Person-Target Profiling

Paul Barrett

Within many domains of enquiry or activity it is important to be able to describe features of individuals associated with an activity or behaviour which contribute to some criterion of current or future success or failure. The span of these domains is wide, encompassing areas such as the assessment of risk of recidivist violence (Webster, Harris, Rice, Cormier, & Quinsey, 2004; Monahan, Steadman, Appelbaum, Robbins, Mulvey, Silver, Roth, & Grisso, 2000), clinical psychopathology profiles (Groth-Marnat, 2003), career suitability (Peterson, Mumford, Borman, Jeanneret, Fleishman, Levin, Campion, Mayfield, Morgeson, Pearlman, Gowing, Lancaster, Silver, and Dye, 2001), job success via personality and intelligence scores (Schmidt and Hunter, 1998, 2004), consumer shopping preference (Dickson, 2001; Taylor and Cosenza, 2002), and internet consumer advertising (Raghu, Kannan, Rao, and Winston, 2001). Within all these domains, the aim is not just to describe the differentiating features of individuals or groups, but to incorporate the knowledge gained into a predictive framework or system which will permit future individuals to be classified, ranked, scored, or chosen/rejected according to their relative "standing" on those feature attributes.

This chapter seeks to describe the substantive logic, methods, coefficients, and comparison procedures most commonly used within the domain of person-target profiling. These generally fall into methods of constructing "prototype" score vectors that are definitional for some preferred criterion, then using these score vectors as "targets" against which new individuals may be compared. Hence, much of this work is concerned with constructing score vectors and with choosing or designing the comparison coefficients which will be used to represent target profile disparity. However, with the advent of a new class of algorithmic data methods into statistical science, a significant part of this chapter is devoted to explaining what they are, and demonstrating their likely superior utility within a typical profiling application. The reason these methods are referred to as "algorithmic" is because they tend to work entirely within a framework of maximising the predictive or profile classification accuracy, not by assuming a standard probabilistic linear model for the data, but by *attempting to "learn" the relations between input variables and criterion without any assumptions necessarily being made as to linearity or distributions of observations on either predictor variables or outcome variables. Indeed, these two approaches at first glance look as though they share nothing in common. But, each in its own way is seeking to produce a template or some function of certain variables such that an individual possessing "target" values will be correctly classified as "member of, not member of, or close to" a criterion target (high performance employees say). The most common approach has been to develop a set of prototypical scores on some variables, and make the decision based upon some distance or covariance-sensitive coefficient value. The new approach outlined here is to treat the profiling problem as a "classifier" construction problem – and so seek to create a classifier "function" or "model" which optimally predicts membership of the target group. By using a "real-world" set of data which is amenable to analysis by both approaches, it is hoped the reader can appreciate the principles of each whilst observing the benefits of perhaps treating "profiling" in future as more of a "classifier construction" issue than a conventional score vector distance estimation problem.*

1. The Profile, the Target, and the Comparison

A profile is defined by Collins English Dictionary (1991,3rd edition) as: "a graph, table, or list of scores representing the extent to which a person, field, or object exhibits various tested characteristics or tendencies". However, a profile might also consist of any set of attributes whose magnitudes, features, ranges, categories, or even functions, are definitive of a particular group, class, cluster, or even a single individual case. Within organizational psychology for instance, a profile may be defined as an outline or shape formed by plotting magnitudes for an individual, group, or job, into a one, two, or three dimensional space. But, a profile of a successful business executive may contain scores on psychological variables and competency attributes as well as categorical and ordered category information, some function of which defines a successful executive rather than merely an "average" one.

Once a profile is constructed, this may be proposed as a template or "target" against which future applicants, consumers, students, or other individuals may be compared in terms of similarity or dissonance. The simple logic behind such a procedure is that if one can obtain a discrete profile which discriminates between say high spending v low spending consumers, or credit-safe v credit-risk individuals, then any further individuals may be compared to this profile such that a judgement or classification may be made as to their similarity to a particular group. In order to maximise advertising return or minimise financial exposure, knowing that an individual is closer to one or another group's profile is likely to be of substantive financial value. Further, the comparison procedure can be automated, is objective, and possesses an "audit-trail". That is, the user of such a profile comparison procedure will know how the final measurement, ranking, or classification decision was made, and which variables or functions contributed to it. Possibly the one exception to this is the use of a neural net for person classification or outcome prediction (Sommer, Olbrich, and Arendasy, 2004), where investigators sometimes barely consider the operational characteristics of the net (because of the intrinsic difficulty of doing so) and focus solely on the predictive outcome.

There are four issues to be considered by an investigator prior to the construction of a profile:

Dimensionality. The question here is how many dimensions will be used to characterise a profile? Most profiles are one-dimensional, with magnitudes or amounts of a set of attributes defining that dimension. But, it is possible to define a profile in two dimensions where two features of a set of attributes are definitive of the profile space for an individual. For example, we may assess the magnitude of an individual's competencies in one dimension, whilst simultaneously assessing their preference for using these competencies during a working day. It is the position of the competencies in two-dimensional space which defines the profile, and against which profile comparisons will be made. Likewise, if we add a third dimension to that profile, that of "Job Effectiveness", we now have a means of describing an individual in three dimensional space, where their competency magnitude, personal work-balance preference, and overall job effectiveness may be maintained as a uniquely defining attribute set for comparison or selection purposes. Examples of each of these kinds of profiles as used in commercial applications are given below in Section 3.

Metrics. Three sub-issues are of concern here. The first is concerned with whether the profile attributes are to be considered quantitative. That is, are the variable magnitudes to be treated as possessing equal-intervals such that arithmetic operations and transformations will be carried out on them? Alternatively, will the profile be said to constitute ordinal magnitudes only, thus constraining profile definition and comparisons to ordinal-ranking operations only? Or, perhaps the profile consists solely of categories, such as one that might be constructed using configural frequency analysis (von Eye, 1990)? Will the profile consist of discrete classes of equal-interval, ordinal, and categorical variables – such that the overall comparison consists of "blocks" of comparison rules – each of which is weighted into an overall comparison index? The second issue is that of linearity. If the profile attributes are to be considered quantitative, then should the profile comparison be made under an assumption of linear or non-linear comparative similarity? For example, if wishing to compute the similarity between two profiles, should a simple linear difference measure be used, or perhaps a pearson or intraclass correlation? If not, perhaps a Euclidean distance or derived non-linear weighted distance function should be used? Finally, if a profile is constituted from several attributes, are these attributes to be considered independent from one another such that component distance calculations between specific attributes are equally weighted into the overall profile similarity or distance measure? In many cases, this is not even considered when using conventional linear profile comparison strategies, such as using Euclidean distances, intraclass or pearson correlations. Yet, if a target profile is built from two unbalanced (in quantity) sets of high within-set correlated attribute variables, then similarity to this profile may be gained from similarity to the majority set of highly correlated variables even though there is almost no similarity to the minor set. Although such an artificial example is unlikely to occur in practice, the studious profile constructor will ensure that the highest attribute variable intercorrelations in the profile are not markedly skewed toward a majority set within those variables. Alternatively, some adjustment might have to be made to the comparison function between the profiles in order to adjust for the profile attribute interrelations.

***Variables and Defined Outcomes or Classes**.* This is a critical issue that concerns a most fundamental perspective on the nature of any profile to be constructed. Put simply, do you wish to construct a profile on the basis of the variable observations you have within a dataset, using a pre-defined criterion or criteria that are definitive of group or class membership, or do you wish to construct a profile on the basis of a set of hypothesised latent variables which are deemed causal for a hypothesised set of class latent classes or groups? The specific area of interest here is constructing profiles for groups of individuals, where either the groups are defined in advance and the profile is constructed directly on the basis of the observations on the profile attribute variables, or where the aim is to simultaneously construct both groups and profile attributes under a set of assumptions that assume both causal

variables and classes/groups are latent (to be inferred from the variable observations and their covariances). Latent Profile analysis is a recent methodology that addresses the latter option (Muthén, 2001; Muthén and Muthén, 2001); a factor mixture model built upon the foundation of Latent Class Analysis (Lazarsfeld and Henry, 1968) and common factor analysis (see also Magidson and Vermunt (2004) for a recent exposition of latent class models). In essence, it assumes that the common latent factor model can be applied to a set of variables (common to all latent classes) whose magnitudes are caused by one or more latent variables, with the estimated scores on the latent variables in turn defining the "categorical" latent classes. The reason for such a model being termed a "mixture" model is because the classes are hypothesised as being discrete latent entities whose composition is not immediately identifiable from the observations comprising the score vector for each member of the sample of individuals. The definition of Latent Class Analysis given by Waller (2004) as "a powerful method for finding homogenous groups in mixed samples" clearly distinguishes this kind of approach to profiling over direct *a priori* defined category class methods. The key feature of latent profiling methods is the apparent impossibility of defining a meaningful "outcome" class or classes in terms of readily available or meaningful criteria which are not themselves part of the observed variable-set that might form one or more profiles. Indeed, latent profile analysis seems to be a very specific method of exploratory inductive reasoning and class "discovery", rather than a focussed attempt to generate profiles from variables which might be predictive of already specified, meaningful classes. The MAXCOV multivariate taxometric methodology of Meehl (1995) and Waller and Meehl (1998) is another approach to this "find the classes" problem, although it is only applicable to the situation where just two latent classes are hypothesised to exist and relies entirely upon conditioning operations on the observed variable data rather than derived latent causes of them. If we profile directly using the variables and data at hand, with a predefined criterion of outcome class, then there is no need whatsoever for Latent Profile Analysis. However, it is possible that an investigator might wish to invoke a latent variable model as underlying the observations on the manifest (observed) variables, or even a derived summary variable model as in principal components analysis. But, if there are no predefined outcome classes against which a defining profile might be constructed for each, then latent profile analysis, along with those other methods for determining clusters or groups within data (e.g. cluster analysis, Q-factor analysis, multi-dimensional scaling, correspondence analysis, Kohonen feature maps and other unsupervised learning algorithms) will be required.

Comparison Strategy. Finally, having decided upon the methodologies appropriate to constructing the target profile defining a group or single case, the investigator must decide upon how comparisons are to be made between new individual person attribute scores and the target profile. With one dimensional score vector profiles, this invariably is a choice between several measures of vector similarity or distance. In two or more dimensions, the choice of comparison methodologies tends to focus on distance-related procedures. If the profile is considered an ordinal representation of attribute magnitudes, then profile comparisons will be constrained invariably to ranking and ordering operations, with the end result being a ranked form of distance or similarity. However, as we replace simple score vector profiles with those based upon classifier functions, categorical variables, or Statistical Prediction Rules (Swets, Dawes, and Monahan, 2000), then the output from such comparisons shifts from a magnitude distance or ranked estimate to one of class membership. That is, the output of such a comparison is not an indicative estimate of distance or similarity from profile to target, but a discrete classifier decision into one or more groups. Latent Profile analysis and other quantitative classification methodologies such as logistic regression or discriminant function analysis will also yield a probability or likelihood estimate of group membership along with the estimated group classification. However, classifier functions invariably are used for decision-support and clinical diagnostics with classifications treated as discrete outcomes (Swets et al , 2000).

In general, the majority of simple profiling solutions in organizational, psychological, and consumer-focussed commercial applications use a score vector to represent the profile of magnitudes on measured attributes for one or more individuals. It is therefore useful to define the essential features of this kind of profile, as many of the usual profile coefficients used to compare score vectors are insensitive to one or more of these transformations. Figure 1 below shows a simple score profile for twelve attributes:

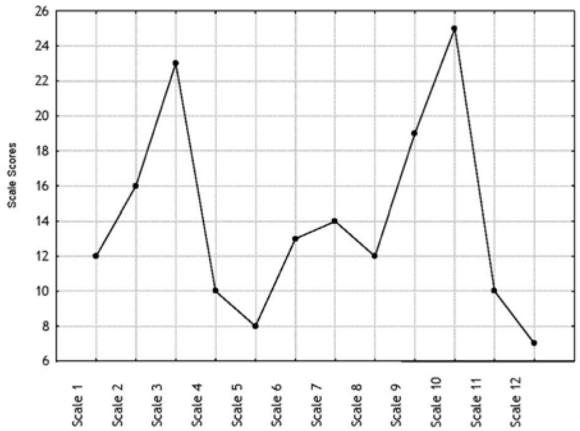


Figure 1: A simple profile example; the scores on twelve attributes scales plotted as a profile.

These scores could be *prototypical* mean values or median values of some group of individuals, the scores from a single individual (a "star performer" say), or even "ideal" scores generated subjectively by an expert.

Cronbach and Gleser (1953) introduced three terms to describe a profile:

- ***Elevation:*** the mean of all scores defined in a single profile. In the above example this is equal to 14.083
- ***Scatter:*** the square root of the sum of squares of a single profile's deviation scores about the Elevation for that profile. Essentially the standard deviation of scores within a profile, multiplied by the square root of the number of attributes constituting the profile (scale scores in our example). The standard deviation for our data (using the unbiased formula for a standard deviation) is: 5.45372. Multiplying this by the square root of the number of scale scores is 18.89224, which is our Scatter Value.
- ***Shape:*** the residual information left in each score of a profile, after equating for the elevation and scatter indexes by subtracting out the elevation and dividing the resultant deviation score by the scatter value.

More formally:

Given a profile of scores x_i, where i = 1 to n scales:

$$Elevation = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

$$Scatter = \left(\sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}\right) \cdot \sqrt{n} \tag{2}$$

$$Shape = \frac{(x_i - \bar{x})}{Scatter} \tag{3}$$

To show the effect of these transformations, let us look at two score profiles, one designated as a target profile the other as a comparison profile. Figure 2 displays the raw profiles.

