly, business decisions are being made automatically by computer systems.
6. Big data: essentially means datasets that are too large for traditional data processing systems, and therefore require new processing technologies

7. Occasionally, big data technologies are actually used for *implementing* data mining techniques. However, much more often the well-known big data technologies are used for data processing *in support of* the data mining techniques and other data science activities

8. One of the fundamental principles of data science: data, and the capability to extract useful knowledge from data, should be regarded as key strategic assets

Chap 2:

9. Classification and regression methods:

   - *Classification* and class *probability estimation* attempt to predict, for each individual in a population, which of a (small) set of classes this individual belongs to. Usually the classes are mutually exclusive. An example classification question would be: "Among all the customers of MegaTelCo, which are likely to respond to a given offer?" In this example the two classes could be called will respond and will not respond. For a classification task, a data mining procedure produces a model that, given a new individual, determines which class that individual belongs to. A closely related task is *scoring* or class *probability estimation*. A scoring model applied to an individual produces, instead of a class prediction, a score representing the probability (or some other quantification of likelihood) that that individual belongs to each class. Classification and scoring are very closely related; as we shall see, a model that can do one can usually be modified to do the other.

   - *Regression* ("value estimation") attempts to estimate or predict, for each individual, the numerical value of some variable for that individual. An example regression question would be: "How much will a given customer use the service?" The property (variable) to be predicted here is *service usage*, and a model could be generated by looking at other, similar individuals in the population and their historical usage. A regression procedure produces a model that, given an individual, estimates the value of the particular variable specific to that individual. Regression is related to classification, but the two are different. Informally, classification predicts *whether* something will happen, whereas regression predicts *how much* something will happen. The difference will become clearer as the book progresses.

   - *Similarity matching* attempts to *identify* similar individuals based on data known about them. Similarity matching can be used directly to find similar entities. For example, IBM is interested in finding companies similar to their best business customers, in order to focus their sales force on the best opportunities. They use

similarity matching based on “firmographic” data describing characteristics of the companies. Similarity matching is the basis for one of the most popular methods for making product recommendations (finding people who are similar to you in terms of the products they have liked or have purchased). Similarity measures underlie certain solutions to other data mining tasks, such as classification, regression, and clustering.

- *Clustering* attempts to *group* individuals in a population together by their similarity, but not driven by any specific purpose. An example clustering question would be: “Do our customers form natural groups or segments?” Clustering is useful in preliminary domain exploration to see which natural groups exist because these groups in turn may suggest other data mining tasks or approaches. Clustering also is used as input to decision-making processes focusing on questions such as: *What products should we offer or develop*? *How should our customer care teams (or sales teams) be structured*?

- *Co-occurrence grouping* (also known as frequent itemset mining, association rule discovery, and market-basket analysis) attempts to find *associations* between entities based on transactions involving them. An example co-occurrence question would be: *What items are commonly purchased together*? While clustering looks at similarity between objects based on the objects’ attributes, co-occurrence grouping considers similarity of objects based on their appearing together in transactions. For example, analyzing purchase records from a supermarket may uncover that ground meat is purchased together with hot sauce much more frequently than we might expect. Deciding how to act upon this discovery might require some creativity, but it could suggest a special promotion, product display, or combination offer. Co-occurrence of products in purchases is a common type of grouping known as market-basket analysis. Some *recommendation* systems also perform a type of affinity grouping by finding, for example, pairs of books that are purchased frequently by the same people (“people who bought X also bought Y”). The result of co-occurrence grouping is a description of items that occur together. These descriptions usually include statistics on the frequency of the co-occurrence and an estimate of how surprising it is.

- *Profiling* (also known as behavior description) attempts to characterize the typical behavior of an individual, group, or population. An example profiling question would be: “What is the typical cell phone usage of this customer segment?” Behavior may not have a simple description; profiling cell phone usage might require a complex description of night and weekend airtime averages, international usage, roam- ing charges, text minutes, and so on. Behavior can be described generally over an entire population, or down to the

level of small groups or even individuals. Profiling is often used to establish behavioral norms for anomaly detection applications such as fraud detection and monitoring for intrusions to computer systems (such as someone breaking into your iTunes account). For example, if we know what kind of purchases a person typically makes on a credit card, we can determine whether a new charge on the card fits that profile or not. We can use the degree of mismatch as a suspicion score and issue an alarm if it is too high.

