Chapter 4

Scores and Rankings

> Money is a scoreboard where you can rank how you're doing against other people.
>
> – Mark Cuban

Scoring functions are measures that reduce multi-dimensional records to a single value, highlighting some particular property of the data. A familiar example of scoring functions are those used to assign student grades in courses such as mine. Students can then be ranked (sorted) according to these numerical scores, and later assigned letter grades based on this order.

Grades are typically computed by functions over numerical features that reflect student performance, such as the points awarded on each homework and exam. Each student receives a single combined score, often scaled between 0 and 100. These scores typically come from a linear combination of the input variables, perhaps giving 8% weight to each of five homework assignments, and 20% weight to each of three exams.

There are several things to observe about such grading rubrics, which we will use as a model for more general scoring and ranking functions:

- *Degree of arbitrariness*: Every teacher/professor uses a different trade-off between homework scores and exams when judging their students. Some weigh the final exam more than all the other variables. Some normalize each value to 100 before averaging, while others convert each score to a Z-score. They all differ in philosophy, yet every teacher/professor is certain that their grading system is the best way to do things.

- *Lack of validation data*: There is no gold standard informing instructors of the "right" grade that their students *should* have received in the course. Students often complain that I should give them a better grade, but self-interest seems to lurk behind these requests more than objectivity. Indeed, I rarely hear students recommend that I lower their grade.

S.S. Skiena, *The Data Science Design Manual,*
Texts in Computer Science, https://doi.org/10.1007/978-3-319-55444-0_4

Without objective feedback or standards to compare against, there is no rigorous way for me to evaluate my grading system and improve it.

- *General Robustness*: And yet, despite using widely-disparate and totally unvalidated approaches, different grading systems generally produce similar results. Every school has a cohort of straight-A students who monopolize a sizable chunk of the top grades in each course. This couldn't happen if all these different grading systems were arbitrarily ordering student performance. C students generally muddle along in the middle-to-lower tiers of the bulk of their classes, instead of alternating As and Fs on the way to their final average. All grading systems are different, yet almost all are defensible.

In this chapter, we will use scoring and ranking functions as our first foray into data analysis. Not everybody loves them as much as I do. Scoring functions often seem arbitrary and ad hoc, and in the wrong hands can produce impressive-looking numbers which are essentially meaningless. Because their effectiveness generally cannot be validated, these techniques are not as scientifically sound as the statistical and machine learning methods we will present in subsequent chapters.

But I think it is important to appreciate scoring functions for what they are: useful, heuristic ways to tease understanding from large data sets. A scoring function is sometimes called a *statistic*, which lends it greater dignity and respect. We will introduce several methods for getting meaningful scores from data.

4.1 The Body Mass Index (BMI)

Everybody loves to eat, and our modern world of plenty provides numerous opportunities for doing so. The result is that a sizable percentage of the population are above their optimal body weight. But how can you tell whether you are one of them?

The *body mass index* (BMI) is a score or statistic designed to capture whether your weight is under control. It is defined as

$$BMI = \frac{mass}{height^2}$$

where mass is measured in kilograms and height in meters.

As I write this, I am 68 inches tall (1.727 meters) and feeling slightly pudgy at 150 lbs (68.0 kg). Thus my BMI is $68.0/(1.727^2) = 22.8$. This isn't so terrible, however, because commonly accepted BMI ranges in the United States define:

- *Underweight*: below 18.5.
- *Normal weight*: from 18.5 to 25.
- *Overweight*: from 25 to 30.

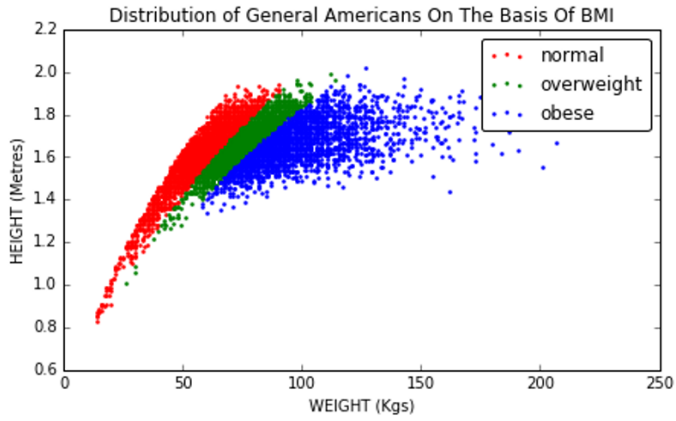

Figure 4.1: Height–weight scatter plot, for 1000 representative Americans. Colors illustrate class labels in the BMI distribution.

- *Obese*: over 30.

Thus I am considered to be in normal range, with another dozen pounds to gain before I officially become overweight. Figure 4.1 plots where a representative group of Americans sit in height–weight space according to this scale. Each point in this scatter plot is a person, colored according to their weight classification by BMI. Regions of seemingly solid color are so dense with people that the dots overlap. Outlier points to the right correspond to the heaviest individuals.

The BMI is an example of a very successful statistic/scoring function. It is widely used and generally accepted, although some in the public health field quibble that better statistics are available.

The logic for the BMI is almost sound. The square of height should be proportional to area. But mass should grow proportional to the *volume*, not area, so why is it not $mass/height^3$? Historically, BMI was designed to correlate with the percentage of body fat in an individual, which is a much harder measurement to make than height and weight. Experiments with several simple scoring functions, including m/l and m/l^3 revealed that BMI works best.

It is very interesting to look at BMI distributions for extreme populations. Consider professional athletes in American football (NFL) and basketball (NBA):

- Basketball players are notoriously tall individuals. They also have to run up and down the court all day, promoting superior fitness.
- American football players are notoriously heavy individuals. In particular, linemen exist only to block or move other linemen, thus placing a premium on bulk.

Let's look at some data. Figure 4.2 shows the BMI distributions of basketball and football players, by sport. And indeed, almost all of the basketball players

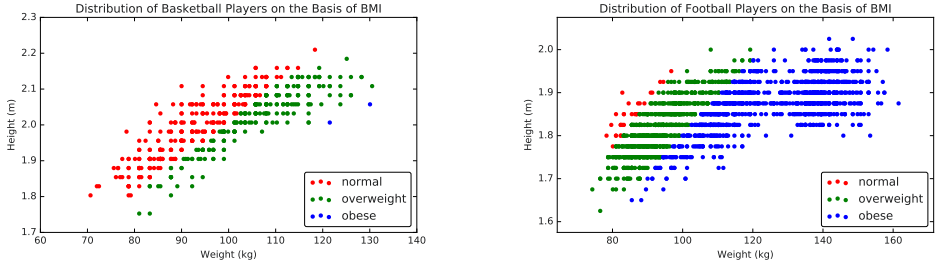

Figure 4.2: BMI distributions of professional basketball (left) and football (right) players.