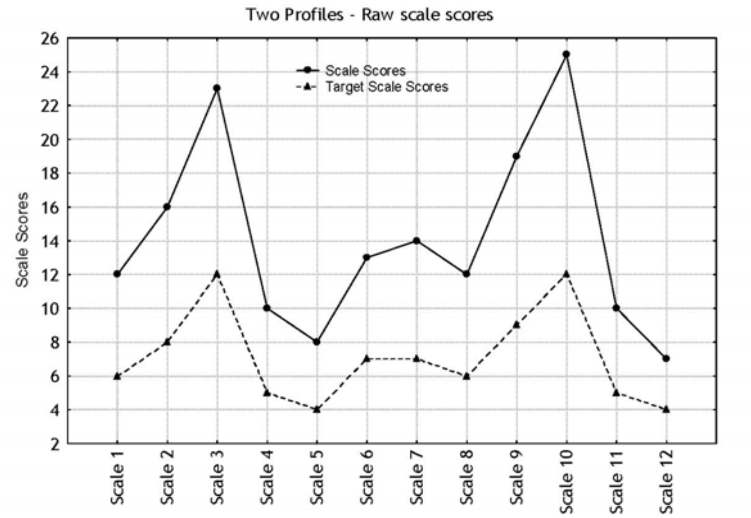


Figure 2: Two profiles of scale scores

What we see is that the two profiles look the same shape (more or less) but are different in that the target scores are all lower than the other set of scale scores. If we subtract each profile's elevation parameter from the respective profiles, the results can be seen in Figure 3.

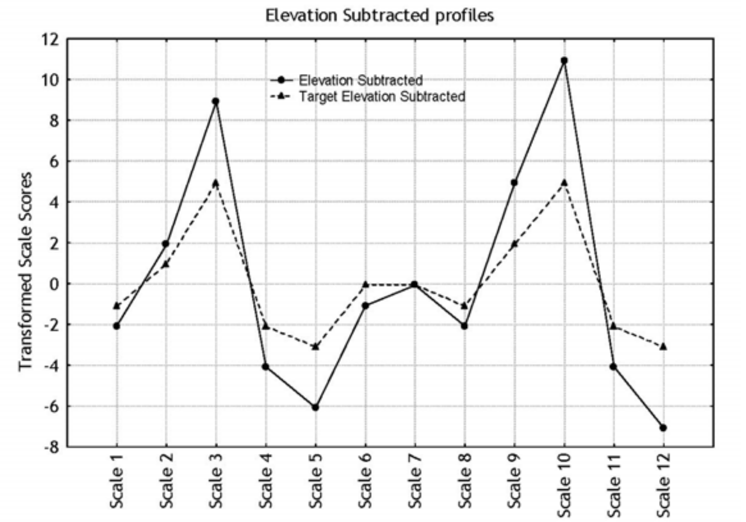


Figure 3: Two profiles of scale scores with their respective elevation (means) subtracted

Now the profiles seem to be much closer to each other - centered as they are around each of their respective means. What has happened is that we have removed the average "elevation" from each profile - so that each may be expressed in a common metric that always possesses a mid-point (average transformed data mean) of 0. Figure 4 shows the effect of equating for the variability associated with each profile. The shape, accentuation, or pattern, has been retained, but we have equated profiles for "level" and "variability".

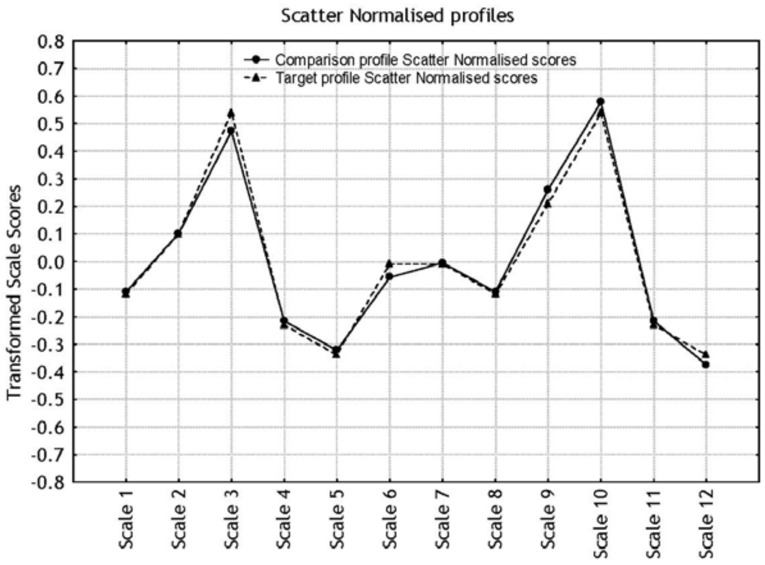


Figure 4: Two profiles of scale scores, scatter normalised

It is instructive to see what difference these transformation make to any simple form of profile similarity coefficient. If we look at the pearson and two forms of the intraclass correlation between the two profiles before and after transformation, we obtain the results as displayed in Table 2.

Table 1: Three profile matching coefficients computed over the profiles in Figure 2

	Pearson r	ICC Model 2	ICC Model 3
Raw profile comparison	0.99	0.35	0.78
De-Elevation comparison	0.99	0.79	0.78
De-Scatter comparison	0.99	0.99	0.99

As can be seen clearly from this table, the data transformations do make a substantive difference to the calculation of these three indices at least. Choice of single vector-score matching coefficient is critical within profiling applications. Not only are some like the Pearson completely blind to differences in elevation and scatter, but their expected distributions make some (like normalised euclidean distance) completely useless for profile comparisons. This issue will be taken up further in Section 3 below.

2. Constructing Target Profiles

This section details the most widely used methods for constructing target profiles or classifier functions/rules in order that comparisons or classifications may be conducted on new instances or cases. In the world of commerce, the prevalence of predefined class-identifying criteria removes the need for latent class analysis, although some investigators do invoke latent variables as a means of reducing dimensionality within large variable-count datasets. Discrete groups and classes are readily defined by many different real-world *a priori* criteria for group membership or class, such as job performance success or failure, profit/cost-neutral/loss, high-performer/low performer, job-goal preference class, consumer product-preference groups, financial-risk groups etc. The inductive process of profile construction is thus confined solely to the patterning of the variables as predictors or descriptors of predefined-class membership.

2.A. Score Vector Profiles

A score vector profile is the most basic form of profile, where scores or magnitudes on one or more attributes in one or more dimensions is used as the target information against which new individuals will be compared. Figure 1 above shows respectively a single vector of attribute scores and their graphical display. Perhaps the most obvious exemplar of this kind of profile is that used within many psychometric personality tests where an individual’s scores on a test are displayed as a profile. This kind of display may also be used to represent a target profile on the personality attributes, such that an individual’s scores can be plotted against this profile. For example, figure 5 below shows the author’s personality test scores compared with a target profile constructed from a group of New Zealand working executive students studying for a professional qualification within the faculty of business at the University of Auckland. The test used is Psytech International’s (www.psytech.co.uk) 15FQ+ multi-attribute personality test with the profiling achieved via the Psytech GeneSys™ profiling module software. The grey line is the target profile against which the individual’s scores are being compared.

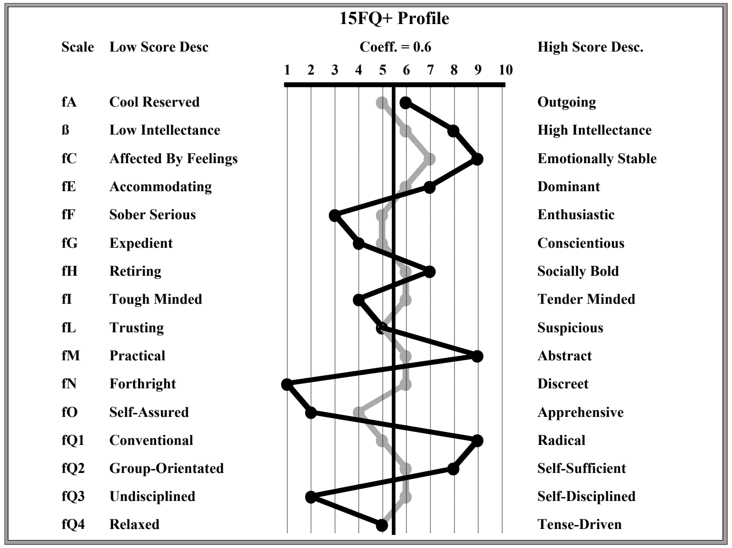


Figure 5: A Person-Target profile matching display, comparing the author's test scores (black line) to those of the target profile (grey line) of a small group of New Zealand Executive Business students, using the Psytech International 15FQ+ personality test within the GeneSys™ Profiler module.

An example of a two-dimensional profile is taken from the StaffCV Inc. (www.staffcv.com) job-preference mapping system. This unique profiling application uses two dimensions in which individuals rate their preference for work-behaviour attributes whilst simultaneously indicating their preferred frequency for engaging in these behaviours. The profiles for IT helpdesk/technical support staff is shown in Figure 6. The full measurement range is between 0 to 100 on each axis, but for clarity the "zoomed" profile display are shown.

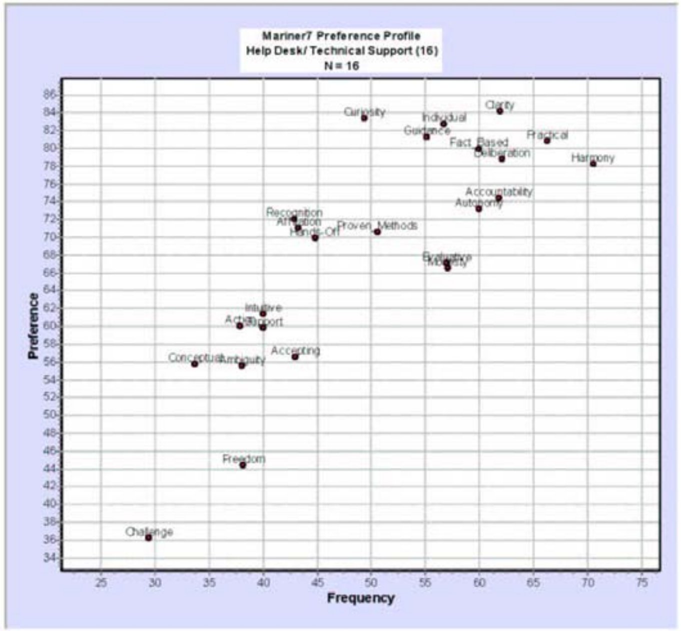


Figure 6: Help Desk/Technical Support staff, 2-dimensional work behaviour attribute target profile.

Finally, an example of a possible model for a three dimensional profile is provided in Figure 7. This profile is intrinsically non-linear. What it depicts is a three dimensional profile for a single attribute (say working alone), where the preference for that attribute (magnitude preference), the amount one wishes to engage in doing it (frequency), and the weight by which departures from an "optimal" comparative match are multiplied, are represented in three-dimensional space.

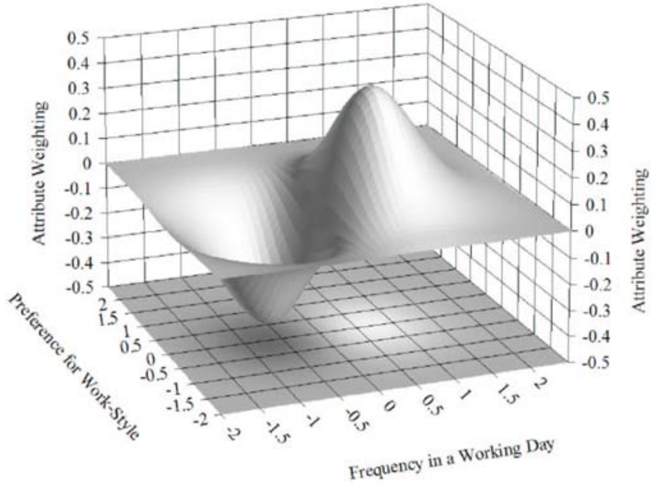


Figure 7: A hypothetical three-dimensional target profile for a job, where the preference for a particular work-behaviour and the frequency of engaging in it during a working day is also associated with a weight-function (attribute-weight) which is assigned so as to provide maximum predictive facility against job-performance.

This kind of profile is quite different in many respects from a standard one and two-dimensional score profile, in that the z-axis dimension includes a weighting function which permits both penalty and reward of a two-dimensional comparative match. The weight function is required to be optimised or modelled using an outcome criterion as the calibration device. In this way, comparative assessment of profile similarity can be suitably "adjusted" by the choice of weight function. Note also that a multi-attribute profiler would include one of these three-dimensional sub-profiles for each attribute which is used as a constituent target.

Construction Issues

With score vector profiles there are two types of construction process, differentiated by whether the profile is constructed upon the basis of subjective reasoning or empirical data. The first process creates an "*ideal profile*". That is, a target profile is created not by empirical analysis but rather based upon what an investigator or subject matter expert (SME) regards as a target set of scores. The profile is considered "ideal" as it represents what the optimal target would look like. Of course, whether it is optimal remains a subjective matter until later examination of any evidence which might indicate that the profile has indeed optimised some other criterion for which the profile was actually used to enhance (e.g. increased productivity from the newly selected employees). The second process is based entirely upon empirical data drawn from either a single individual's scores or from a group of such scores. Usually, the use of a single individual's scores to form the basis of a target profile is known as "*star performer*" profiling, based upon the fact that in this particular target construction process the single case is an exceptional employee (e.g. the person with the highest sales figures, the highest consistent job performance rating, the most highly rated person from a subordinate 360 rating process etc.). This is an extremely risky form of profile construction. There are many reasons why an individual may be a "star performer", and it's quite possible that few of them are included in the set of attributes defining the profile. Further, how will you know whether the attribute magnitudes are in fact in any way unique? That is, the attribute magnitudes for the "star performer" may be found to be equivalent to those of all other employees or persons for whom the profile is being constructed. Unless some criterion separation work is undertaken by sampling the attributes amongst say a group of individuals known to differ markedly in the criterion of interest, and determining the key "star performer" profile attribute differences, then the use of such a profile resembles naïve casino gambling rather than a considered business strategy.