- *Link prediction* attempts to predict connections between data items, usually by suggesting that a link should exist, and possibly also estimating the strength of the link. Link prediction is common in social networking systems: "Since you and Karen share 10 friends, maybe you'd like to be Karen's friend?" Link prediction can also estimate the strength of a link. For example, for recommending movies to customers one can think of a graph between customers and the movies they've watched or rated. Within the graph, we search for links that do *not* exist between customers and movies, but that we predict should exist and should be strong. These links form the basis for recommendations
- *Data reduction* attempts to take a large set of data and replace it with a smaller set of data that contains much of the important information in the larger set. The smaller dataset may be easier to deal with or to process. Moreover, the smaller dataset may better reveal the information. For example, a massive dataset on consumer movie-viewing preferences may be reduced to a much smaller dataset revealing the consumer taste preferences that are latent in the viewing data (for ex- ample, viewer genre preferences). Data reduction usually involves loss of information. What is important is the trade-off for improved insight.
- *Causal modeling* attempts to help us understand what events or actions actually *influence* others. For example, consider that we use predictive modeling to target advertisements to consumers, and we observe that indeed the targeted consumers purchase at a higher rate subsequent to having been targeted. Was this because the advertisements influenced the consumers to purchase? Or did the predictive mod- els simply do a good job of identifying those consumers who would have purchased anyway? Techniques for causal modeling include those involving a substantial investment in data, such as randomized controlled experiments (e.g., so-called "A/B tests"), as well as sophisticated methods for drawing causal conclusions from observational data. Both experimental and observational methods for causal modeling generally can be viewed as "counterfactual" analysis: they attempt to understand what would be the difference between the situations—which cannot both happen —where the

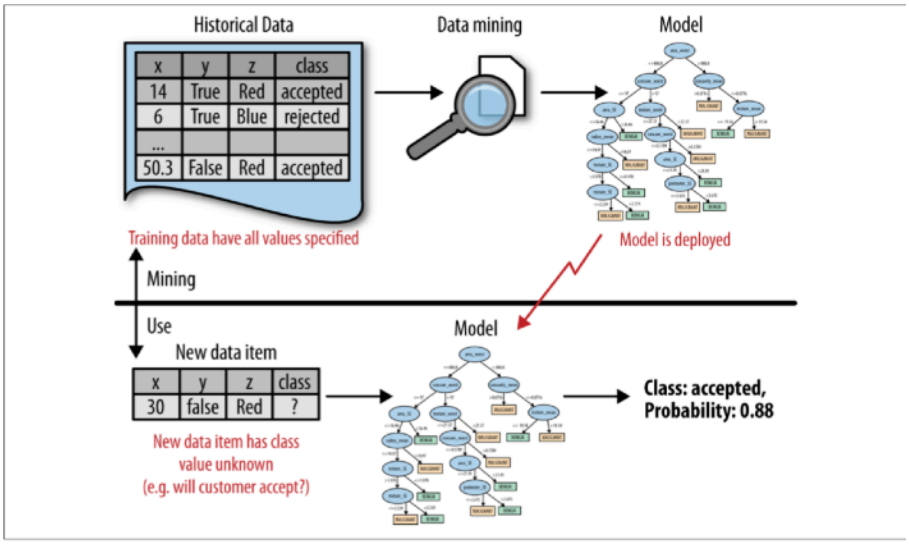
"treatment" event (e.g., showing an advertisement to a particular individual) were to happen, and were not to happen.