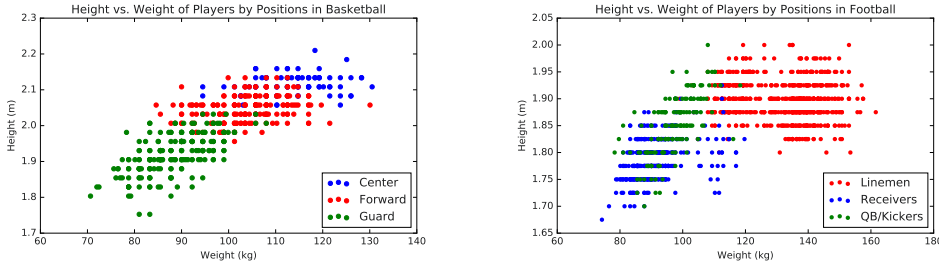

Figure 4.3: Position in basketball (left) and football (right) is largely determined by size.

have normal BMI despite their very abnormal heights. And the football players are almost uniformly animals, with most scored as obese despite the fact that they are also well-conditioned athletes. These football players are generally optimized for strength, instead of cardiovascular fitness.

In Chapter 6, we will discuss visualization techniques to highlight the presentation of data, but let's start to develop our aesthetic here. We use *scatter plots* to show each individual as a point in height–weight space, with labels (weight class or player position) shown as colors.

The breakdown of BMI by position is also revealing, and shown in Figure 4.3. In basketball, the guards are quick and sleek while the centers are tall and intimidating. So all of these positions segregate neatly by size. In football, the skill players (the quarterbacks, kickers, and punters) prove to be considerably smaller than the sides of beef on the line.

4.2 Developing Scoring Systems

Scores are functions that map the features of each entity to a numerical value of merit. This section will look at the basic approaches for building effective scoring systems, and evaluating them.

4.2.1 Gold Standards and Proxies

Historically, paper currencies were backed with gold, meaning that one paper dollar could always be traded in for $1 worth of gold. This was why we knew that our money was worth more than the paper it was printed on.

In data science, a *gold standard* is a set of labels or answers that we trust to be correct. In the original formulation of BMI, the gold standard was the body fat percentages carefully measured on a small number of subjects. Of course, such measurements are subject to some error, but by defining these values to be the gold standard for fitness we accept them to be the right measure. In gold we trust.

The presence of a gold standard provides a rigorous way to develop a good scoring system. We can use curve-fitting technique like linear regression (to be discussed in Section 9.1) to weigh the input features so as to best approximate the "right answers" on the gold standard instances.

But it can be hard to find real gold standards. *Proxies* are easier-to-find data that *should* correlate well with the desired but unobtainable ground truth. BMI was designed to be a proxy for body fat percentages. It is easily computable from just height and weight, and does a pretty good job correlating with body fat. This means it is seldom necessary to test buoyancy in water tanks or "pinch an inch" with calipers, more intrusive measures that directly quantify the extent of an individual's flab.

Suppose I wanted to improve the grading system I use for next year's data science course. I have student data from the previous year, meaning their scores on homework and tests, but I don't really have a gold standard on what grades these students *deserved*. I have only the grade I gave them, which is meaningless if I am trying to improve the system.

I need a proxy for their unknown "real" course merit. A good candidate for this might be each student's cumulative GPA in their *other* courses. Generally speaking, student performance should be conserved across courses. If my scoring system hurts the GPA of the best students and helps the lower tier, I am probably doing something wrong.

Proxies are particularly good when evaluating scoring/ranking systems. In our book *Who's Bigger?* [SW13] we used Wikipedia to rank historical figures by "significance." We did not have any gold standard significance data measuring how important these people *really* were. But we used several proxies to evaluate how we were doing to keep us honest:

- The prices that collectors will pay for autographs from celebrities *should* generally correlate with the celebrity's significance. The higher the price people are willing to pay, the bigger the star.

- The statistics of how good a baseball player is *should* generally correlate with the player's significance. The better the athlete, the more important they are likely to be.
- Published rankings appearing in books and magazines list the top presidents, movie stars, singers, authors, etc. Presidents ranked higher by historians should generally be ranked higher by us. Such opinions, in aggregate, should generally correlate with the significance of these historical figures.

We will discuss the workings of our historical significance scores in greater detail in Section 4.7.

4.2.2 Scores vs. Rankings

Rankings are permutations ordering n entities by merit, generally constructed by sorting the output of some scoring system. Popular examples of rankings/rating systems include:

- *Football/basketball top twenty*: Press agencies generally rank the top college sports teams by aggregating the votes of coaches or sportswriters. Typically, each voter provides their own personal ranking of the top twenty teams, and each team gets awarded more points the higher they appear on the voter's list. Summing up the points from each voter gives a total score for each team, and sorting these scores defines the ranking.
- *University academic rankings*: The magazine *U.S News and World Report* publishes annual rankings of the top American colleges and universities. Their methodology is proprietary and changes each year, presumably to motivate people to buy the new rankings. But it is generally a score produced from statistics like faculty/student ratio, acceptance ratio, the standardized test scores of its students and applicants, and maybe the performance of its football/basketball teams :-). Polls of academic experts also go into the mix.
- *Google PageRank/search results*: Every query to a search engine triggers a substantial amount of computation, implicitly ranking the relevance of every document on the web against the query. Documents are scored on the basis of how well they match the text of the query, coupled with ratings of the inherent quality of each page. The most famous page quality metric here is *PageRank*, the network-centrality algorithm that will be reviewed in Section 10.4.
- *Class rank*: Most High Schools rank students according to their grades, with the top ranked student honored as class valedictorian. The scoring function underlying these rankings is typically grade-point average (GPA), where the contribution of each course is weighted by its number of credits, and each possible letter grade is mapped to a number (typically

$A = 4.0$). But there are natural variants: many schools choose to weigh honors courses more heavily than lightweight classes like gym, to reflect the greater difficulty of getting good grades.

Generally speaking, sorting the results of a scoring system yields a numerical ranking. But thinking the other way, each item's ranking position (say, 493th out of 2196) yields a numerical score for the item as well.

Since scores and rankings are duals of each other, which provides a more meaningful representation of the data? As in any comparison, the best answer is that it depends, on issues like:

- *Will the numbers be presented in isolation?* Rankings are good at providing context for interpreting scores. As I write this, Stony Brook's basketball team ranks 111th among the nation's 351 college teams, on the strength of our RPI (ratings percentage index) of 39.18. Which number gives you a better idea of whether we have a good or bad team, 111th or 39.18?

- *What is the underlying distribution of scores?* By definition, the top ranked entity has a better score than the second ranked one, but this tells you nothing about the magnitude of the difference between them. Are they virtually tied, or is #1 crushing it?

  Differences in rankings *appear* to be linear: the difference between 1 and 2 seems the same as the difference between 111 and 112. But this is not generally true in scoring systems. Indeed, small absolute scoring differences can often yield big ranking differences.

- *Do you care about the extremes or the middle?* Well-designed scoring systems often have a bell-shaped distribution. With the scores concentrated around the mean, small differences in score can mean large differences in rank. In a normal distribution, increasing your score from the mean by one standard deviation (σ) moves you from the 50th percentile to the 84th percentile. But the same sized change from 1σ to 2σ takes you only from the 84th to 92.5th percentile.