If the profile is based upon a group of individuals, it is known as "*homogenous group*" profiling. This denotes the main assumption of this approach; a group of individuals are considered homogenous with respect to some criterion (e.g. best sales team, most productive machinists, highest performing digital image photo-processing group, a particular job-type such as accountants, general managers, HR administrators etc.). The target profile is constructed by using the average or median score on each attribute upon which each member of the group has been assessed. For example, the target profile in Figure 5 above is from a small group of New Zealand executive business students, using the average score across the 15FQ+ personality traits to represent the mean profile. In Figure 6, the two-dimensional profiles are constructed using the mean scores on both the preference and frequency dimensions across the 24 work-attributes. The key problem with this form of "*homogenous group*" is bound up in that word "*homogenous*". The group of individuals may well be homogenous with respect to a criterion (sales performance in dollars), but quite heterogeneous with respect to the attributes used in the profile. Further, by using a single "point-estimate" for each attribute such as the mean or median, no consideration is given at all to the variability of scores on any attribute. Probably the best way to cope with variability is to use it via a simple relative weight function, where more or less weight is given to a distance or similarity comparison based upon the variability of scores on a particular attribute within the target group. In this way, less weight is given to a profile attribute comparison value if the variability of the target group on that attribute is large. Likewise, where the variability is small, then more weight is assigned to that comparison value.

In relation to the construction of three-dimensional profiles, the empirical process is somewhat more complex. The key problem concerns the design of the response function over the two-dimensional plane formed by the two rating scales. In Figure 7 above, the two-dimensional plane is formed by the magnitudes on the two scales "Preference for Work-Style" and "Frequency in a Working Day". It is instructive to show the kind of construction processes and decisions that have to be made when constructing such a profile. In contrast to the kind of profile shown in Figure 7, let us assume we measure two preference attributes simultaneously: the "Preference to be the Leader" and the "Preference for being an Adviser" amongst members of our high performing work team. Note here that we are creating a preference profile where an individual makes a judgement on two attributes simultaneously in direct relation to one another, but without the requirement that showing a preference for one attribute must therefore imply a lack of preference for the other (the reader might have noticed how much this process mimics that of an ipsative rating questionnaire measure of personality, in some respects it does, but this kind of profiler assessment avoids the assumption that choices are oppositional or are indeed linearly related). So, we would first assess each individual on these attributes and "create a single summary point in the two-dimensional plane defined by the preference ratings of "Being a Leader" and "Being an Adviser".

The third attribute (the z-axis) is as yet undefined and unmeasured. At this stage, we could work entirely within a linear metric, comparing individuals against this "ideal point" and using a conventional distance indicator such as Euclidean or simple absolute distance. However, at the moment, no information is being conveyed by the 3rd dimension of "Criterion Weight". This weight can be used to boost or attenuate the distance/similarity between an individual's scores on both attributes and the target "profile point". This allows the profile constructor to be sensitive to the critical importance perhaps of values on either attribute which may be considered so "off-target" as to indicate that the corrected similarity or distance should be boosted so as to have a substantive effect on the overall profile similarity or distance calculation. If we have reason to believe that for example too much of both attributes actually has a poor relationship with our criterion, whilst too little is very poor, we might construct a non-linear weighting scheme (either from empirical data or from SME) such as that shown in Figure 8.

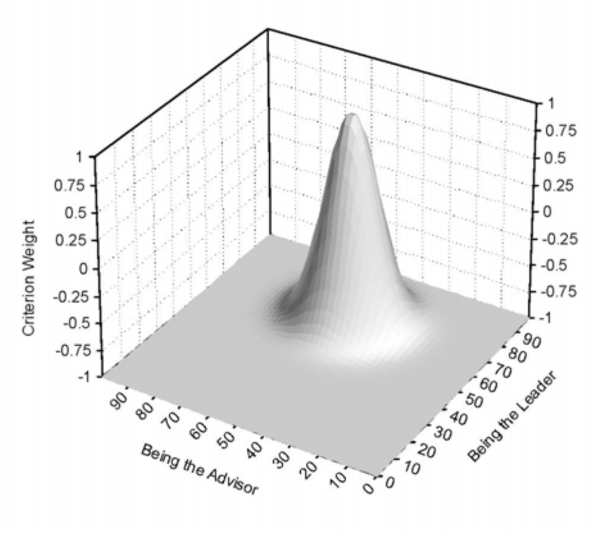


Figure 8: A single average point estimate of scores for a high performance work-team on the preference attributes of "Being an Advisor" and "Being a Leader", but now using a symmetric non-linear criterion weight scheme.

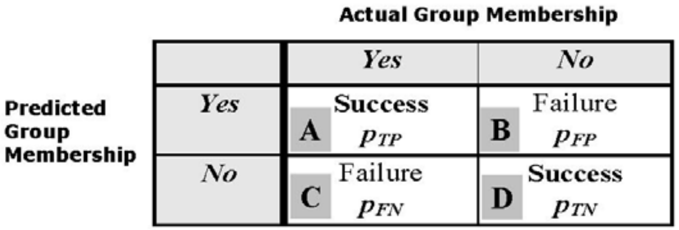
This scheme uses the equation for a symmetric bivariate unit normal distribution (r=0.0) to construct a weight "cone" surrounding the optimal target profile point. The width of the base (and acceleration) of the weights from high positive to low negative is controlled by the standard deviation parameters. As with the homogenous group one and two-dimensional profiling discussion above, the variability (and subsequent importance) of a target profile point-estimate can be incorporated directly into the scheme here by adjusting the width of the base and the acceleration of weight values, as well as the peak magnitude of the weight "cone". So, as an individual two-dimensional score departs from the optimal target score, its effect on an overall profile coefficient is weighted non-linearly. It is this "shaping" of the distance or similarity function that is critical to the design of multi-attribute profiles, as the designer can "sharpen" up or make more "diffuse" each particular attribute match, which is especially important for aggregated profile comparison coefficients. However, as will be outlined in Section 3 below, the cost of "shaping" the distance function is that simulation work is required in order to generate expected coefficient distributions for both random and "typical" profile data. But, what this small example shows is that profile construction can and should be much more of a careful optimal design process than a simple "sum the scores, take the average, and start profiling" exercise.

2.B. Statistical Prediction Rule Classifiers

In Section A, we saw that a profile can consist of a set of target scores in vector format, against which new scores are compared. But, a profile might also consist of one or more functions of variables, where some weighted combination of the profile variables and their scores is predictive of being a member of a particular group (e.g. successful or unsuccessful call-centre operatives, or five groups differentiated by a job performance rating). Whereas score-vector profiling tends to be utilized in ranking schemes, where comparison profiles are ordered in relation to their similarity or distance to the target, function-based profiles tend to be used as "classifiers". That is, their primary purpose is to classify or partition individuals into one or more ordered or unordered categories or classes based upon the function value.

When working with such models, their success or failure is directly measured using their classification accuracy. In contrast score-vector target profiling solutions produce ordered comparison magnitudes, where a decision/cut-off value is chosen by the user (e.g. when using a similarity index which varies between 0 and 1, with 1 indicating identity, a value above 0.7 might be chosen). It is rare in score-vector profiling to see a categorization of the profile matches into a set of categorized outcomes. In classifier-based profiling, this is the norm. Before discussing classifier specifics, it is as well to explain some of the terms and statistics most popularly associated with profile classifiers and any such "decision-support" process. Many applications use a dichotomous outcome variable, such as employ/not employ, success/failure, promote/not promote, sales > $50k/sales < $30k, absenteeism > 20 days/absenteeism < 10 days etc. These kinds of data can be expressed in a classification matrix or table such as that shown in Table 2.

Table 2: A 2x2 classification table with cell notation. The probabilities might also be expressed as frequencies or even percentages in many applications.



		Actual Group Membership	
		Yes	*No*
Predicted Group Membership	*Yes*	A Success p_{TP}	B Failure p_{FP}
	No	C Failure p_{FN}	D Success p_{TN}

where:
p_{TP} = probability of a True Positive decision
p_{FP} = probability of a False Positive decision
p_{FN} = probability of a False Negative decision
p_{TN} = probability of a True Negative decision

Such a table is computed using known *a priori* group membership, which is then contrasted with the predicted group membership via the classifier. The four cells of such a table define the four possible outcomes. So for example, predicting membership of a group of High Performance Work (HPW) Teams, a True Positive (cell A) outcome is where the prediction of being a member is in agreement with actual membership. A True Negative (Cell D) outcome is where the prediction of NOT being a member of the HPW teams is also correct. A False Positive (cell B) outcome is where the classifier predicted an individual as being a member of the HPW teams, but in reality they are not. Likewise, a False Negative (cell C) outcome is where the classifier predicts an individual as NOT being a member of the HPW teams, but in reality they are. Each of these four cell values is a proportion or probability which can easily be calculated by dividing the number of classification counts in a cell (A, B, C, or D) by the total number of classifications made (A+B+C+D). The goal of a classifier is to obtain 100% classification efficiency (overall predictive accuracy or **P**redictive **E**fficiency) which is calculated as:

$$PE = p_{TP} + p_{TN} \tag{2.1}$$

We might also wish to express the probability that a "*group member"* prediction correctly predicted the actual group membership relative to the number of positive (group member) predictions made. This is known as the **P**ositive **P**ower of **P**rediction and is calculated as:

$$PPP = \frac{p_{TP}}{p_{TP} + p_{FP}} \tag{2.2}$$

Likewise we might wish to express the probability that a "NOT a *group member"* prediction correctly predicted the actual group non-membership relative to the number of negative (NOT a group member) predictions made. This is known as the **N**egative **P**ower of **P**rediction and is calculated as:

$$NPP = \frac{p_{TN}}{p_{TN} + p_{FN}} \tag{2.3}$$

As we increase the decisions to be made beyond two categories, these statistics no longer become relevant. However, classification accuracy and the classification/misclassification matrix is still an extremely important guide to where accurate and inaccurate predictions are being made.

Finally, the most basic goal of any classifier profiling method is achieving classification accuracy whilst showing some degree of cross-validation (Witten and Frank, 2000; Webb, 2002). That is, not only should the initial production of a classifier function demonstrate good classification accuracy, it should also be shown to cross-validate or replicate this accuracy on further samples of data. By increasing the number of parameters in any classifier function, the likelihood is that classification accuracy will increase accordingly. In fact, when the number of parameters in a linear equation equals the number of cases, the prediction will be perfect as a parameter is available to account for each case's unique variability. This is where cross-validation is essential.

There are five methods of cross-validation used by classifier designers:

- ***Holdout Sample Validation.*** Here the analyst will partition a dataset into one or more "training" and "holdout" samples. A training sample is one which is used to construct the classifier function (calculating the weights in say a regression model). A holdout sample is a sample of data which is not used to construct the classifier. Instead, it is used once the classifier is constructed, in order to determine whether the classifier functions as expected on a completely different set of data. So, if a classifier produces an overall 70% classification accuracy, then we would expect it to produce roughly the same level of accuracy on a new dataset if we had not capitalised on chance in the training sample (too many parameters causing over-fitting of the training sample data).
- ***V-fold Cross-Validation.*** This type of cross-validation is required when no holdout sample is available and the total sample is too small to have a holdout sample partitioned from it. V-fold cross-validation involves resampling data from the total sample, taking *V* subsamples as equal in size as possible. The classifier function is then computed *V* times, each time leaving out one of the sub-samples from the computations, and using that subsample as a holdout sample for cross-validation, so that each subsample is used *V* - 1 times in the training sample and just once as the holdout sample. The classifier parameters are then averaged over the V-folds to produce the final classifier function. So, if we used 3-fold cross-validation, we would use two-thirds of the data each time to construct the classifier, and one-third of the data as a holdout validation sample.
- ***Global/Stratified Cross-Validation.*** This method maximizes the size of the training sample relative to the holdout sample, but increases the number of folds, *V,* to compensate for the smaller holdout samples. The technique partitions the total dataset into *V* equal-sized folds, where the composition of the data in each fold maintains the relative class proportions as found in the total sample. Then, the training dataset is constructed from *V* – 1 folds chosen at random from the collection of *V* folds, with the *Vth* fold retained as the holdout sample upon which the classifier will be tested. This procedure is completed *V* times, with the final solution being composed from the average of all the classifier functions. So, if we used 10-fold cross-validation, we would use nine-tenths of the data each time to construct the classifier, and one-tenth of the data as a holdout validation sample. Witten and Frank (2000) recommend this method be used with *V*=10 whenever insufficient data exist for a large holdout and training sample. This is also considered the standard method for cross-validation using a single sample of data.
- ***Jackknife/Leave-One-Out Cross-Validation.*** This is a technique, first introduced by Quenouille (1949) and later expanded upon by Tukey (1958) and Mosteller (1971). Essentially, the jackknife procedure involves removing a

single observation from a set of data, then computing the classifier on the remaining *n*-1 cases. For *n* cases in a dataset, *n* jackknife classifier solutions are computed, and the parameter values averaged over all jackknife solutions to provide "robust" parameter estimates. Essentially, this is *V*-fold cross-validation where *V*= *n*. Each "holdout" case is fitted by the classifier and the aggregated fit statistics form the estimated error-rate of the averaged classifier. When using Discriminant Function Analysis to form a classifier, the Lachenbruch (1975) is sometimes used to estimate the robust classification accuracy – but, since this is equivalent to computing a standard jackknife procedure, the use of Mosteller's (1971) procedure is now more straightforward, and it allows for an optimisation procedure to be used on the final averaged classification function profile attribute variable weights. Using the three key indexes for a dichotomous classifier (false positive rate, false negative rate, and overall classification accuracy/predictive efficiency), it is possible to optimise the jackknifed classifier which provides its output as a real valued function. By selecting "threshold" classifier function values, one at a time, and computing the classification table and indices using that threshold as a cutoff for group selection, it is possible to calculate an optimisation function that minimizes the false positive and false negative error rates whilst maximizing the predictive efficiency. Figure 9 displays such a function for a set of data from a company which required a classifier developed to select two groups of individuals, one of which whose members showed little interest in "Leading People", the other group's members showed great interest. The classifier was a multi-attribute discriminant function model which was constructed using jackknifed estimates. In this particular case, the optimization procedure could not improve upon the Lachenbruch/jackknife procedure of 66% classification accuracy. However, for some other attributes, this optimization procedure produced an 8% increase in classification accuracy (from 70 to 78% for example).