10. Supervised vs Unsupervised Methods: specific target

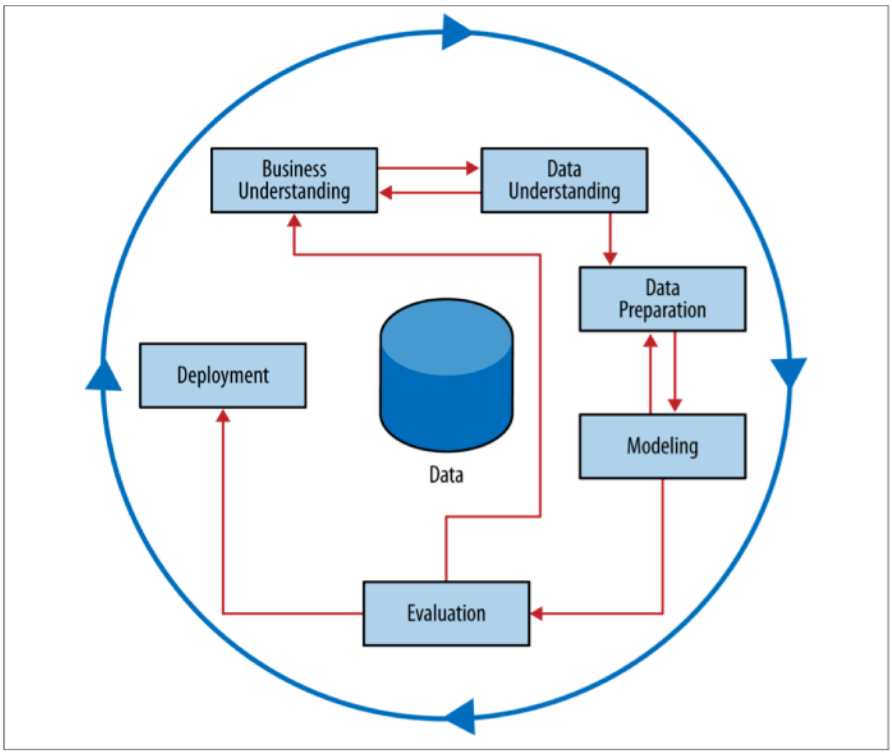
- Supervised: Provide target and there must be data on the target. (Classification, regression, casual modeling)
- Unsupervised: be left to form its own conclusions about what the examples have in common. (Clustering, co-occurrence grouping, and profiling)
- Similarity matching, link prediction, and data reduction could be either
- Label: value of target
- Classification vs. Regression: Regression involves a numeric target while classification involves a categorical (often binary) target. For business applications we often want a numerical *prediction* over a categorical target.
- In the churn example, a basic yes/no prediction of whether a customer is likely to continue to sub- scribe to the service may not be sufficient; we want to model the *probability* that the customer will continue. This is still considered classification modeling rather than regression because the underlying target is categorical. Where necessary for clarity, this is called "class probability estimation."

11. Data mining and results:

- Upper: mining of the (historical) data to find patterns and build models
- Bottom: *using* the results of data mining. (Apply to the new data)



12. Data Mining Process



- Business Understanding: Understand the business problem to be solved
- Data understanding: Estimate the costs and benefits of each data source and deciding whether further investment is merited
- Data preparation: a data preparation phase often proceeds along with data understanding, in which the data are manipulated and converted into forms that yield better results. numerical values must often be normalized or scaled so that they are comparable. A ***leak*** is a situation where a variable collected in historical data gives information on the target variable—information that appears in historical data but is not actually available when the decision has to be made
- Modeling
- Evaluation: assess the data mining results rigorously and to gain confidence that they are valid and reliable before moving on.
- Deployment: Increasingly, the data mining techniques themselves are deployed because:1) the world may change faster than the data science team can adapt, as with fraud and intrusion detection, and 2) a business has too many modeling tasks for their data science team to manually curate each model individually. In these cases, it may be best to deploy the data mining phase into production

13. It is tempting—but usually a mistake—to view the data mining process as a software development cycle

14. Other Analytics Techniques and Technologies:

- statistics : 1)it is used as a catchall term for the computation of particular numeric values of interest from data. These values often include sums, averages, rates, and so on ; 2) to denote the field of study that goes by that name, for which we might differentiate by using the proper name, Statistics. The field of Statistics provides us with a huge amount of knowledge that underlies analytics, and can be thought of as a component of the larger field of Data Science
- Database querying: A *query* is a specific request for a subset of data or for statistics about data, formulated in a technical language and posed to a database system
- Data warehousing: Data warehouses collect and coalesce data from across an enterprise, often from multiple transaction-processing systems, each with its own database.
- Regression Analysis
- Machine Learning and data mining

Chap 3: (Predictive Modeling)

15. Information: is a quantity that reduces uncertainty about something.

16. Model: is a simplified representation of reality created to serve a purpose. It is simplified based on some assumptions about what is and is not important for the specific purpose, or sometimes based on constraints on information or tractability.