  So when an organization slips from first to tenth, heads should roll. But when Stony Brook's team slides from 111th to 120th, it likely represents an insignificant difference in score and should be discounted. Rankings are good at highlighting the very best and very worst entities among the group, but less so the differences near the median.

4.2.3 Recognizing Good Scoring Functions

Good scoring functions are good because they are easily interpretable and generally believable. Here we review the properties of statistics which point in these directions:

- *Easily computable*: Good statistics can be easily described and presented. BMI is an excellent example: it contains only two parameters, and is evaluated using only simple algebra. It was found as the result of a search through all simple functional forms on a small number of easily obtained, relevant variables. It is an excellent exercise to brainstorm possible statistics from a given set of features on a data set you know well, for practice.

- *Easily understandable*: It should be clear from the description of the statistics that the ranking is relevant to the question at hand. "Mass adjusted by height" explains why BMI is associated with obesity. Clearly explaining the ideas behind your statistic is necessary for other people to trust it enough to use.

- *Monotonic interpretations of variables*: You should have a sense of how each of the features used in your scoring function correlate with the objective. Mass *should* correlate positively with BMI, because being heavy requires that you weigh a lot. Height *should* correlate negatively, because tall people naturally weigh more than short people.

  Generally speaking, you are producing a scoring function without an actual gold standard to compare against. This requires understanding what your variables mean, so your scoring function will properly correlate with this mushy objective.

- *Produces generally satisfying results on outliers*: Ideally you know enough about certain individual points to have a sense of where they belong in any reasonable scoring system. If I am truly surprised by the identity of the top entities revealed by the scoring system, it probably is a bug, not a feature. When I compute the grades of the students in my courses, I already know the names of several stars and several bozos from their questions in class. If my computed grades do not grossly correspond to these impressions, there is a potential bug that needs to be tracked down.

  If the data items really are completely anonymous to you, you probably should spend some time getting to know your domain better. At the very least, construct artificial examples ("Superstar" and "Superdork") with feature values so that they should be near the top and bottom of the ranking, and then see how they fit in with the real data.

- *Uses systematically normalized variables*: Variables drawn from bell-shaped distributions behave sensibly in scoring functions. There will be outliers at the tails of either end which correspond to the best/worst items, plus a peak in the middle of items whose scores should all be relatively similar.

  These normally-distributed variables should be turned into Z-scores (see Section 4.3) before adding them together, so that all features have comparable means and variance. This reduces the scoring function's dependence on magic constants to adjust the weights, so no single feature has too dominant an impact on the results.

$$\mu(B) = 21.9 \quad \sigma(B) = 1.92$$
$$\mu(Z) = 0 \quad \sigma(Z) = 1$$

B	19	22	24	20	23	19	21	24	24	23
Z	−1.51	0.05	1.09	−0.98	0.57	−1.51	−0.46	1.09	1.09	0.57

Figure 4.4: Taking the Z-scores of a set of values B normalizes them to have mean $\mu = 0$ and $\sigma = 1$.

Generally speaking, summing up Z-scores using the correct signs (plus for positively correlated variables and minus for negative correlations) with uniform weights will do roughly the right thing. A better function might weigh these variables by importance, according to the strength of the correlation with the target. But it is unlikely to not make much difference.

- *Breaks ties in meaningful ways*: Ranking functions are of very limited value when there are bunches of ties. Ranking the handiness of people by how many fingers they have won't be very revealing. There will be a very select group with twelve, a vast majority tied with ten, and then small groups of increasingly disabled accident victims until we get down to zero.

  In general, scores should be real numbers over a healthy range, in order to minimize the likelihood of ties. Introducing secondary features to break ties is valuable, and makes sense provided these features also correlate with the property you care about.

4.3 Z-scores and Normalization

An important principle of data science is that we must try to make it as easy as possible for our models to do the right thing. Machine learning techniques like linear regression purport to find the line optimally fitting to a given data set. But it is critical to normalize all the different variables to make their range/distribution comparable before we try to use them to fit something.

Z-scores will be our primary method of normalization. The Z-score transform is computed:

$$Z_i = (a_i - \mu)/\sigma$$

where μ is the mean of the distribution and σ the associated standard deviation.

Z-scores transform arbitrary sets of variables to a uniform range. The Z-scores of height measured in inches will be exactly the same as that of the height measured in miles. The average value of a Z-score over all points is zero. Figure 4.4 shows a set of integers reduced to Z-scores. Values greater than the mean become positive, while those less than the mean become negative. The standard deviation of the Z-scores is 1, so all distributions of Z-scores have similar properties.

Transforming values to Z-scores accomplishes two goals. First, they aid in visualizing patterns and correlations, by ensuring that all fields have an identical

mean (zero) and operate over a similar range. We understand that a Z-score of 3.87 must represent basketball-player level height in a way that 79.8 does not, without familiarity with the measurement unit (say inches). Second, the use of Z-scores makes it easier on our machine learning algorithms, by making all different features of a comparable scale.

In theory, performing a linear transformation like the Z-score doesn't *really* do anything that most learning algorithms couldn't figure out by themselves. These algorithms generally find the best coefficient to multiply each variable with, which is free to be near σ if the algorithm really wants it to be.

However, the realities of numerical computation kick in here. Suppose we were trying to build a linear model on two variables associated with U.S. cities, say, area in square miles and population. The first has a mean of about 5 and a max around 100. The second has a mean about 25,000 and a max of 8,000,000. For the two variables to have a similar effect on our model, we must divide the second variable by a factor of 100,000 or so.

This causes numerical precision problems, because a very small change in the value of the coefficient causes a very large change in how much the population variable dominates the model. Much better would be to have the variables be grossly the same scale and distribution range, so the issue is whether one feature gets weighted, say, twice as strongly as another.

Z-scores are best used on normally distributed variables, which, after all, are completely described by mean μ and standard deviation σ. But they work less well when the distribution is a power law. Consider the wealth distribution in the United States, which may have a mean of (say) \$200,000, with a $\sigma = \$200,000$. The Z-score of \$80 billion dollar Bill Gates would then be 4999, still an incredible outlier given the mean of zero.

Your biggest data analysis sins will come in using improperly normalized variables in your analysis. What can we do to bring Bill Gates down to size? We can hit him with a log, as we discussed in Section 2.4.

4.4 Advanced Ranking Techniques

Most bread-and-butter ranking tasks are solved by computing scores as linear combinations of features, and then sorting them. In the absence of any gold standard, these methods produce statistics which are often revealing and informative.

That said, several powerful techniques have been developed to compute rankings from specific types of inputs: the results of paired comparisons, relationship networks, and even assemblies of other rankings. We review these methods here, for inspiration.

4.4.1 Elo Rankings

Rankings are often formed by analyzing sequences of binary comparisons, which arise naturally in competitions between entities:

- *Sports contest results*: Typical sporting events, be they football games or chess matches, pit teams A and B against each other. Only one of them will win. Thus each match is essentially a binary comparison of merit.

- *Votes and polls*: Knowledgeable individuals are often asked to compare options and decide which choice they think is better. In an election, these comparisons are called votes. A major component of certain university rankings come from asking professors: which school is better, A or B?