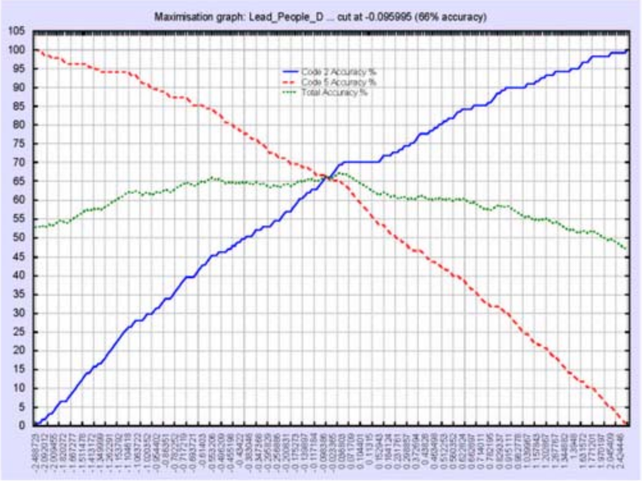


Figure 9: An optimisation function for a dichotomous outcome classifier, minimizing false positive and false negative rates whilst maximizing predictive efficiency.

- ***Bootstrap/Simulations.*** The final method of cross-validation involves resampling the total dataset *with replacement*. That is, subsamples of data are sampled from the total dataset, but as a sample observation is taken from the dataset, it is not removed from the total dataset but is able to be selected "at random" again via the sampling process. This differs from the *V* fold methods in that these sample *without replacement*. Efron (1979) first introduced this technique for parameter estimation in statistics. In classifier profiling, the bootstrap is used to construct several instances of a training sample, with the holdout sample each time composed of those values not bootstrapped into the training sample. The averaged classifier parameters are used to construct the final classifier function with the error rates computed from the average of the holdout sample classifications.

Regression and Discriminant Models

Invariably, ordered rating or category functions are built upon a simple linear regression such as that shown in equation 2.1

$$g_{\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}} = b_0 + b_1 p_1 + b_2 p_2 + b_3 p_3 + \ldots + b_n p_n \qquad (2.4)$$

where
g = groups 1-5, ordered in terms of performance ratings
b_o = an equation constant
$b_{1..n}$ = profile attribute weights
$p_{1..n}$ = profile attribute scores

The classification into groups [1..5] is achieved by rounding the real-valued equation result 'g' for a set of profile scores. The equation tries to assign individuals with scores on a set of profile attributes into one of the five groups based upon the weighted linear combination of those scores. The optimal weights are found by sampling from the five groups and computing the regression equation that best predicts group membership. This equation is then used for classify new individuals into the groups based upon the rounded values of *g.* However, whilst this works well where there are just two possible values for *g,* and is in fact the basis for Fisher's linear discriminant function analysis, multiple regression analysis assumes a continuous valued dependent variable and tends to produce poor classifications of discrete values when more than two outcomes are expected. In this case, we would use multiple linear discriminant function analysis. This retains the notion of a linear set of predictors, but works on the basis of producing more than one equation to predict group membership. In fact, an optimal prediction equation is produced for each group or category of the dependent variable. New cases are assigned to a category based upon the highest classification score computed from each equation (Nunnally and Bernstein, 1994, Tabachnick and Fidell (2001).

If we do not wish to assume that the predictor attributes are normally distributed (one of the assumptions in the multiple regression/discriminant function analysis model), we might use a model based upon the logistic distribution which makes no assumption about the distribution of predictor variables. For example, ordered or multinomial response logistic regression is a nonlinear prediction model which will yield a probability of outcome for each of a set of outcome categories. For dichotomous outcome dependent variables such as success/failure, high performer v low performer etc., only one classifier equation is computed. In our example of the linear model above, there are (5-1) = 4 equations computed. Each equation determines the probability of an individual being in a performance category above the immediate lowest one. For example, the equation probabilities can be read as:

- Eq.1: the probability that individual x is above performance category 1
- Eq.2: the probability that individual x is above performance category 2
- Eq.3: the probability that individual x is above performance category 3
- Eq.4: the probability that individual x is above performance category 4

Which, when the attribute values are entered into each equation, might look like:

- Eq.1: the probability that individual x is above performance category 1 (0.01)
- Eq.2: the probability that individual x is above performance category 2 (0.23)
- Eq.3: the probability that individual x is above performance category 3 (0.52)
- Eq.4: the probability that individual x is above performance category 4 (0.03)

What we conclude here is that individual x is most likely to be a member of performance category 4. The basic equation for this model is:

$$P(x > j) = \frac{e^{(b_{0j} + b_{1j} p_1 + b_{2j} p_2 + b_{3j} p_3 + \ldots + b_{nj} p_n)}}{1 + e^{(b_{0j} + b_{1j} p_1 + b_{2j} p_2 + b_{3j} p_3 + \ldots + b_{nj} p_n)}} \qquad (2.5)$$

where
(x > j) = the probability that individual x is located in a group > j (where j= 1 to 5)
b_{oj} = an equation constant for group j
$b_{1..n}$ = profile attribute weights for group j
$p_{1..n}$ = profile attribute scores for individual x

which, when expressed in a linear form by taking logs is:

$$\mathrm{Log}_e\left(\frac{P(x > j)}{1 - P(x > j)}\right) = b_{0j} + b_{1j} p_1 + b_{2j} p_2 + b_{3j} p_3 + \ldots + b_{nj} p_n \qquad (2.6)$$

where $\mathrm{Log}_e\left(\frac{P(x > j)}{1 - P(x > j)}\right)$ is the logit or log odds of an individual being in group (*j*+1); which is given by the ratio of the probability of being in a lower group divided by the probability of being in the group above, where *j* = group [1 …4]. As the reader will see, this kind of multinomial regression is in fact the logistic analogue of multiple discriminant function analysis as described above.

These three techniques, linear regression, multiple linear discriminant analysis, and logistic regression form the backbone of the essentially linear function classifiers. They have the advantage of statistical elegance, simplicity, and the capacity to refer to hypothetical sampling distributions for estimates of statistical significance for parameter values. However, this very simplicity sometimes mitigates against their use for real-world problems where a high level of classification accuracy is the aim. Moreover, few samples used in profiling research and applications can be said to be random samples of any kind of “population” whose scores may be distributed according to some specified hypothetical distribution. The utility of any significance test is thus highly questionable in these scenarios.

2.C. Algorithmic Classifiers

Algorithmic classifiers approach the classification problem from a completely different perspective than the statistical classifiers outlined above. Rather than fit data using predefined (generally linear) models to a data, an algorithmic model solves the problem using a set of simple rules or heuristics which when followed, yield a solution or classifier that maximizes cross-validated predictive/classification accuracy. No assumptions need be made about the distribution of the data, and there is no requirement that relationships amongst variables need be linear or even continuous over the range of any variable. In short, there is no requirement or indeed necessity for an *a priori* stochastic data model to be utilized. This is the reverse of the models in section 2.B above, and indeed for all conventional multivariate statistical methods. Possibly the most striking examples of computational algorithms for pattern and complex systems feature generation are found within Cellular Automata (Wolfram, 2002) and Genetic Algorithms (Mitchell, 1998). However, within the classifier construction domain, two methodologies are dominant. These are decision trees and neural nets.

2.C.1. Decision Trees.

The concept and major algorithms for a constructing a decision tree were introduced by Breiman, Friedman, Olshen, and Stone (1984) with the Classification and Regression Tree algorithm (CART), and Quinlan (1986, 1993) with the ID3 and C.4.5 classifier algorithms. A decision tree is a structure built from a series of decisions that aim to maximize classification accuracy of two or more outcome classes, levels, or measures. The analogy with the form of a tree is what gives these structures their name. For example, let us take a simple problem, attempting to classify high and low rated job performers using two attributes: Self-Report Job Stress and Self-Report Satisfaction with Managerial Style. The data table for this problem is shown in Table 3 below.

Table 3: Data table for Decision Tree example, where we are trying to predict job group membership of high and low job performance from the ratings on two attributes.

Job Performance Rating	Self-Report Job-Stress	Satisfaction with Managerial Style
Low	8	5
Low	7	5
Low	6	3
Low	3	4
Low	8	7
Low	7	5
Low	5	5
Low	7	3
Low	6	6
Low	7	5
High	2	6
High	5	7
High	4	8
High	5	6
High	3	6
High	4	7
High	3	7
High	5	9
High	3	7
High	2	8

We have 10 individuals in each of our criterion groups of high and low rated performers. Analyzing the data using decision tree analysis results in the structure displayed in Figure 10.

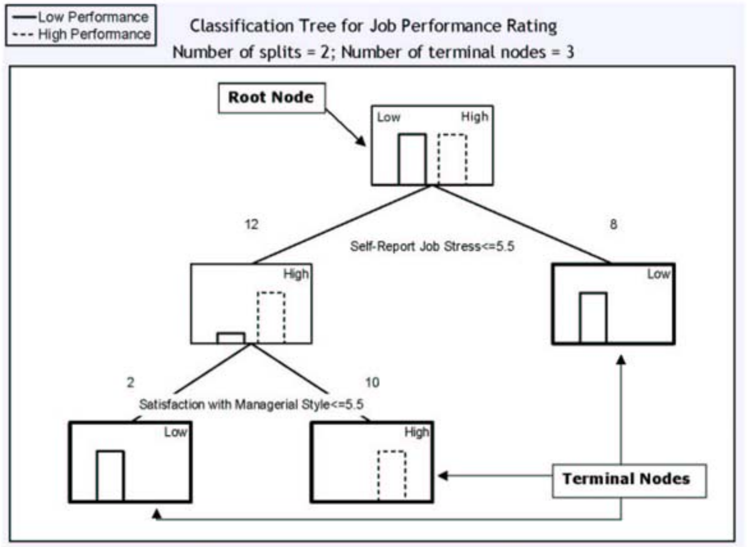

Low Performance
High Performance
Classification Tree for Job Performance Rating
Number of splits = 2; Number of terminal nodes = 3
Root Node
Low
High
12
8
Self-Report Job Stress<=5.5
High
Low
2
10
Satisfaction with Managerial Style<=5.5
Low
High
Terminal Nodes

Figure 10: a Decision tree for the data in Table 5, classifying individuals who show high or low job performance ratings on the basis of their self-report job stress and satisfaction with their boss' managerial style.

The first box in this figure, labelled "Root Node" shows the initial starting position for our decision tree – with the High and Low Performance group frequencies shown as histograms within the box, with the high group shown as a solid black line, and the low group as a dotted line. The algorithm then searches all the independent variables for the one that provides an optimum separation of the members (cases) of the two groups into their respective classes (high and low performance). In our case, this is found to be "Self Report Job Stress". Because we are dealing with ordinal variables here (ratings and scores), the split can be made at that value which maximally separates the two groups. This is a value of 5.5. Above this value, cases are classified as Low Performance members, below this value, they are classed as High Performance cases. We see from figure 13 that 8 cases are classed as low performers, and 12 as high performers. The decision produces what are called *nodes* – each node "branching" from the *root node*. The 8-case Low Performance node is also referred to as a *terminal node*, which in this particular example means no more classification is possible from this node as all cases in this "branch" of the tree have now been classified into one class within the node. Indeed we can see that 8 Low Performance cases have been correctly classified by the decision statement. However, the other node includes two Low Performance cases which have been incorrectly classified by this single decision as High Performance cases, along with the 10 "actual" High Performance cases. So, the algorithm searches for the best variable which will accurately discriminate or "*split*" the cases at this node, it finds that scoring <= 5.5 on the variable "Satisfaction with Managerial Style" discriminates perfectly between Low and High Performers at this node. So, two more nodes are created which contain the two Low Performance cases and the 10 High Performance cases. These nodes, are also referred to as *terminal nodes* because no more classification is possible here. However, a node may be "terminal" even when cases from two classes are required to be split within the node, but where a constraint on how "deep" the tree may extend has forced termination of the tree at that node.

From this simple tree we can see that classification accuracy is 100%. That is, from the two decisions made, all cases can be accurately classified into High and Low Performers. This tree in fact forms our profile for predicting in future, potential high and low performers from the basis of Job Stress scores and boss' Managerial Style ratings. The final step in decision tree analysis is thus to create the decision statement-block which forms the basis for future case allocation. This requires "parsing" the tree, which in reality means simply creating the decision block required to be slotted in to a new-case scoring program or computer-based allocation system. For example, for the current tree, the decision block constituting the classifier profile is simply:

```
If Self-Report Job-Stress > 5.5 then                              {Decision #1}
    Membership Class  = Low Performance
Else If Satisfaction with Managerial Style <= 5.5 then            {Decision #2}
    Membership Class = Low Performance
Else
    Membership Class = High Performance
```

In terms of classification accuracy, this tree is 100% accurate. If we fit a multiple linear regression model of the form shown in equation 2.4 above to the data, with the two independent variables of Self-Report Job Stress and Self-Report Satisfaction with Managerial Style used as predictors of Job Performance group (high-low), we obtain an R^2 of 0.77 (77% explained variation), but, if we convert the predicted values into rounded integer format, we also obtain 100% classification accuracy. This simple example acts as a warning to those who use linear model R^2 values as measures of function accuracy when the aim is to classify rather than predict a continuous-valued dependent variable. Fitting a logistic model of the form shown in equation 2.5 also results in 100% classification accuracy.