17. Predictive model: a formula for estimating the unknown value of interest: the target. **Prediction** more generally means *to estimate an unknown value*. This value could be something in the future (in common us- age, true prediction), but it could also be something in the present or in the past

18. Descriptive Modeling: the primary purpose of the model is not to estimate a value but instead to gain insight into the underlying phenomenon or process

19. Supervised learning is model creation where the model describes a relationship between a set of selected variables (*attributes* or *features*) and a predefined variable called the *target* variable. The model estimates the value of the target variable as a function (possibly a probabilistic function) of the features

20. Instance: An *instance* or *example* represents a fact or a data point. An instance is described by a set of *attributes* (fields, columns, variables, or features). An instance is also sometimes called a *feature vector*, because it can be represented as a fixed-length ordered collection (vector) of feature values.
21. Induction and deduction: Induction is contrasted with deduction. Deduction starts with general rules and specific facts, and creates other specific facts from them. The *use* of our models can be considered a procedure of (probabilistic) deduction.
22. Training data: they are called *labeled* data because the value for the target variable (the label) is known.
23. Supervised Segmentation:
    - Selecting informative attributes
      - Pure: By pure we mean homogeneous with respect to the target variable.
      - **Information gain (Entropy)** : Entropy is a measure of disorder that can be applied to a set, such as one of our individual segments
      - The larger the value of entropy, the more disordered the data set is.
      - Information gain 越大越好
      - **entropy = - p_1 log (p_1)- p_2 log (p_2)- ⋯**
      - $IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots]$
      - $p(c_1)$ = size of c1/ size of parent
      - Entropy range: [0,1]
      - *We also can rank a set of attributes by their informativeness, in particular by their information gain.*
24. Decision tree

- The non-leaf nodes are often referred to as “decision nodes,” because when descending through the tree, at each node one uses the values of the attribute to make a decision about which branch to follow
- In summary, the procedure of classification tree induction is a recursive process of divide and conquer, where the goal at each step is to select an attribute to

partition the current group into subgroups that are as pure as possible with respect to the target variable

- It should be clear that we would stop when the nodes are pure, or when we run out of variables to split on
- For a problem of *n* variables, each node of a classification tree imposes an *(n–1)*-dimensional “hyperplane” decision boundary on the instance space.
- Solution to **Overfitting**: Laplace correction, the purpose of which is to moderate the influence of leaves with only a few instances.
- *p(c) = (n+1) / (n+m+2). where n is the number of examples in the leaf belonging to class c, and m is the number of examples not belonging to class c. n*和*m*很小的时候容易出现*overfitting,* 通过减小*p(probability)*来增加*uncertainty.*
- *Q:* ***However, the order in which features are chosen for the tree doesn’t exactly correspond to their ranking by the value of information gain.***

A: The answer is that the table ranks each feature by how good it is independently, evaluated separately on the entire population of instances. Nodes in a classification tree depend on the instances above them in the tree. The information gain of a feature depends on the set of instances against which it is evaluated, so the ranking of features for some internal node may not be the same as the global ranking

25. 2 Measures of attribute information:

- *information gain*, which is based on a purity measure called *entropy*
- variance reduction

Chap 4

26. *linear* classifier: is essentially a weighted sum of the values for the various attributes

27. the larger the magnitude of a feature’s weight, the more important that feature is for classifying the target

28. Linear Discriminant Functions:

- *f*(**x**) will be zero when **x** is sitting on the decision boundary (technically, **x** in that case is one of the points of the line or hyperplane). *f*(**x**) will be relatively small when **x** is near the boundary. Thus *f*(**x**) itself—the output of the linear discriminant

function—gives an intuitively satisfying ranking of the instances by their (estimated) likelihood of belong- ing to the class of interest.