  In the movie *The Social Network*, Facebook's Mark Zuckerberg is shown getting his start with FaceMash, a website showing viewers two faces and asking them to pick which one is more attractive. His site then ranked all the faces from most to least attractive, based on these paired comparisons.

- *Implicit comparisons*: From the right vantage point, feature data can be meaningfully interpreted as pairwise comparisons. Suppose a student has been accepted by both universities A and B, but opts for A. This can be taken as an implicit vote that A is better than B.

What is the right way to interpret collections of such votes, especially where there are many candidates, and not all pairs of players face off against each other? It isn't reasonable to say the one with the most wins wins, because (a) they might have competed in more comparisons than other players, and (b) they might have avoided strong opponents and beaten up only inferior competition.

The *Elo system* starts by rating all players, presumably equally, and then incrementally adjusts each player's score in response to the result of each match, according to the formula:

$$r'(A) = r(A) + k(S_A - \mu_A),$$

where

- $r(A)$ and $r'(A)$ represent the previous and updated scores for player A.

- k is a fixed parameter reflecting the maximum possible score adjustment in response to a single match. A small value of k results in fairly static rankings, while using too large a k will cause wild swings in ranking based on the latest match.

- S_A is the scoring result achieved by player A in the match under consideration. Typically, $S_A = 1$ if A won, and $S_A = -1$ if A lost.

- μ_A was the expected result for A when competing against B. If A has exactly the same skill level as B, then presumably $\mu_A = 0$. But suppose that A is a champion and B is a beginner or chump. Our expectation is that A should almost certainly win in a head-to-head matchup, so $\mu_A > 0$ and is likely to be quite close to 1.

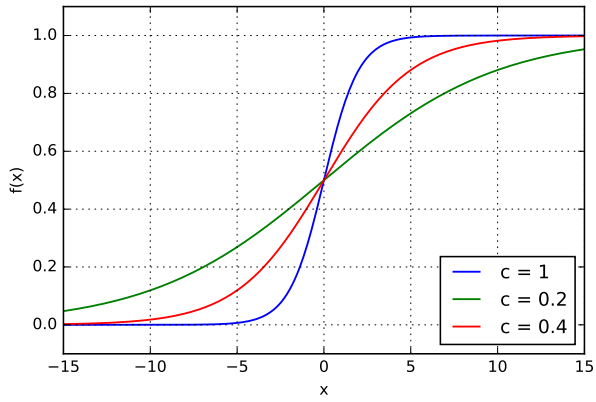

Figure 4.5: The shape of the logit function, for three different values for c.

All is clear here except how to determine μ_A. Given an estimate of the probability that A beats B ($P_{A>B}$), then

$$\mu_A = 1 \cdot P_{A>B} + (-1) \cdot (1 - P_{A>B}).$$

This win probability clearly depends on the magnitude of the skill difference between players A and B, which is exactly what is supposed to be measured by the ranking system. Thus $x = r(A) - r(B)$ represents this skill difference.

To complete the Elo ranking system, we need a way to take this real variable x and convert it to a meaningful probability. This is an important problem we will repeatedly encounter in this book, solved by a bit of mathematics called the *logit function*.

The Logit Function

Suppose we want to take a real variable $-\infty < x < \infty$ and convert it to a probability $0 \leq p \leq 1$. There are many ways one might imagine doing this, but a particularly simple transformation is $p = f(x)$, where

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

The shape of the logit function $f(x)$ is shown in Figure 4.5. Particularly note the special cases at the mid and endpoints:

- When two players are of equal ability, $x = 0$, and $f(0) = 1/2$, reflects that both players have an equal probability of winning.
- When player A has a vast advantage, $x \to \infty$, and $f(\infty) = 1$, defining that A is a lock to win the match.

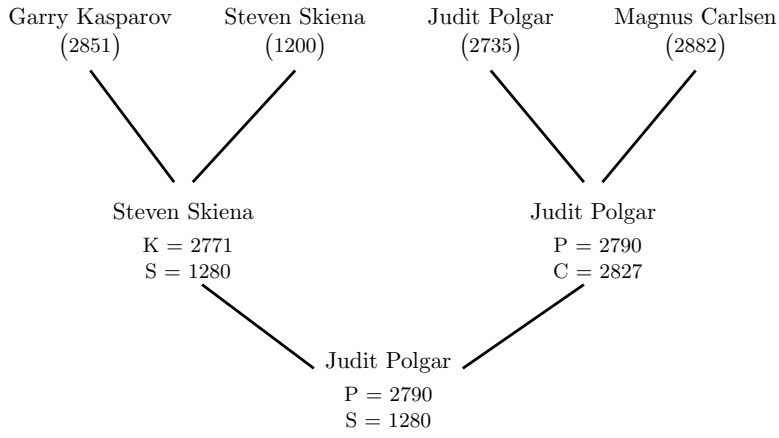

Figure 4.6: Changes in ELO scores as a consequence of an unlikely chess tournament.

- When player B has a vast advantage, $x \to -\infty$, and $f(-\infty) = 0$, denoting that B is a lock to win the match.

These are exactly the values we want if x measures the skill difference between the players.

The logit function smoothly and symmetrically interpolates between these poles. The parameter c in the logit function governs how steep the transition is. Do small differences in skill translate into large differences in the probability of winning? For $c = 0$, the landscape is as flat as a pancake: $f(x) = 1/2$ for all x. The larger c is, the sharper the transition, as shown in Figure 4.5. Indeed, $c = \infty$ yields a step function from 0 to 1.

Setting $c = 1$ is a reasonable start, but the right choice is domain specific. Observing how often a given skill-difference magnitude results in an upset (the weaker party winning) helps specify the parameter. The Elo Chess ranking system was designed so that $r(A) - r(B) = 400$ means that A has ten times the probability of winning than B.

Figure 4.6 illustrates Elo computations, in the context of a highly unlikely tournament featuring three of the greatest chess players in history, and one low-ranked patzer. Here $k = 40$, implying a maximum possible scoring swing of 80 points as a consequence of any single match. The standard logit function gave Kasparov a probability of 0.999886 of beating Skiena in the first round, but through a miracle akin to raising Lazarus the match went the other way. As a consequence, 80 points went from Kasparov's ranking to mine.

On the other side of the bracket two real chess champions did battle, with the more imaginable upset by Polgar moving only 55 points. She wiped the floor with me the final round, an achievement so clearly expected that she gained essentially zero rating points. The Elo method is very effective at updating ratings in response to surprise, not just victory.

1	A	B	A	A		A: 5
2	C	A	B	B		B: 8
3	B	C	C	D	⟶	C: 12
4	D	D	E	C		D: 16
5	E	E	D	E		E: 19

Figure 4.7: Borda's method for constructing the consensus ranking of $\{A, B, C, D, E\}$ from a set of four input rankings, using linear weights.

4.4.2 Merging Rankings

Any single numeric feature f, like height, can seed $\binom{n}{2}$ pairwise comparisons among n items, by testing whether $f(A) > f(B)$ for each pair of items A and B. We could feed these pairs to the Elo method to obtain a ranking, but this would be a silly way to think about things. After all, the result of any such analysis would simply reflect the sorted order of f.