Whilst the above example helps explains the basic terminology and logic of a decision-tree, it is perhaps helpful to see what a more realistic profile construction/design process actually involves. Our second example is drawn from a dataset of individuals from differing occupational groups who completed an assessment of their work-behaviour preferences (24 preference attributes with each measured over a 0 to 100 range). Figure 11 provides the median score vector group profiles for two of these employee groups, those individuals working in Human Resources (N=93), and those in Finance/accounting positions (103).

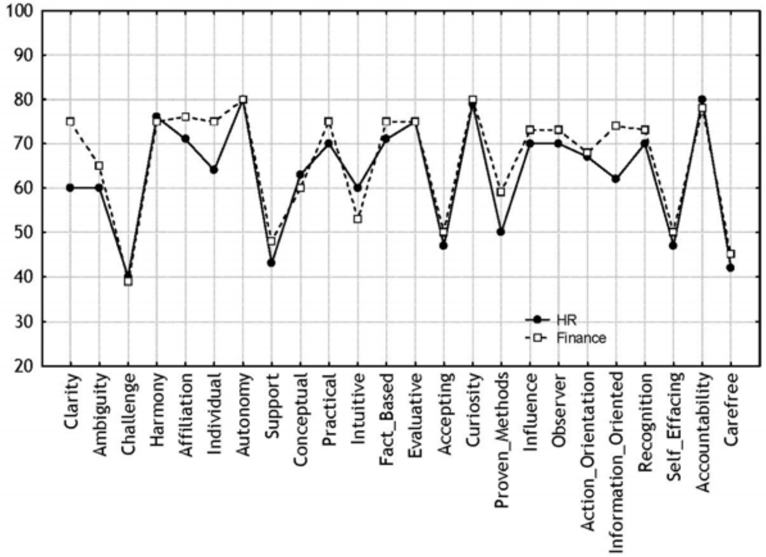


Figure 11: Score Vector Profiles using the median scores for two employee groups (HR = Human Resources) assessed on their preferences for work-based activities

However, unlike the methods expanded upon in Section 3 below, where we are interested in how near/distant a new individual's score profile may be to either profile so that we might for example advise them as to possible career options, a classifier approach is attempting to assign them to one class or the other. If we fit a multiple linear regression equation (equation 2.4) to these data, with 24 predictor variables, we obtain an R^2 of 0.15. As before, transforming the continuous predicted values into rounded class-value integers (1=HR, 2=Finance) for classification purposes yields a classification accuracy of 61%, but in looking carefully at the errors we see that the equation is biased in over-predicting finance cases and under-predicting human resource cases such that the false negative rate (where we predict a case as being from the finance class yet it is in fact from human resources) is 80%. In short, its apparent overall profile classification accuracy disguises the fact that it is very poor for the human resources class. If we turn to a logistic model as defined in equation 2.5 above, we find a classification accuracy of 60% (adjusting the class boundaries to reflect the prior class frequencies of 93 and 103 respectively). The error rates are however more balanced with false-positive and false negative rates each of 41% and 40% respectively. Moving now to the construction of a decision tree classifier, we have to make certain decisions regarding how to construct the tree in terms of how to select the optimal variable upon which to make a split and form a node, and when to stop splitting or growing the tree. For example, if we choose to use a splitting method based upon finding optimal values on a single variable that best discriminates between the classes with a "stopping rule" which states that if fewer than 2% of the observations are in one or the other class at a node, then no further splits will take place from that node, we obtain the following tree as shown in Figure 12.

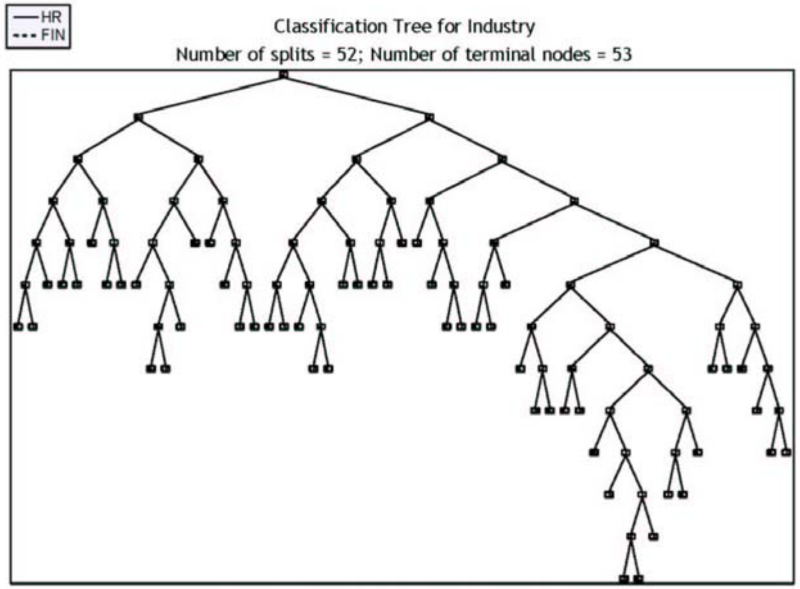

HR
FIN
Classification Tree for Industry
Number of splits = 52; Number of terminal nodes = 53

Figure 12:a decision tree for classifying two groups of employees (HR = Human Resources and FIN – Finance) on the basis of their work-behaviour preferences as assessed over 24 attributes. A 2% "fraction of objects" stopping rule is employed, with a univariate optimal split algorithm criterion.

This tree possesses 52 splits (decisions) and 53 nodes, its classification accuracy is 95%. At first glance this is a remarkable increase in predictive accuracy from the other methods we have used. But, as stated earlier, the capacity of algorithmic methods to over-fit data must not be underestimated. Unless care is taken to cross-validate these results using at least one of the five methods outlined in section 2.B, then such results might mistakenly be viewed as "excellent". This result used *V*-fold cross validation with *V*=3 folds, which is a clear demonstration that this method is not very sensitive to over-fitting bias. So, we use a more powerful method – the global/stratified cross-validation procedure with *V*=10 folds. This shows us that this tree is actually more likely to possess just 48% classification accuracy on new cases, with balanced false positive and false negative rates of 52%. This solution highlights the need to produce a tree with fewer decisions, whilst maximizing cross-validated classification accuracy. In short we need to "prune" the tree. So, we fix our fraction of objects say at 25%, which states that if fewer than 25% of the observations are in one or the other class at a node, then no further splits will take place from that node. We obtain the tree shown in Figure 13.

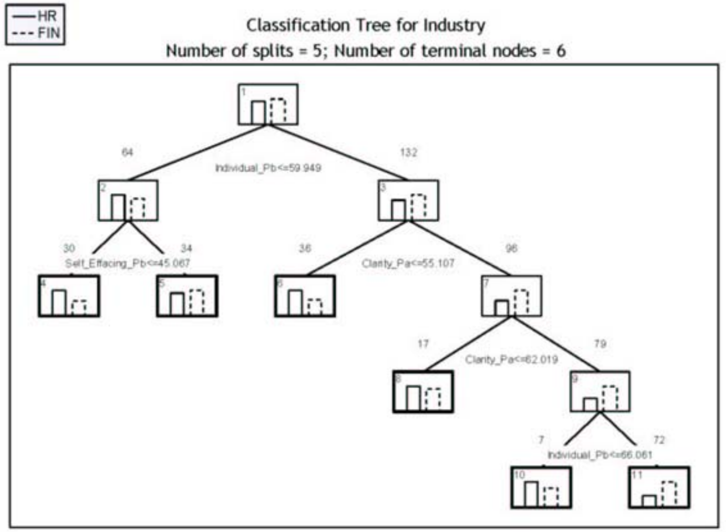


Figure 13: a decision tree for classifying two groups of employees (HR = Human Resources and FIN – Finance) on the basis of their work-behaviour preferences as assessed over 24 attributes. A 25% "fraction of objects" stopping rule is employed, with a univariate optimal split algorithm criterion.

This tree has far fewer splits/decisions than that shown in figure 15. Its *V*-fold (=3) cross validation classification accuracy is however, reduced to 62% (from 95%), whilst the Global *V*-fold (=10) classification accuracy is 55% (up from 48%). However, in algorithmic modelling, there are always varying ways of approaching data that may capitalize on features of the data or other algorithms so as to produce better cross-validated classification accuracy. So, we might try another splitting rule given we have ordered variable predictors (with a maximum measurement range between 0 and 100). One powerful method for doing this is to produce splits in the tree, not on the basis of single variable ranges, but on functions of several variables. A convenient methodology for this is linear discriminant function analysis. Each split is based upon a discriminant function cut-off, where the discriminant scale is constructed from all predictor variables, and where the chosen cut-off maximizes the discrimination between the two classes at a node. Each node thus requires its own discriminant function equation. I now choose this splitting criterion along with a fraction of objects stopping rule of 50%. The resultant tree is shown in Figure 14.

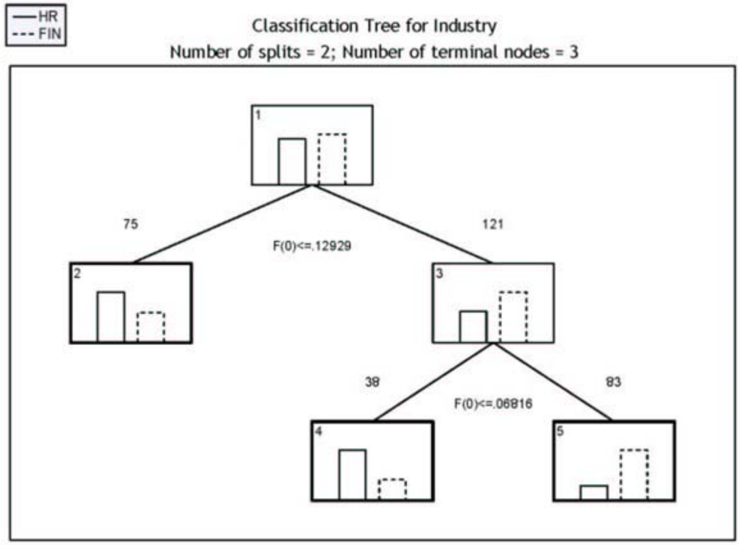

HR
FIN
Classification Tree for Industry
Number of splits = 2; Number of terminal nodes = 3
1
75
121
F(0)<=.12929
2
3
38
83
F(0)<=.06816
4
5

Figure 14: a decision tree for classifying two groups of employees (HR = Human Resources and FIN – Finance) on the basis of their work-behaviour preferences as assessed over 24 attributes. A 25% "fraction of objects" stopping rule is employed, with a linear discriminant function optimal split algorithm criterion.

This tree has fewer splits/decisions than that shown in figure 16 (just two now). Its *V*-fold (=3) cross validation classification accuracy is 70% whilst the Global *V*-fold (=10) classification accuracy is 56%. So, we have obtained marginally better cross-validated prediction with markedly fewer decisions, although those decisions are now made by scoring each new case on two 24-variable discriminant functions in order to produce the classification score for each decision. Whilst we could (and should) continue trying various methods for tree splitting, pruning, and cross-validation, I hope it has become more apparent to the reader that algorithmic modelling is quite unlike statistical modelling in that a greater capacity exists for extracting the maximum possible information from data whilst cross-validating the results achieved. Further, in the cases above, the simple *V*-fold cross-validated solutions for a decision tree outperformed the non-cross-validated conventional regression methods. The significance of this fact should not be lost on the reader.

2.C.2. Neural Nets or Artificial Neural Networks (ANN)

A neural net or network draws its name from the basis of this kind of approach to constructing predictive solutions of outcomes, based upon a biophysical model of a neuron (Koch, 1999) and how the brain's use of neurons in "networks" allows it to process information and ultimately solve problems. The initial proposition that an artificial network might be constructed which could simulate some aspects of neural interconnectivity and form the foundation of a system which might make logic-based decisions was a conjunction of two lines of investigation. The first was in the area of mathematical logic and axiomatic formalisation of the rules for arithmetic and calculation, the second in the field of neurophysiology (Arbib, 1995). It was McCulloch and Pitts (1943) who published what might be considered the foundational paper in this area, making explicit reference to the neuron as a threshold-based logical unit, and showing that a system of interconnected artificial neurons could form the basis of a Turing machine. It was also around this time that the field of cybernetics (feedback-contingent control systems) was starting to attract the interest of engineers and scientists (Rosenblueth, Wiener, and Bigelow (1943)), with the notion that the brain was itself a biological control system rather than the older view of it being a "reflexive" system. That is, the brain was fed inputs, it would process these, produce responses, which in turn produced new inputs which in turn would provide feedback for the original responses such that the brain might adjust its new output on the basis of its own previous output and subsequent input. The neural network was a mechanism by which such a control system might be implemented.

Basically, a neuron has two states – it is either quiescent or it produces a signal; the basis of a Boolean logic off/on decision. From biological studies of neural tissues, individual neurons, and the properties of such neurons and their interconnectedness, five assumptions/propositions are made about neural computations which form the basis of many of the algorithms embodied in artificial neural networks (McLeod, Plunkett, and Rolls (1998):

1. *Neurons integrate information*. That is, a neuron integrates information from a set of multiple inputs (other neurons) and passes a signal to other neurons.
2. Neurons pass information about the level of their input. Not only does a neuron integrate information, it also encodes information about the level of magnitudes of its input via its firing ("on/off") rate. In an ANN, this adaptive firing rate is

encoded as a single "activity" magnitude which varies in response to the perceived input level.