29. Support Vector Machines

- The SVM's objective function incorporates the idea that a wider bar is better
- Once the widest bar is found, the linear discriminant will be the center line through the bar
- The distance between the dashed parallel lines is called the ***margin*** around the linear discriminant, and thus the objective is to maximize the margin.
- In the objective function that measures how well a particular model fits the training points, we will simply penalize a training point for being on the wrong side of the decision boundary. In the case where the data indeed are linearly separable, we incur no penalty and simply maximize the margin. If the data are *not* linearly separable, the best fit is some balance between a fat margin and a low total error penalty. The penalty for a misclassified point is proportional to the distance from the decision boundary, the error function is called "**hinge loss**"

30. Loss functions:

- The hinge loss only becomes positive when an example is on the wrong side of the boundary and beyond the margin
- Zero-One loss: assigns a loss of zero for a correct decision and one for an incorrect decision.
- Squared error loss: *Squared error* specifies a loss proportional to the square of the distance from the boundary. Squared error loss usually is used for numeric value prediction (regression), rather than classification. **Using squared error for classification also penalizes points far on the *correct* side of the decision boundary**

31. Linear Regression:

- Recall that for regression problems the target variable is numeric
- Minimize the error (the distance between the estimated values and the true values on the training data): For a particular training dataset, we could compute this error for each individual data point and sum up the results. Then the model that fits the data best would be the model with the minimum sum of errors on the training data
- Instead of minimizing the sum of absolute errors or equivalently the mean of the absolute errors across the training data, standard linear regression procedures

minimize the sum or mean of the *squares* of these errors—which gives the procedure its common name “least squares” regression (Because it’s convenience)

- Drawback: For least squares regression a serious drawback is that it is very sensitive to the data: erroneous or otherwise outlying data points can severely skew the resultant linear function.

32. Logistic Regression:

- Odds: The odds of an event is the ratio of the probability of the event occurring to the probability of the event not occurring.
- For probability estimation, logistic regression uses the same linear model as do our linear discriminants for classification and linear regression for estimating numeric target values
- The output of the logistic regression model is interpreted as the log-odds of class membership.
- These log-odds can be translated directly into the probability of class membership. Therefore, logistic regression often is thought of simply as a model for the proba- bility of class membership. You have undoubtedly dealt with logistic regression models many times without even knowing it. They are used widely to estimate quantities like the probability of default on credit, the probability of response to an offer, the probability of fraud on an account, the probability that a document is relevant to a topic, and so on.
- p(x): represent the model’s estimate of the probability of class membership of a data item represented by feature vector

Equation 4-3. Log-odds linear function

$$\log\left(\frac{p_+(\mathbf{x})}{1 - p_+(\mathbf{x})}\right) = f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots$$

Thus, Equation 4-3 specifies that for a particular data item, described by feature-vector $\mathbf{x}$, the log-odds of the class is equal to our linear function, $f(\mathbf{x})$. Since often we actually want the estimated probability of class membership, not the log-odds, we can solve for $p_+(\mathbf{x})$ in Equation 4-3. This yields the not-so-pretty quantity in Equation 4-4.

Equation 4-4. The logistic function

$$p_+(\mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}}$$

Although the quantity in Equation 4-4 is not very pretty, by plotting it in a particular way we can see that it matches exactly our intuitive notion that we would like there to be relative certainty in the estimations of class membership far from the decision boundary, and uncertainty near the decision boundary.