Integrating a collection of rankings by several different features makes for a more interesting problem, however. Here we interpret the sorted order of the ith feature as defining a permutation P_i on the items of interest. We seek the consensus permutation P, which somehow best reflects all of the component permutations $P_1, \ldots, P_k$.

This requires defining a distance function to measure the similarity between two permutations. A similar issue arose in defining the Spearman rank correlation coefficient (see Section 2.3.1), where we compared two variables by the measure of agreement in the relative order of the elements.[1]

Borda's method creates a consensus ranking from multiple other rankings by using a simple scoring system. In particular, we assign a cost or weight to each of the n positions in the permutation. Then, for each of the n elements, we sum up the weights of its positions over all of the k input rankings. Sorting these n scores determines the final consensus ranking.

All is now clear except for the mapping between positions and costs. The simplest cost function assign i points for appearing in the ith position in each permutation, i.e. we sum up the ranks of the element over all permutations. This is what we do in the example of Figure 4.7. Item A gets $3 \cdot 1 + 1 \cdot 2 = 5$ points on the strength of appearing first in three rankings and second in one. Item C finishes with 12 points by finishing 2, 3, 3, and 4. The final consensus ranking of $\{A, B, C, D, E\}$ integrates all the votes from all input rankings, even though the consensus disagrees at least in part with all four input rankings.

But it is not clear that using linear weights represents the best choice, because it assumes uniform confidence in our accuracy to position elements

[1] Observe the difference between a similarity measure and a distance metric. In correlation, the scores get bigger as elements get more similar, while in a distance function the difference goes to zero. Distance metrics will be discussed more thoroughly in Section 10.1.1.

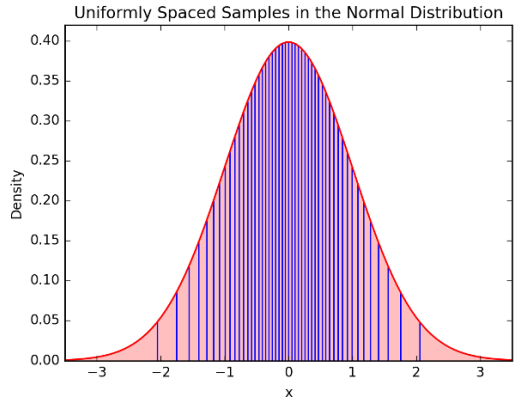

Figure 4.8: Equally-spaced values by the normal distribution are closer in the middle than the ends, making appropriate weights for Borda's method.

throughout the permutation. Typically, we will know the most about the merits of our top choices, but will be fairly fuzzy about exactly how those near the middle order among themselves. If this is so, a better approach might be to award more points for the distinction between 1st and 2nd than between 110th and 111th.

This type of weighting is implicitly performed by a bell-shaped curve. Suppose we sample n items at equal intervals from a normal distribution, as shown in Figure 4.8. Assigning these x values as the positional weights produces more spread at the highest and lowest ranks than the center. The tail regions really are as wide as they appear for these 50 equally-spaced points: recall that 95% of the probability mass sits within 2σ of the center.

Alternately, if our confidence is not symmetric, we could sample from the half-normal distribution, so the tail of our ranks is weighted by the peak of the normal distribution. This way, there is the greatest separation among the highest-ranked elements, but little distinction among the elements of the tail.

Your choice of weighting function here is domain dependent, so pick one that seems to do a good job on your problem. Identifying the very *best* cost function turns out to be an ill-posed problem. And strange things happen when we try to design the perfect election system, as will be shown in Section 4.6.

4.4.3 Digraph-based Rankings

Networks provide an alternate way to think about a set of votes of the form "A ranks ahead of B." We can construct a directed graph/network where there is a vertex corresponding to each entity, and a directed edge (A, B) for each vote that A ranks ahead of B.

The optimal ranking would then be a permutation P of the vertices which

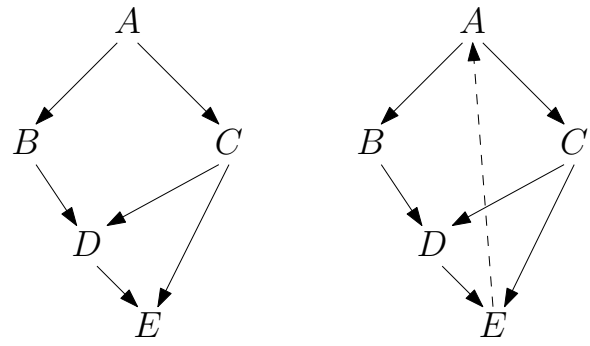

Figure 4.9: Consistently ordered preferences yield an acyclic graph or DAG (left). Inconsistent preferences result in directed cycles, which can be broken by deleting small sets of carefully selected edges, here shown dashed (right).

violates the fewest number of edges, where edge (A, B) is violated if B comes before A in the final ranking permutation P.

If the votes were totally consistent, then this optimal permutation would violate exactly zero edges. Indeed, this is the case when there are no directed cycles in the graph. A directed cycle like (A, C), (C, E), (E, A) represents an inherent contradiction to any rank order, because there will always be an unhappy edge no matter which order you choose.

A directed graph without cycles is called a *directed acyclic graph* or DAG. An alert reader with a bit of algorithms background will recall that finding this optimal vertex order is called *topologically sorting* the DAG, which can be performed efficiently in linear time. Figure 4.9 (left) is a DAG, and has exactly two distinct orders consistent with the directed edges: $\{A, B, C, D, E\}$ and $\{A, C, B, D, E\}$.

However, it is exceedingly unlikely that a real set of features or voters will all happen to be mutually consistent. The *maximum acyclic subgraph* problem seeks to find the smallest number of edges to delete to leave a DAG. Removing edge (E, A) suffices in Figure 4.9 (right). Unfortunately, the problem of finding the best ranking here is NP-complete, meaning that no efficient algorithm exists for finding the optimal solution.

But there are natural heuristics. A good clue as to where a vertex v belongs is the difference d_v between its in-degree and its out-degree. When d_v is highly negative, it probably belongs near the front of the permutation, since it dominates many elements but is dominated by only a few. One can build a decent ranking permutation by sorting the vertices according to these differences. Even better is incrementally inserting the most negative (or most positive) vertex v into its logical position, deleting the edges incident on v, and then adjusting the counts before positioning the next best vertex.

4.4.4 PageRank

There is a different and more famous method to order the vertices in a network by importance: the PageRank algorithm underpinning Google's search engine.

The web is constructed of webpages, most of which contain links to other webpages. Your webpage linking to mine is an implicit endorsement that you think my page is pretty good. If it is interpreted as a vote that "you think my page is better than yours," we can construct the network of links and treat it as a maximum acyclic-subgraph problem, discussed in the previous subsection.

But dominance isn't really the right interpretation for links on the web. PageRank instead rewards vertices which have the most in-links to it: if all roads lead to Rome, Rome must be a fairly important place. Further, it weighs these in-links by the strength of the source: a link to me from an important page should count for more than one from a spam site.

The details here are interesting, but I will defer a deeper discussion to Section 10.4, when we discuss network analysis. However, I hope this brief introduction to PageRank helps you appreciate the following tale.