3. *Brain networks are "layered"*. That is, banks or "layers" of neurons may be interconnected with one another. Entire arrays of interconnected neurons may themselves be connected together in order to form a composite processing network. Layers not immediately connected to the initial inputs or the final outputs are called "hidden layers". Figure 15 shows a schematic of a neural network with one hidden layer.

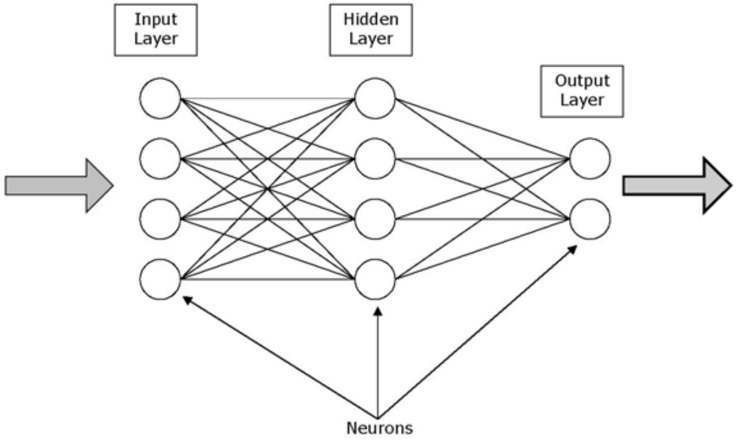


Figure 15: a schematic of a multilayer neural network with one hidden layer.

4. *Neurons are influenced by the strength of connections between them.* Essentially, this proposition states that the influence of one neuron on another is affected by the strength of connection between them. Propensity to fire in a biological system can be controlled by the abundance/depletion of neurotransmitters within the synapse (the area of connection of two neurons) and properties of the membrane of either neuron. Within an ANN, this influence is modelled as a weight function applied to the connection between two neurons.
5. *Learning is achieved by changing the strength of connections between neurons.* This is the crux of an ANN. In order to produce "learning" as for example in order to develop a classifier, the network has to adjust the weights (strength of connections) between neurons. It does this by starting with a set of random weights, producing some output from the inputs, then if the output is not correct, re-adjusts the weights according to one or more algorithmic principles, then re-produces the new outputs from the inputs, and so on until some criterion accuracy is reached. This is the essence of the iterative "*backpropagation*" algorithm used in so many network training applications. The algorithms, network types, and fundamental training and learning issues within ANN generation are simply too numerous and complex to be summarised in what is essentially a chapter on person-target profiling methods. The interested reader should consult references such as Hastie, Tibshirani, and Friedman (2001), Arbib (1995), and McLeod et al (1998). However, neural nets are a key methodology for producing classifiers for profiling groups of individuals, as evidenced by the recent study from Sommer et al (2004) for airline pilot selection. In order to allow the reader to judge the classifier effectiveness of a neural network, I have fit (using STATISTICA Neural Networks software) what is known as a multilayer feed-forward perceptron network to the complex 24 work-behaviour preference problem above, attempting to classify members of two employee groups; human resources and accounting/finance, on the basis of the network formed using 24 predictor inputs. This model was initially fit onto a training sample (N=140), then a verification sample (N=56). Classification accuracy (computed over the verification sample) of this model was 71%, using a hidden layer with 8 "hidden" processing units. However, in order to fully test such a model using so few cases, one might use several bootstrap samples in order to construct multiple training and verification samples. However, the capacity to maximize cross-validated classification accuracy via algorithmic methods is the very reason why these models are becoming so prevalent in the prediction/classifier, and now profiling applications.

3. Score Vector Comparison Coefficients

Returning to our popular score vector profile as outlined in section 2.A above, it is useful to detail the main coefficients used for expressing a comparison in terms of similarity or distance from the person to a one-dimensional target score profile vector. Broadly speaking, there are two types of coefficient, those based upon covariance between profile vectors, and those on Euclidean distance.

3.A.1. Covariance Based Coefficients

Pearson Correlation

Given a vector of person scores p_i, and a vector of target scores t_i, where i ranges from 1 to N profile attributes, the pearson correlation r =

$$r = \frac{\text{cov}_{pt}}{s_p \cdot s_t} \quad \text{where } N\text{=the number of paired observations} \tag{3.1}$$

where cov_{pt} is the covariance between variables p and t.

The coefficient varies between +1.0 and -1.0, with +1.0 indicating maximum similarity, -1.0 = maximum inverse similarity, and 0.0 indicating no relationship whatsoever. The coefficient is sensitive only to shape, as both elevation and scatter are equated by the deviation and normalisation by standard deviation. The definitions of these terms have been given above in section 1, and in equations 1.1, 1.2, and 1.3.

Intraclass Correlation (ICC)

This coefficient is computed specifically for the situation where all observations are considered to be from the same class. This distinguishes it from the pearson correlation which assumes observations are acquired from different variables (Norman and Streiner, (2000)). A class for example might be a pair of twins, an individual measured across two occasions, or as in our case, a profile of scores acquired from two individuals, or a an individual being compared to a target profile. The class in each cases being the score profile. Two models for the coefficient are utilized in person-target profiling, corresponding to models 2 and 3 from Shrout and Fleiss (1979).

Model 2: We assume the profiles are randomly selected (sampled) from some population of profiles (a two-way random effects model). That is, the target profile is considered an instance from some population of all possible profiles, as is the profile from an individual (which is to be compared to the target). This means that we consider that we have sampled an individual's profile from some population of possible profiles. Likewise the target profile…

$$r_{icc} = \frac{MS_p - MS_{res}}{MS_p + (n_r - 1) * MS_{res} + \left(\frac{n_r * (MS_r - MS_{res})}{n_e} \right)}$$

$$\begin{aligned} \text{where } MS_p &= \text{Between profiles mean square} \\ MS_{res} &= \text{Residual (interaction) mean square} \\ MS_r &= \text{Between profile elements (scale scores) mean square} \\ n_r &= \text{The number of profiles (=2)} \\ n_e &= \text{The number of elements/scales in a profile} \end{aligned} \tag{3.4}$$

Model 3: We assume the profiles are the "population" profiles for a target and individual to be compared to that target profile (a two-way, fixed-effects model). It is assumed that these are the only two profiles - they are **not** "samples" from some theoretical population of possible target and individual profiles (in that the target is unique, and the individual profile is considered unique).

$$r_{icc} = \frac{MS_p - MS_{res}}{MS_p + (n_r - 1) * MS_{res}}$$

$$\begin{aligned} \text{where } MS_p &= \text{Between profiles mean square} \\ MS_{res} &= \text{Residual (interaction) mean square} \\ n_r &= \text{The number of profiles (=2)} \end{aligned} \tag{3.6}$$

The intraclass coefficient varies between +1.0 and -1.0, with +1.0 indicating maximum similarity, -1.0 = maximum inverse similarity, and 0.0 indicating no relationship whatsoever. The coefficient is differentially sensitive to elevation, scatter, and shape, depending upon the model chosen.

Congruence and Alienation Coefficients

The congruence coefficient is normally encountered in the domain of factor similarity (comparing factors from a factor analysis across samples). It is, in effect, a pearson correlation computed using unstandardized variables. Overall and Klett (1972, pp. 392-393) also refer to this coefficient as the raw vector product coefficient (contrasted with the normalized vector product coefficient - i.e. the Pearson correlation). Guttman (1981) derived exactly the same coefficient for computing the similarity between derived and observed distances within multidimensional scaling (MDS); he called it the monotonicity coefficient (mu or μ). Coxon (1982, p. 89-90) also shows that μ is directly related to measures of stress (in multidimensional scaling), as well as Gutmman's coefficient of Alienation (K). This alienation coefficient is a measure of the "unexplained" variation between the comparison and target profiles. So for example, if a congruence coefficient of 0.99 is computed, Guttman's K = 0.14. A brief discussion of these issues can be found in Borg and Groenen (1997, pp 203-204). In relation to this issue, they note that μ takes on values close to 1.0 even if the MDS solution is "far from perfect". It was for this reason that Guttman converted μ to K (the coefficient of Alienation), in order to expand the indicative range. If we subtract K from 1.0, we can obtain a measure of "similarity" (lack of alienation) that is based upon the expanded range of K.Of interest is that Gorsuch (1983, p. 285), in relation to the use of the congruence coefficient for comparing factors, states "Occasionally, coefficients of congruence can give ridiculous results. Pinneau and Newhouse (1964) have pointed out that the index is highly influenced by the level and sign of loadings. Factors whose loadings are the same size will, of necessity, have a high coefficient of congruence even if the patterns are unrelated".

$$r_c = \frac{\sum_{i=1}^{N} p_i \cdot t_i}{\sqrt{\sum_{i=1}^{N} p_i^2} \cdot \sqrt{\sum_{i=1}^{N} t_i^2}} \equiv \text{Guttman's } \mu \text{ ..... where}$$

$p =$ the comparison profile scores and

$t =$ the target profile scores (for $i = 1..N$)

(3.7)

Note that if p and t are expressed as deviation scores around their respective means

$$r_c = \frac{\sum_{i=1}^{N} (p_i - \overline{p}) \cdot (t_i - \overline{t})}{\sqrt{\sum_{i=1}^{N} (p_i - \overline{p})^2} \cdot \sqrt{\sum_{i=1}^{N} (t_i - \overline{t})^2}} \equiv \frac{\operatorname{cov}_{xy}}{s_x \cdot s_y} \equiv \text{Pearson r}$$

$$Alienation(K) = \sqrt{\left(1 - \mu^2\right)} \equiv \sqrt{\left(1 - r_c^2\right)} \qquad (3.8)$$

The congruence coefficient varies between +1.0 and -1.0, with +1.0 indicating maximum similarity, -1.0 = maximum inverse similarity, and 0.0 indicating no relationship whatsoever. The congruence coefficient tends to be sensitive only to shape, although the 1-Alienation coefficient does possess some sensitive to elevation and scatter differences.

3.A.2. Distance Based Coefficients

The second most popular kind of profile matching coefficient is that based upon the concept of "Euclidean distance" or some kind of "difference score" measure of person-target distance/similarity.

Euclidean Distance

The Euclidean metric is that which corresponds to everyday experience and perceptions. That is, the kind of one, two, and three-dimensional linear metric world where the distance between any two points in space corresponds to the length of a straight line drawn between them. The formula for this distance (given pairs of points in 1 dimensional linear space, as in a typical scale score profile comparison scenario is:

$$d_{ij} = \sqrt{\sum_{i=1}^{N} \left(p_i - t_i\right)^2} \qquad (3.9)$$

where

$p_i =$ the profile score to be compared on scale i of N scales.

$t_i =$ the target profile score on scale i of N scales.

Euclidean distance has no obvious bound value for the maximum distance, merely one for absolute identity. Its range of values vary from 0 (absolute identity) to some maximum possible discrepancy value which remains unknown until specifically computed. So. profiles can be compared amongst one another in terms of their Euclidean distance, given they are expressed in the same metric range as one another - but the distances permit only the relative ordering of profiles amongst one another, without regard to what the distance values imply in terms of absolute disparity. Put simply, Euclidean distance varies as a function of the magnitudes of the profile observations. Assume I compute a Euclidean distance of 26.19. If I divided every profile score by 10, and recomputed the Euclidean distance between the profiles, I would now obtain a distance value of 2.619. So, raw Euclidean distance is acceptable only if relative ordering amongst a fixed set of profile attributes is required. But, even here, what does a figure of 26.19 actually convey? If the maximum possible observable distance is 30, then we know that the profile being compared is almost maximally discrepant. But, if the maximum observable distance is 1000, then a value of 26.91 indicates near identity. The fact of the matter is that unless we know the minimum and maximum possible values for a Euclidean distance, we can do little more than rank profiles in terms of their relative similarity, without ever knowing whether any or them are actually similar or not to a target profile. Note also here that Euclidean distance is "blind" to the direction of the differences between a target profile and comparison profile. That is, by taking the square of the difference, it effectively removes any information about the direction (above or below target) of such a difference. This can matter greatly in some profiling applications where being high or low on an attribute implies a completely different meaning about "lack of fit".

Normalised Euclidean Distance

The most obvious solution for expressing Euclidean distance in a standardized metric where 0.0 = absolute identity through to 1.0 = maximum dissimilarity is to normalize the raw Euclidean distance, dividing it by the maximum possible Euclidean distance observable between any comparison target score and the target profile score. Using the scores on the target profile, we can compute the maximum distance observable between any two profiles, given the maximum and minimum possible observable values for each attribute being compared, *relative* to the target value. That is, the maximum discrepancy between a target and comparison score is the largest possible difference between a profile target score on an attribute and the minimum or maximum possible score on that profile. The target profile attribute scores represent the minimum possible values for a comparison match as differences from these to either the minimum or maximum observable scores represents the range of dispersion. For example, take a profile consisting of three attributes 1, 2, and 3. The target values for these attributes are 5, 7, and 15. Two individuals are compared to this target using Euclidean and normalized Euclidean distance calculations. The relevant data are shown in Table 4 below.

Table 4: Data for Euclidean distance calculations, with two individuals' score profiles compared to a target score vector profile consisting of three attributes.