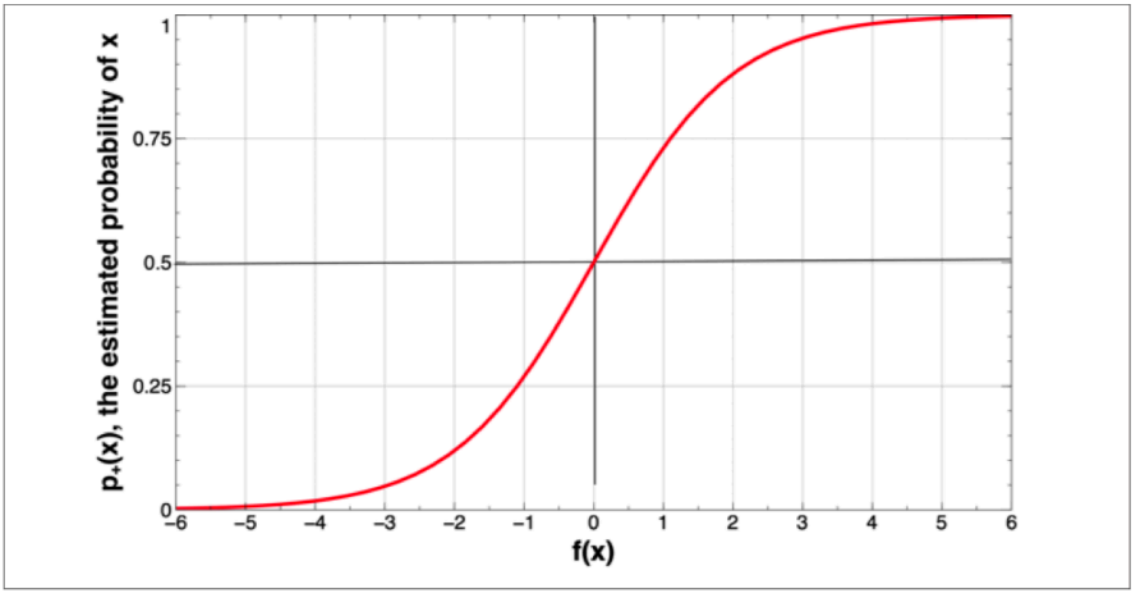


*Figure 4-10. Logistic regression's estimate of class probability as a function of f(**x**), (i.e., the distance from the separating boundary). This curve is called a "sigmoid" curve because of its "S" shape, which squeezes the probabilities into their correct range (between zero and one).*

- The figure shows that at the decision boundary (at distance $x = 0$), the probability is 0.5 (a coin toss). The probability varies approximately linearly near to the decision boundary, but then approaches certainty farther away.

This leads us to the standard objective function for fitting a logistic regression model to data. Consider the following function computing the "likelihood" that a particular labeled example belongs to the correct class, given a set of parameters **w** that produces class probability estimates $p_+(\mathbf{x})$:

$$g(\mathbf{x}, \mathbf{w}) = \begin{cases} p_+(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a } + \\ 1 - p_+(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a } \bullet \end{cases}$$

- Summing the *g* values across all the examples in a labeled dataset. And do that for different parameterized models—in our case, different sets of weights (**w**) for the logistic regression. The model (set of weights) that gives the highest sum is the model that gives the highest "likelihood" to the data—the "maximum likelihood" model. The maximum likelihood model "on average" gives the highest probabilities to the positive examples and the lowest probabilities to the negative examples.

33. Logistic regression v. Tree Induction

- A classification tree uses decision boundaries that are *perpendicular* to the instance- space axes (see Figure 4-1), whereas the linear classifier can use decision boundaries of any direction or orientation (see Figure 4-3). This is a direct consequence of the fact that classification trees select a single attribute at a time whereas linear classifiers use a weighted combination of all attributes.
- A classification tree is a "piecewise" classifier that segments the instance space re- cursively when it has to, using a divide-and-conquer approach. In principle, a classification tree can cut up the instance space arbitrarily finely into very small regions (though we will see reasons to avoid that in Chapter 5). A linear classifier places a *single* decision surface through the entire space. It has great freedom in the orientation of the surface, but it is limited to a single division into two segments. This is a direct consequence of there being a single (linear) equation that uses all of the variables, and must fit the entire data space

34. Nonlinear Functions

- The two most common families of techniques that are based on fitting the parameters of complex, nonlinear functions are *nonlinear support- vector machines* and *neural networks*
- Support vector machines have a so-called "kernel function" that maps the original features to some other feature space
- Neural networks also implement complex nonlinear numeric functions, based on the fundamental concepts of this chapter. Neural networks offer an intriguing twist. One can think of a neural network as a "stack" of models. On the bottom of the stack are the original features. From these features are learned a variety of relatively simple models. Let's say these are logistic regressions. Then, each subsequent layer in the stack applies a simple model (let's say, another logistic regression) to the outputs of the next layer down. So in a two-layer stack, we

would learn a set of logistic regressions from the original features, and then learn a logistic regression using as features the outputs of the first set of logistic regressions. We could think of this very roughly as first creating a set of "experts" in different facets of the problem (the first-layer models), and then learning how to weight the opinions of these different experts (the second-layer model)

- How are the lower-layer logistic regressions trained? The stack of models can be represented by one big parameterized numeric function. The parameters now are the coefficients of all the models, taken together. So once we have decided on an objective function representing what we want to optimize (e.g., the fit to the training data, based on some fitting function), we can then apply an optimization procedure to find the best parameters to this very complex numeric function. When we're done, we have the parameters to all the models, and thereby have learned the "best" set of lower-level experts and also the best way to com- bine them, all simultaneously.