4.5 War Story: Clyde's Revenge

During my sophomore year of high school, I had the idea of writing a program to predict the outcome of professional football games. I wasn't all that interested in football as a sport, but I observed several of my classmates betting their lunch money on the outcome of the weekend football games. It seemed clear to me that writing a program which accurately predicted the outcome of football games could have significant value, and be a very cool thing to do besides.

In retrospect, the program I came up with now seems hopelessly crude. My program would average the points scored by team x and the points allowed by team y to predict the number of points x will score against y.

$$P_x = \frac{((\text{points scored by team } x) + (\text{points allowed by team } y))}{2 \times (\text{games played})}$$

$$P_y = \frac{((\text{points scored by team } y) + (\text{points allowed by team } x))}{2 \times (\text{games played})}$$

I would then adjust these numbers up or down in response to other factors, particularly home field advantage, round the numbers appropriately, and call what was left my predicted score for the game.

This computer program, *Clyde*, was my first attempt to build a scoring function for some aspect of the real world. It had a certain amount of logic going for it. Good teams score more points than they allow, while bad teams allow more points than they score. If team x plays a team y which has given up a lot of points, then x should score more points against y than it does against

teams with better defenses. Similarly, the more points team x has scored against the rest of the league, the more points it is likely to score against y.

Of course, this crude model couldn't capture all aspects of football reality. Suppose team x has been playing all stiffs thus far in the season, while team y has been playing the best teams in the league. Team y might be a much better team than x even though its record so far is poor. This model also ignores any injuries a team is suffering from, whether the weather is hot or cold, and whether the team is hot or cold. It disregards all the factors that make sports inherently unpredictable.

And yet, even such a simple model can do a reasonable job of predicting the outcome of football games. If you compute the point averages as above, and give the home team an additional three points as a bonus, you will pick the winner in almost two-thirds of all football games. Compare this to the even cruder model of flipping a coin, which predicts only half the games correctly. That was the first major lesson *Clyde* taught me:

Even crude mathematical models can have real predictive power.

As an audacious 16 year-old, I wrote to our local newspaper, *The New Brunswick Home News*, explaining that I had a computer program to predict football game results and was ready to offer them the exclusive opportunity to publish my predictions each week. Remember that this was back in 1977, well before personal computers had registered on the public consciousness. In those days, the idea of a high school kid actually *using* a computer had considerable gee-whiz novelty value. To appreciate how much times have changed, check out the article the paper published about *Clyde* and I in Figure 4.10.

I got the job. *Clyde* predicted the outcome of each game in the 1977 National Football League. As I recall, *Clyde* and I finished the season with the seemingly impressive record of 135–70. Each week, they would compare my predictions against those of the newspaper's sportswriters. As I recall, we all finished within a few games of each other, although most of the sportswriters finished with better records than the computer.

The *Home News* was so impressed by my work that they didn't renew me the following season. However, *Clyde*'s picks for the 1978 season were published in the *Philadelphia Inquirer*, a much bigger newspaper. I didn't have the column to myself, though. Instead, the *Inquirer* included me among ten amateur and professional prognosticators, or touts. Each week we had to predict the outcomes of four games against the point spread.

The point spread in football is a way of handicapping stronger teams for betting purposes. The point spread is designed to make each game a 50/50 proposition, and hence makes predicting the outcome of games much harder.

Clyde and I didn't do very well against the spread during the 1978 National Football League season, and neither did most of the other *Philadelphia Inquirer* touts. We predicted only 46% of our games correctly against the spread, a performance good (or bad) enough to finish 7th out of the ten published prognosticators. Picking against the spread taught me a second major life lesson:

Student uses computers to predict football winners

By JEFF LEEBAW
Home News staff writer

EAST BRUNSWICK — A 16-year-old East Brunswick High School student has found a way to combine an interest in football with a fascination for computers.

Steven Skiena says he can determine, with a high degree of accuracy, the outcome of professional football games by feeding a computer pertinent information about competing teams.

"The winners will almost always be correct," said the high school junior who lives at 5 Currier Road off Dunhams Corner Road. "I had an 86 per cent accuracy rate when I started predicting at the end of last season."

He does it by feeding the computer a myriad of statistics that include team records, points scored and allowed, average yards gained and allowed during a game, a breakdown of the yards gained and allowed into rushing and passing categories, performances at home and on the road, and more.

The information is gathered from weekly compilations of football statistics and standings. Skiena puts the facts on index cards and then types them into one of the six computer terminals at the high school or a terminal at The Library where he works part-time after school.

"I get a winning team, a decimal score for each team and a point spread," said the teen-ager who completed a computer programming course last year at the high school.

His first attempt at picking winners involved a Monday night game between the Oakland Raiders, the eventual Super Bowl victors, and the Cincinnati Bengals.

It was a difficult game to analyze because Cincinnati was fighting for a playoff berth while Oakland had already clinched a spot in the post season competition.

"Nobody knew whether Oakland would be giving 100 per cent," Skiena said. "But my calculations indicated they would win by 24-20. The final score was 35-20. They went all out."

Skiena said he went on to pick 12 of the 14 winners the following week and accurately predicted Oakland would defeat Minnesota in the Super Bowl.

The National Football League's 1976 Record Book, which breaks down last year's statistics for each of the league's 28 teams, will supply most of Skiena's information for the first few weeks of the 1977 season. He will also use statistics from the final two exhibition games played this year by each of the teams.

Skiena wrote a computer program based on 17 statistical variables that might come into play during a football game.

The computer, in essence, asks him questions and he types the answers.

"It starts out by asking for the names of the teams," he said. "Then it will ask for records, points scored, etc...."

The computer program also attempts to include such intangible variables as injuries.

"The injuries are broken down into offense, defense and quarterback," he explained. "Obviously a quarterback injury is the most serious. It's too difficult to break down injuries for every position. When the computer asks for the number of injuries on defense, I'll type in one, two or whatever the figure is.

Will Skiena use his computer results to enter the variety of football pools and contests that are available during the season?

"No," he said. "I don't like to bet on my own predictions. Last season a friend bet on a game I predicted and it happened to be one of the few that were wrong."

STEVEN SKIENA
...to test his accuracy

Predictions published

Steven Skiena will get a chance to display his skill as a pro football prognosticator each Sunday in The Home News.

The youngster's weekly selections will be an "added ingredient to our football coverage," according to Home News Executive Editor Robert E. Rhodes.

"I think it's interesting enough for us to give it a shot," Rhodes said of the teen-ager's computer method of determining the outcome of games. "He seems like an earnest young man and we'll stand behind him."

Skiena will receive a "modest stipend" for predicting the winners, scores for each team and briefly explaining the reasons for his conclusions.

His column will appear for the first time in Sunday's sports section when the National Football League (NFL) opens its 1977 season with a slate of 13 games. Skiena will also predict the outcome of the league's Monday night games.

The Home News is publishing the column in the sports section to test the youngster's system and to offer football fans an entertaining feature. Its purpose is not to encourage betting.