Attribute	Person 1 Scores	Person 2 Scores	Target Profile Scores	Minimum Possible Attribute Score	Maximum Possible Attribute Score
1	6	8	5	0	14
2	5	12	7	0	18
3	10	20	15	0	20

The raw Euclidean distance is calculated using formula 3.9 for both persons, yielding values of 5.48 and 7.68 for person 1 and 2 respectively. In order to compute the normalized distance, we need to calculate the maximum possible discrepancy for each attribute (1 to 3), *relative to the target value on each attribute*. So, for attribute 1, the maximum possible discrepancy will be between 5 (the target) and 14 (the maximum possible attribute score) = 9. For attribute 2, it will be between 7 and 18 = 11. For attribute 3, it will be between 0 and 15 = 15. Thus, our maximum attribute distances are 9, 11, and 15 respectively. We sum the square of these and take the square root of this sum to yield the maximum possible Euclidean distance between the target and any set of profile scores. Then, each person's raw Euclidean distance is normalized into a 0-1 range by this maximum Euclidean distance constant. So, in our example, the sum of the squared maximum attribute differences is 427, whose square root is 20.664. This is our normalizing constant. Thus person 1's normalized Euclidean distance is 5.48/20.664 = 0.27. Person 2's is 0.37. It is useful to express distance in terms of similarity, as this makes it easier to compare coefficients. Thus we simply subtract our coefficients from 1.0 to express the distance in a similarity metric, so person 1 and person 2 are assigned profile similarity measures of 0.73 and 0.63 respectively. The coefficient is differentially sensitive to elevation, scatter, and shape.

The Profile Similarity Coefficient (r_p)

The profile similarity coefficient (Cattell, 1969) or what has also been called the *pattern similarity coefficient* (Cattell, Coulter, and Tsuijoka, 1966; Cattell, Eber, and Tatsuoka, 1970; Cattell, 1978) was first introduced by Cattell in 1949. It was designed by Cattell (*taken from: Cattell, Coulter, and Tsuijoka, 1966, p.296*) to:

1) take into account the metric and number of dimensions comprising the profiles to be compared
2) compare the coefficient with the magnitude to be expected by chance
3) provide a convenient function which behaves e.g. as regards distribution, in essentially the same general way as a pearson coefficient, varying from +1.0 indicating complete agreement between profiles to 0 for no relation, and -1.0 for complete inverse relation.

$$r_p = \frac{E_k - \sum_{i=1}^{k}\left(z_{p_i} - z_{t_i}\right)^2}{E_k + \sum_{i=1}^{k}\left(z_{p_i} - z_{t_i}\right)^2} \qquad (3.10)$$

where

$z_{p_i} =$ the standard score of a comparison profile attribute i of k elements (scores)

e.g. $z_{p_i} = \dfrac{\left(p_i - \overline{p}\right)}{s_p}$ where

$s_p =$ the standard deviation of the comparison profile scores

Likewise for the target profile scores.

$E_k =$ the chance-expected sum of squared deviations between two profiles of k elements

E_k is defined by considering the variance of the difference between two variables ...

$\sigma^2_{p-t} = \sigma^2_p + \sigma^2_t$ .... and the expected value of a chi-square variable (the median value in this case)

Given a chi-square variable is the sum of squared values of a set of k standard scores, then the median value corresponds to the 50th percentile of a distribution of all such possible scores, for $df = k$.

This value multiplied by the sum of variances of the two sets of profile scores provides the value of E_k for any particular profile comparison that varies in terms of k attributes.

So: for **standard scores** with an SD and variance of 1.0, the value of E_k is:

$$E_k = \chi^2_{50} \cdot \sigma^2_p + \sigma^2_t = \chi^2_{50} \cdot \left(1+1\right) = 2 \cdot \chi^2_{50}$$

Given the median χ^2 for any $\mathrm{k} \approx \mathrm{k}$, then this is why Cattell (1969, 1978) sometimes defines it as:

$$E_k = 2 \cdot k$$

whereas for **sten scores** (with an SD of 2.0, and a variance of 4), the value of E_k is:

$$E_k = \chi^2_{50} \cdot \sigma^2_p + \sigma^2_t = \chi^2_{50} \cdot \left(4+4\right) = 8 \cdot \chi^2_{50} = 8 \cdot k$$

The meaning of an r_p of +1.0 is that two persons or patterns have exactly the same profiles and fall on the same point in multidimensional space. A value of 0 indicates that they fall as far apart as we would expect for any two points taken at random. A value of -1.0 means that they are at opposite ends of the distribution. Since the ends of a distribution are ill-defined and asymptotic, the value -1.0 is in actual practice approached but never quite reached, and there is in consequence a small asymmetry (positive skewing) in the distribution of r_p about its median value of 0. The coefficient is sensitive only to shape, as both elevation and scatter are equated by the deviation and normalisation by standard deviation. In order to make the r_p

sensitive to elevation and scatter, we have to compute the r_p using the actual raw data values themselves rather than use their standardized form in equation 3.10. To do this requires calculating the correct chance-expected squared deviation difference between two such profiles E_k, replacing the standardized variable standard deviations with the raw score counterparts.

Importantly, this coefficient provides a significance test for its value - the null hypothesis being that a coefficient observed of size X has been sampled from a population distribution where the mean value is 0. Put another way, the comparison profile is more similar to the target profile than expected by chance alone. Horn (1961) provides the formula derivation and tables for such a test.

Kernel Distance Coefficient

My design of this coefficient was based upon the idea that a distance function could be *shaped* in such a way that if the simple arithmetic unsigned difference between a person's attribute value and the target attribute value was computed to be within a certain range, then the computed distance should reflect a very small distance, almost regardless of the actual distance. But, as that distance grew larger, then the computed distance should be accelerated in size. In short, an "inertial" effect was aimed for – translated into a distance metric. The "shaped distance" design is in direct response to a need to "take control" or "shape" the distance function between two objects in space, as a function of their distance. So, small disparities between two profile scores would have little effect on the distance calculation, but, as the disparity grew greater, so would the distance be degraded in an accelerated fashion. The function chosen to achieve this was the normal distribution curve equation, with distance degradation acceleration controlled by the standard deviation parameter and the target profile value represented by the mean of the distribution function. Within the area of computational data smoothing and trend analysis, this kind of "inertial" effect is known as kernel density smoothing (Hastie et al, 2001). It has been applied here to permit the control of a distance function rather than to smooth data trends. As the distance between a target score value and a comparison score value increases, so the distance is accelerated between them once the distance moves beyond a region of little change (near the peak of such a distribution). The parameter controlling the "plateau" effect is the standard deviation of such a distribution; likewise, the acceleration in distance at larger values. For example, figures 16 and 17 below show the distance curves with standard deviations of 10 and 30 for five different target profile score values of 10, 20, 35, 50, and 75. The curves are constructed by systematically varying comparative scale score values (those score values to be compared to the fixed target) from 0 through 100 in 1-unit steps.

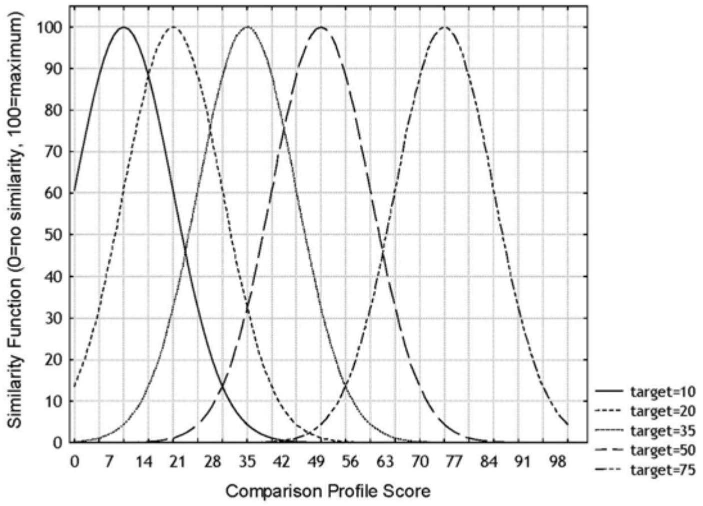


Figure 16: A kernel distance weighting function with standard deviation = 10. Comparison scores ranging between 0 and 100 are compared to five target scores of 5, 10, 35, 50, and 75.

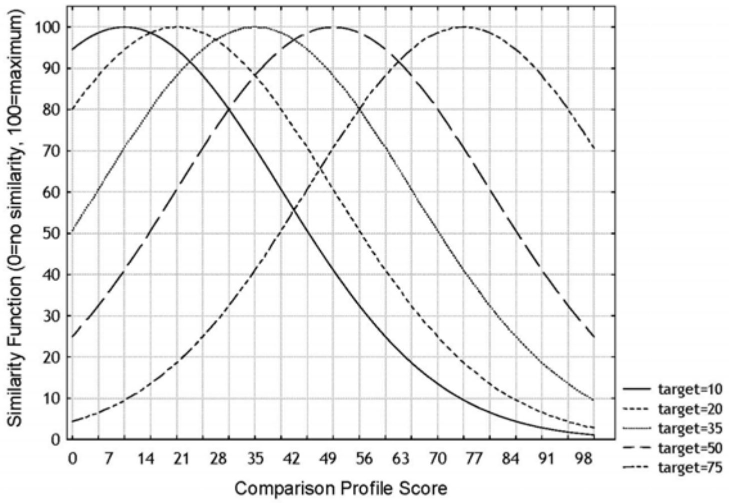


Figure 17: A kernel distance weighting function with standard deviation = 30. Comparison scores ranging between 0 and 100 are compared to five target scores of 5, 10, 35, 50, and 75.

The formula for this coefficient is:

$$KDC\_score = \frac{\sum_{i=1}^{N}\left[\frac{1}{s\sqrt{2\pi}}\,e^{-\left[\frac{(p_i-t_i)^2}{2s^2}\right]}\right]\cdot\left(100\cdot\left(s\cdot\sqrt{2\pi}\right)\right)}{N} \qquad (3.11)$$

where s = standard deviation

p = the comparison score for an attribute

t = the target score for an attribute

N = the number of attributes in the target profile

It is constructed to yield a percentage similarity measure, with 0% indicating maximum dissimilarity and 100% being absolute identity. The coefficient is differentially sensitive to elevation, scatter, and shape.

3.A.3. The Expected Values for these Coefficients

Two datasets were used to construct observed profile comparison coefficient distributions. One was a sample of 2011 individuals who had completed Psytech International's Occupational Personality Profiler (OPP) questionnaire, and formed part of thei normative UK sample database of cases. The test provides raw scores ranging between 0 and 60 (although maximum possible values were different for each scale) on 10 personality scales. These data were used to investigate the likely mean, median, and inter-quartile range of each of the profile comparison coefficients listed above, computed using each person's profile both as a comparison and as a target score profile. That is, each person was compared to every other person in the dataset where one individual in the pair was selected as a target profile and the other the comparison profile; in all over 2 million (2,021,055) coefficient values were computed for each coefficient. A second dataset of 2011 cases was constructed with each case consisting of 10 integer-value scores constrained to range between 0 and 60, sampling from a random normal distribution with a mean of 30 and standard deviation of 10. The scores were assigned randomly to each case, producing an essentially completely random score dataset. The purpose of this second dataset was to investigate the expected distributions of these coefficients when using random data instead of data which contained systematic covariance relations and, by definition, a higher degree of between-case profile similarities. As with the first dataset, every case was compared to every other case with one being designated as a target profile and the other as a comparison. For the kernel distance coefficient, I used a value of 4.0 for the standard deviation.The results of these analyses are provided in Tables 5 and 6 below.

Table 5: Summary statistics for eight profile comparison coefficients computed over 2011 random data cases yielding 2,021,055 values per profile comparison coefficient. There are with 10 attributes per profile, each profile attribute score range is between 0 and 60. All coefficients are expressed as similarity measures (the nearer to +1.0, the greater degree the similarity).

Coefficient	Mean	Median	Interquartile Range
Pearson correlation	0.00	0.00	-0.24 / +0.24
Intraclass Model 2	-0.01	0.00	-0.23 / +0.22
Intraclass Model 3	0.00	0.00	-0.23 / +0.23
Congruence/Guttman’s μ	0.91	0.92	0.89 / 0.94
1-Alienation	0.60	0.60	0.54 / 0.66
1-Normalised Euclidean Distance	0.57	0.58	0.51 / 0.65
Cattell Profile Similarity	-0.06	-0.09	-0.19 / -0.04
Kernel Distance Coefficient	0.27	0.27	0.19 / 0.35

Table 6: Summary statistics for eight profile comparison coefficients computed over 2011 cases of personality questionnaire data (Psytech International’s Occupational Personality Profiler), yielding 2,021,055 values per profile comparison coefficient. There are with 10 attributes per profile. All coefficients are expressed as similarity measures (the nearer to +1.0, the greater degree the similarity).

Coefficient	Mean	Median	Interquartile Range
Pearson correlation	0.53	0.58	0.37 / 0.74
Intraclass Model 2	0.50	0.55	0.34 / 0.71
Intraclass Model 3	0.50	0.55	0.34 / 0.71
Congruence/Guttman’s μ	0.96	0.95	0.94 / 0.98
1-Alienation	0.75	0.75	0.70 / 0.80
1-Normalised Euclidean Distance	0.62	0.63	0.55 / 0.70
Cattell Profile Similarity	0.30	0.30	0.13 / 0.48
Kernel Distance Coefficient	0.43	0.43	0.34 / 0.53

Table 5 provides the coefficient data computed using the random dataset. As can be seen from this table, using the random data sets, the Congruence, Alienation, and 1-Normalised Euclidean Distance are all yielding coefficients that would ordinarily be taken as indicative of reasonable-to-excellent matches. These data show that these coefficients should never be used for profiling applications because of their propensity to report coefficients from entirely random data which would be indistinguishable from coefficients computed over real data. This fact is brought home in reviewing the results from Table 6, where we see that little changes for these coefficients. The others by contrast show good separation from the random to real datasets. These results demonstrate the importance of ensuring that prior to selecting any coefficient for person-target profiling, it is critical to establish the likely range of values for such a coefficient prior to its use. However, as the data displayed in Figures 18 and 19 indicate, even this task is not sufficient to assure coefficient behavioral consistency.