"We'll be printing his selections close enough to the time of the game to prevent betting," Rhodes said. "There's big interest in pro football and more than anything else we want to test his system. I'll be rooting for him."

Figure 4.10: My first attempt at mathematical modeling.

Crude mathematical models do not have real predictive power when there is real money on the line.

So Clyde was not destined to revolutionize the world of football prognostication. I pretty much forgot about it until I assigned the challenge of predicting the Super Bowl as a project in my data science class. The team that got the job was made up of students from India, meaning they knew much more about cricket than American football when they started.

Still, they rose to the challenge, becoming fans as they built a large data set on the outcome of every professional and college game played over the past ten years. They did a logistic regression analysis over 142 different features including rushing, passing, and kicking yardage, time of possession, and number of punts. They then proudly reported to me the accuracy of their model: correct predictions on 51.52% of NFL games.

"*What!*" I screamed, "*That's terrible!*" "Fifty percent is what you get by flipping a coin. Try averaging the points scored and yielded by the two teams, and give three points to the home team. How does that simple model do?"

On their data set, this Clyde-light model picked 59.02% of all games correctly, much much better than their sophisticated-looking machine learning model. They had gotten lost in the mist of too many features, which were not properly normalized, and built using statistics collected over too long a history to be representative of the current team composition. Eventually the students managed to come up with a PageRank-based model that did a little bit better (60.61%), but Clyde did almost as well serving as a baseline model.

There are several important lessons here. First, garbage in, garbage out. If you don't prepare a clean, properly normalized data set, the most advanced machine learning algorithms can't save you. Second, simple scores based on a modest amount of domain-specific knowledge can do surprisingly well. Further, they help keep you honest. Build and evaluate simple, understandable baselines before you invest in more powerful approaches. Clyde going baseline left their machine learning model defenseless.

4.6 Arrow's Impossibility Theorem

We have seen several approaches to construct rankings or scoring functions from data. If we have a gold standard reporting the "right" relative order for at least some of the entities, then this could be used to train or evaluate our scoring function to agree with these rankings to the greatest extent possible.

But without a gold standard, it can be shown that no best ranking system exists. This is a consequence of *Arrow's impossibility theorem*, which proves that no election system for aggregating permutations of preferences satisfies the following desirable and innocent-looking properties:

- The system should be complete, in that when asked to choose between alternatives A and B, it should say (1) A is preferred to B, (2) B is preferred to A, or (3) there is equal preference between them.

Voter	Red	Green	Blue
x	1	2	3
y	2	3	1
z	3	1	2

Figure 4.11: Preference rankings for colors highlighting the loss of transitivity. Red is preferred to green and green preferred to blue, yet blue is preferred to red.

- The results should be transitive, meaning if A is preferred to B, and B is preferred to C, then A must be preferred to C.
- If every individual prefers A to B, then the system should prefer A to B.
- The system should not depend only upon the preferences of one individual, a dictator.
- The preference of A compared to B should be independent of preferences for any other alternatives, like C.

Figure 4.11 captures some of the flavor of Arrow's theorem, and the non-transitive nature of "rock-paper-scissors" type ordering. It shows three voters (x, y, and z) ranking their preferences among colors. To establish the preference among two colors a and b, a logical system might compare how many permutations rank a before b as opposed to b before a. By this system, red is preferred to green by x and y, so red wins. Similarly, green is preferred to blue by x and z, so green wins. By transitivity, red should be preferred to blue by implication on these results. Yet y and z, prefer blue to red, violating an inherent property we want our election system to preserve.

Arrow's theorem is very surprising, but does it mean that we should give up on rankings as a tool for analyzing data? Of course not, no more than Arrow's theorem means that we should give up on democracy. Traditional voting systems based on the idea that the *majority rules* generally do a good job of reflecting popular preferences, once appropriately generalized to deal with large numbers of candidates. And the techniques in this chapter generally do a good job of ranking items in interesting and meaningful ways.

Take-Home Lesson: We do not seek correct rankings, because this is an ill-defined objective. Instead, we seek rankings that are useful and interesting.

4.7 War Story: Who's Bigger?

My students sometimes tell me that I am history. I hope this isn't true quite yet, but I am very interested in history, as is my former postdoc Charles Ward. Charles and I got to chatting about who the most significant figures in history

were, and how you might measure this. Like most people, we found our answers in Wikipedia.

Wikipedia is an amazing thing, a distributed work product built by over 100,000 authors which somehow maintains a generally sound standard of accuracy and depth. Wikipedia captures an astonishing amount of human knowledge in an open and machine-readable form.

We set about using the English Wikipedia as a data source to base historical rankings on. Our first step was to extract feature variables from each person's Wikipedia page that should clearly correlate with historical significance. This included features like:

- *Length*: Most significant historical figures should have longer Wikipedia pages than lesser mortals. Thus article length in words provides a natural feature reflecting historical wattage, to at least some degree.

- *Hits*: The most significant figures have their Wikipedia pages read more often than others, because they are of greater interest to a larger number of people. My Wikipedia page gets hit an average of twenty times per day, which is pretty cool. But Issac Newton's page gets hit an average of 7700 times per day, which is a hell of a lot better.

- *PageRank*: Significant historical figures interact with other significant historical figures, which get reflected as hyperlink references in Wikipedia articles. This defines a directed graph where the vertices are articles, and the directed edges hyperlinks. Computing the PageRank of this graph will measure the centrality of each historical figure, which correlates well with significance.

All told, we extracted six features for each historical figure. Next, we normalized these variables before aggregating, essentially by combining the underlying rankings with normally-distributed weights, as suggested in Section 4.4.2. We used a technique called *statistical factor analysis* related to principal component analysis (discussed in Section 8.5.2), to isolate two factors that explained most of the variance in our data. A simple linear combination of these variables gave us a scoring function, and we sorted the scores to determine our initial ranking, something we called *fame*.

The top twenty figures by our fame score are shown in Figure 4.12 (right). We studied these rankings and decided that it didn't really capture what we wanted it to. The top twenty by fame included pop musicians like Madonna and Michael Jackson, and three contemporary U.S. presidents. It was clear that contemporary figures ranked far higher than we thought they should: our scoring function was capturing current fame much more than historical significance.

Our solution was to decay the scores of contemporary figures to account for the passage of time. That a current celebrity gets a lot of Wikipedia hits is impressive, but that we still care about someone who died 300 years ago is much more impressive. The top twenty figures after age correction are shown in Figure 4.12 (left).

Signif	Name
1	Jesus
2	Napoleon
3	William Shakespeare
4	Muhammad
5	Abraham Lincoln
6	George Washington
7	Adolf Hitler
8	Aristotle
9	Alexander the Great
10	Thomas Jefferson
11	Henry VIII
12	Elizabeth I
13	Julius Caesar
14	Charles Darwin
15	Karl Marx
16	Martin Luther
17	Queen Victoria
18	Joseph Stalin
19	Theodore Roosevelt
20	Albert Einstein

Fame	Person
1	George W. Bush
2	Barack Obama
3	Jesus
4	Adolf Hitler
5	Ronald Reagan
6	Bill Clinton
7	Napoleon
8	Michael Jackson
9	W. Shakespeare
10	Elvis Presley
11	Muhammad
12	Joseph Stalin
13	Abraham Lincoln
14	G. Washington
15	Albert Einstein
16	John F. Kennedy
17	Elizabeth II
18	John Paul II
19	Madonna
20	Britney Spears

Figure 4.12: The top 20 historical figures, ranked by significance (left) and contemporary fame (right).

Now *this* was what we were looking for! We validated the rankings using whatever proxies for historical significance we could find: other published rankings, autograph prices, sports statistics, history textbooks, and Hall of Fame election results. Our rankings showed a strong correlation against all of these proxies.

Indeed, I think these rankings are wonderfully revealing. We wrote a book describing all kinds of things that could be learned from them [SW13]. I proudly encourage you to read it if you are interested in history and culture. The more we studied these rankings, the more I was impressed in their general soundness.

That said, our published rankings did not meet with universal agreement. Far from it. Dozens of newspaper and magazine articles were published about our rankings, many quite hostile. Why didn't people respect them, despite our extensive validation? In retrospect, most of the flack we fielded came for three different reasons:

- *Differing implicit notions of significance*: Our methods were designed to measure *meme-strength*, how successfully these historical figures were propagating their names though history. But many readers thought our methods should capture notions of historical *greatness*. Who was most important, in terms of changing the world? And do we mean world or just the English-speaking world? How can there be no Chinese or Indian

figures on the list when they represent over 30% of the world's population?

We must agree on what we are trying to measure before measuring it. Height is an excellent measure of size, but it does not do a good job of capturing obesity. However, height is very useful to select players for a basketball team.

- *Outliers*: Sniff tests are important to evaluating the results of an analysis. With respect to our rankings, this meant checking the placement of people we knew, to confirm that they fell in reasonable places.

  I felt great about our method's ranking of the vast majority of historical figures. But there were a few people who our method ranked higher than any reasonable person would, specifically President George W. Bush (36) and teenage TV star Hilary Duff (1626). One could look at these outliers and dismiss the entire thing. But understand that we ranked almost 850,000 historical figures, roughly the population of San Francisco. A few cherry-picked bad examples must be put in the proper context.

- *Pigeonhole constraints*: Most reviewers saw only the rankings of our top 100 figures, and they complained about exactly where we placed people and who didn't make the cut. The women's TV show *The View* complained we didn't have enough women. I recall British articles complaining we had Winston Churchill (37) ranked too low, South African articles that thought we dissed Nelson Mandela (356), Chinese articles saying we didn't have enough Chinese, and even a Chilean magazine whining about the absence of Chileans.

  Some of this reflects cultural differences. These critics had a different implicit notion of significance than reflected by English Wikipedia. But much of it reflects the fact that there are exactly one hundred places in the top 100. Many of the figures they saw as missing were just slightly outside the visible horizon. For every new person we moved into the top hundred, we had to drop somebody else out. But readers almost never suggested names that should be omitted, only those who had to be added.

What is the moral here? Try to anticipate the concerns of the audience for your rankings. We were encouraged to explicitly call our measure *meme-strength* instead of *significance*. In retrospect, using this less-loaded name would have permitted our readers to better appreciate what we were doing. We probably also should have discouraged readers from latching on to our top 100 rankings, and instead concentrate on relative orderings within groups of interest: who were the top musicians, scientists, and artists? This might have proved less controversial, better helping people build trust in what we were doing.

4.8 Chapter Notes

Langville and Meyer [LM12] provide a through introduction to most of the ranking methods discussed here, including Elo and PageRank.

One important topic not covered in this chapter is *learning to rank* methods, which exploit gold standard ranking data to train appropriate scoring functions. Such ground truth data is generally not available, but proxies can sometimes be found. When evaluating search engines, the observation that a user clicked the (say) fourth item presented to them can be interpreted as a vote that it should have been higher ranked than the three placed above it. SVMrank [Joa02] presents a method for learning ranking functions from such data.

The heuristic proposed minimizing edge conflicts in a vertex order is due to Eades et. al. [ELS93]. My presentation of Arrow's impossibility theorem is based on notes from Watkins [Wat16].

The war stories of this chapter were drawn very closely from my books *Calculated Bets* and *Who's Bigger?* Don't sue me for self-plagiarism.

4.9 Exercises

Scores and Rankings

4-1. *[3]* Let X represent a random variable drawn from the normal distribution defined by $\mu = 2$ and $\sigma = 3$. Suppose we observe $X = 5.08$. Find the Z-score of x, and determine how many standard deviations away from the mean that x is.

4-2. *[3]* What percentage of the standard normal distribution ($\mu = 0$, $\sigma = 1$) is found in each region?

(a) $Z > 1.13$.

(b) $Z < 0.18$.

(c) $Z > 8$.

(d) $|Z| < 0.5$.

4-3. *[3]* Amanda took the Graduate Record Examination (GRE), and scored 160 in verbal reasoning and 157 in quantitative reasoning. The mean score for verbal reasoning was 151 with a standard deviation of 7, compared with mean $\mu = 153$ and $\sigma = 7.67$ for quantitative reasoning. Assume that both distributions are normal.

(a) What were Amanda's Z-scores on these exam sections? Mark these scores on a standard normal distribution curve.

(b) Which section did she do better on, relative to other students?

(c) Find her percentile scores for the two exams.

4-4. *[3]* Identify three successful and well-used scoring functions in areas of personal interest to you. For each, explain what makes it a good scoring function and how it is used by others.

4-5. *[5]* Find a data set on properties of one of the following classes of things:

(a) The countries of the world.

(b) Movies and movie stars.

(c) Sports stars.

(d) Universities.

Construct a sensible ranking function reflecting quality or popularity. How well is this correlated with some external measure aiming at a similar result?

4-6. *[5]* Produce two substantially different but sensible scoring functions on the same set of items. How different are the resulting rankings? Does the fact that both have to be sensible constrain rankings to be grossly similar?

4-7. *[3]* The scoring systems used by professional sports leagues to select the most valuable player award winner typically involves assigning positional weights to permutations specified by voters. What systems do they use in professional baseball, basketball, and football? Are they similar? Do you think they are sensible?

Implementation Projects

4-8. *[5]* Use Elo ratings to rank all the teams in a sport such as baseball, football, or basketball, which adjusts the rating in response to each new game outcome. How accurately do these Elo ratings predict the results of future contests?

4-9. *[5]* Evaluate the robustness of Borda's method by applying k random swaps to each of m distinct copies of the permutation $p = \{1, 2, \ldots, n\}$. What is the threshold where Borda's method fails to reconstruct p, as a function of n, k, and m?

Interview Questions

4-10. *[5]* What makes a data set a gold standard?

4-11. *[5]* How can you test whether a new credit risk scoring model works?

4-12. *[5]* How would you forecast sales for a particular book, based on Amazon public data?

Kaggle Challenges

4-13. Rating chess players from game positions.
`https://www.kaggle.com/c/chess`

4-14. Develop a financial credit scoring system.
`https://www.kaggle.com/c/GiveMeSomeCredit`

4-15. Predict the salary of a job from its ad.
`https://www.kaggle.com/c/job-salary-prediction`