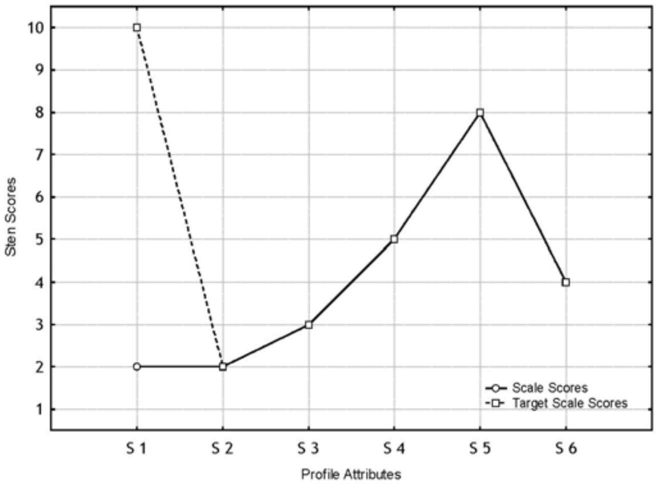


Figure 18: comparing two profiles composed of 6 attributes, whose values differ in only one attribute.

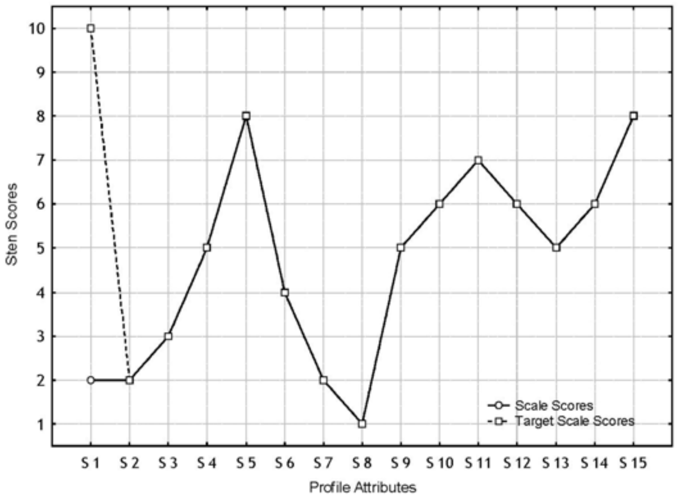


Figure 19: comparing two profiles composed of 15 attributes, whose values differ in only one attribute

Here I have created two sets of profile data using sten scores across 6 and 15 profile attributes respectively, in which the comparison and target profiles are exactly the same except for one scale which differs by 8 stens. Calculating the Pearson correlation, Pattern Similarity Coefficient, and Kernel Distance Coefficient (standard deviation = 1.5) on the data in Figure 18 yields values of 0.29, -0.20, and 83% respectively. Likewise for the data in Figure 19, the values are 0.63, 0.28, and 93% respectively. These figures serve as a warning that the pearson correlation and pattern similarity coefficient are extremely poor indicators of similarity in this kind of scenario. In Figure 19, all 14 scales have exactly equal scores – only the 15th attribute varies, albeit dramatically. Which coefficient of 0.63, 0.28, or 93% seems to best "capture" the similarity inherent in such data? This is a subjective choice – but one which will critically affect applications which rely upon a single coefficient to rank candidates. The overriding message I hope readers take from this brief analytical exposition is that constructing profiles is as much a design issue as it is an empirical issue, in that it requires careful computational simulation of expected values and boundary conditions, aligned to subjective visual perception of what might be said to constitute similarity and dissimilarity. After all, a profile containing 5 scores exactly the same and one discrepant score seems to be more similar than not, yet the Pearson correlation for these data was just 0.29, and the profile similarity coefficient was -0.20.

4. Conclusion

With respect to some of the analyses and coefficients considered above, especially with regard to shaping the distance metric and the two-dimensional profile distance metrics, it is important to mention a recent paper by Breiman (2001) describing the features and characteristics of what he calls the "two cultures" of statistics. The reason for this is that the kind of approach to profiling and the classification problem espoused in many sections of this chapter find much in common with the spirit and outlook of Breiman's second culture. The essence of this paper is given in the abstract…

"There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools."

I view computational profile construction and person-target profiling as very much a "second-culture" application field. This is evident in my construction of the kernel distance coefficient as well as in the two and three dimensional profiling applications outlined above. The aim is always to maximize cross-validated accuracy and not to simply use coefficients or other procedures with which others may be more familiar or which possess known hypothetical distribution characteristics and an accompanying set of assumptions, however desirable these may be in terms of statistical theory. Hence, the stress placed here, and by algorithmic modelers, on computational simulation of any chosen coefficient prior to its use as a means of fully understanding the coefficient behaviour and its expected values over a wide set of data. Simply knowing the boundary conditions for maximum agreement/disagreement is only partially useful – it's what happens "in between" these that matters, and the conditions under which the coefficient will yield apparently spurious albeit consistent results. Modern algorithmic-driven person-target profiling is in its infancy, but with the new tools, procedures, and capability to model most conditions and complex sample data for which a preferred solution might be used, the chances of making a magnitude difference in outcomes must surely be attractive to those areas of commercial practice which would benefit from profiling individuals. Person-Target profiling has the capacity to provide a very substantial return on investment to a company – but only if applied intelligently using computationally-intensive procedures for cross—validation and scenario-exploration. It remains to be seen which companies will seriously explore this capacity.

References

Arbib, M.A. (Ed). (1995) *The Handbook of Brain Theory and Neural Networks*. Cambridge, Massachusetts: MIT Press.

Borg, I., and Groenen, P. (1997) *Modern Multidimensional Scaling: Theory and Applications*. New York: Springer

Breiman, L. (2001) Statistical Modeling: the two cultures. *Statistical Science*, 16, 3, 199-231.

Breiman, L. Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees*. Monterey CA: Wadsworth.

Cattell, R.B. (1969) The profile similarity coefficient, rp, in vocational guidance and diagnostic classification. *British Journal of Educational Psychology*, 39, 131-142

Cattell, R.B. (1978) *The Scientific Use of Factor Analysis in the Behavioral and Life Sciences*. Plenum Press. ISBN: 0-306-30939-4

Cattell, R.B., Eber, H.W., and Tasuoka, M.M. (1970) *Handbook for the Sixteen Personality Factor Questionnaire*. IPAT.

Cattell, R.B., Coulter, M.A., and Tsujioka, B. (1966) The taxonometric recognition of types and functional emergents. In Cattell, R.B. (ed). *The Handbook of Experimental Multivariate Psychology*. Rand McNally.

Collins English Dictionary, 3rd Edition. (1991) Glasgow: Harper Collins.

Coxon, A.P.M. (1982) The User's Guide to Multidimensional Scaling. *London: Heinemann Educational Books*

Cronbach, L.J., and Gleser, G.C. (1953) Assessing similarity between profiles. *Psychological Bulletin*, 50, 456-473.

Dickson, M.A. (2001) Utility of no-sweat labels for apparel consumers: profiling label users and predicting their purchases. *The Journal of Consumer Affairs*, 35, 1, 96-119.

Efron, B. (1979) *Bootstrap methods: another look at the jackknife*. Annals of Statistics, 7, 1-26.

Gorsuch, R.L. (1983) Factor Analysis 2nd Edition. New York: Lawrence Erlbaum.

Groth-Marnat, G. (2003) *Handbook of Psychological Assessment 4th Edition*. New York: Wiley.

Guttman, L. (1981) Efficacy Coefficients for differences among averages. In I. Borg (ed). *Multidimensional Data Representations: when and why*. (p 1-10). Ann Arbor, Michigan: Mathesis Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Horn, J. L. (1961) Significance tests for use with rp and related profile statistics. *Educational and Psychological Measurement,* 21, 363-370

Koch, C. (1999) *Biophysics of Computation: Information processing in Single Neurons*. Oxford: Oxford University Press.

Lachenbruch, P.A. (1975) *Discriminant Analysis*. New York: Hafner Press.

Lazarsfeld, P.F. and Henry, N.W. (1968) *Latent Structure Analysis*. Boston: Houghton Mifflin.

Magidson, J. and Vermunt, J.K. (2004) Latent Class Models. In Kaplan, D. (ed) The Sage Handbook of Quantitative Methodology for the Social Sciences (p. 175-198). New York: Sage Publications.

McCulloch, W.S. and Pitts, W.H. (1943) A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.

Monnahan, J., Steadman, H.J., Appelbaum, P.S., Robbins, P.C., Mulvey, E.P., Silver, E., Roth, L.H., & Grisso, T. (2000) Developing a clinically useful actuarial tool for assessing violence risk. *British Journal of Psychiatry*, 176, 312-319.

McLeod, P., Plunkett, K., and Rolls, E.T. (1998) *Introduction to Connectionist Modelling of Cognitive Processes*. Oxford: Oxford University Press.

Meehl, P. (1995) Bootstrap Taxometrics. *American Psychologist*, 50, 4, 266-275.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press.

Mitchell, T. (1997) *Machine Learning*. New York: McGraw Hill.

Mosteller, F. (1971) The jackknife. *Review of the International Statistical Institute,* 39, 3, 1-6.

Muthén, B. O. (2001). Second-generation structural equation modeling with a combination of categorical and continuous latent variables: New opportunities for latent class/latent growth modeling. In L. Collins & A. Sayer (Eds.), New methods for the analysis of change (p. 291-322). Washington, D.C.: APA.

Muthén, L. K., & M Muthén, B. O. (2001). M*plus* User's Guide. Los Angeles, CA: Muthén and Muthén.

Norman, G.R. and Streiner, D.L. (2000) *Biostatistics: the Bare Essentials*. New York: Decker Inc.

Nunnally, J.C., and Bernstein, I.H. (1994) *Psychometric Methods 3rd. Edition*. New York: McGraw-Hill

Overall, J.E. and Klett, C.J. (1972) *Applied Multivariate Analysis*. New York: McGraw-Hill

Peterson, N.G., Mumford, M.D., Borman, W.C., Jeanneret, P.R., Fleishman E.A., Levin,K.Y., Campion, M.A., Mayfield, M.S., Morgeson, F.P., Pearlman, K., Gowing, M.K., Lancaster, A.R., Silver, M.B. and Dye,D.M. (2001) n Understanding work using the occupational information network (O*NET): Implications for practice and research. *Personnel Psychology*, 54, 451-492.

Pinneau, S. R., and Newhouse, A. (1964) Measures of invariance and comparability in factor analysis for fixed variables. *Psychometrika*, 29, 3, 271-281.

Quenouille, M.H. (1949) Approximate tests of correlation in time series. *Journal of the Royal Statistical Society Series B*, 11, 18-84.

Quinlan, J.R. (1986) Induction of decision trees. Machine Learning, 1, 1, 81-106.

Quinlan, J.R. (1993) *C.4.5 Programs for Machine Learning*. San Francisco: Morgan Kaufmann.

Ragju, T.S., Kannan, P.K., Rao, H.R., and Winston, A.B. (2001) Dynamic profiling of consumers for customized offerings over the internet: a model and analysis. *Decision Support Systems*, 32, 117-134.

Rosenblueth, A., Wiener, N., and Bigelow, J. (1943) Behavior, purpose, and telology. *Philosophical Science*, 10, 18-24.

Schmidt, F.L., and Hunter, J.E. (1998) The Validity and Utility of Selection Methods in Personnel Psychology: practical and theoretical implications of 85 years of research findings. *Psychological Bulletin*, 124, 2, 262-274.

Schmidt, F.L., and Hunter, J. E. (2004) General mental ability in the world of work: occupational attainment and Job Performance. *Journal of Personality and Social Psychology*, 88, 6, 162-173.

Shrout, P. E., and Fleiss, J. L. (1979) Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86, 2, 420-428.

Sommer, M., Olbrich, A., and Arendasy, M. (2004) Improvements in personnel selection with neural networks: a pilot study in the field of aviation psychology. *The International Journal of Aviation Psycholog*y, 14, 1, 103-115.

Swets, J.A., Dawes, R.M., and Monahan, J. (2000) Psychological Science Can Improve Diagnostic Decisions. *Psychological Science in the Public Interest*, 1, 1, 1-26.

Tabachnick, B.G. and Fidell, L.S. (2001) *Using Multivariate Statistics 4th Edition*. Boston: Allyn and Bacon.

Taylor, S.L. and Cosenza, R.M. (2002) Profiling later aged female teens: mall shopping behaviour and clothes choice. *Journal of Consumer Marketing*, 19, 5, 393-408

Tukey, J.W. (1958) Bias and confidence in not quite large samples (abstract). A*nnals of Mathematical Statistics*, 29, 614.

von Eye, A. (1990) *Introduction to Configural Frequency Analysis*. Cambridge: Cambridge University Press.

Webb, A. (2002) *Statistical Pattern Recognition 2nd Edition*. New York: Wiley.

Waller, N.G., and Meehl, P.E. (1998) *Multivariate Taxometric Procedure*s. Sage Publications. ISBN: 0-7619-0257-0.

Webster, C.D., Harris, G.T., Rice, M.E., Cormier, C., & Quinsey, V.L. (1994) *The Violence Prediction Scheme: Assessing Dangerousness in High Risk Men*. University of Toronto, Centre of Criminology.

Waller, N. (2004) LCA 1.1: an R package for exploratory latent class analysis. *Applied Psychological Measurement*, 28, 2, 141-142.

Witten, I. & Frank, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. New York: Morgan Kaufmann Publishers.

Wolfram, S (2002) *A New Kind of Science*. Champaign, Illinois: Wolfram Media Inc.