Learning Vectors for Case Study Analysis*

James Honaker†

September 5, 2011

Abstract

The "case study" is the most common method of analysis in political science, although the definition of good case study analysis remains elusive, and the objectives of this means of analysis varies widely across applications. Critical to all implementations though, is the appropriate selection of cases to achieve some objective (which may vastly range from theory generation, to exploration, to classification to causal effects or effects of causes, to theory testing).

We show that case selection is fundamentally a quantitative problem, although it is not necessarily statistical, and that it can be formalized without harm to the competing rules for good case study design. This formalization is achieved by showing the equivalency of case selection to a class of supervised machine learning problems, and most rules for case selection can be written as learning vector algorithms for this problem. Formalization allows direct comparison among alternate selection design rules, and a better understanding of the inherent differences between statistical models and case study inference.

We show that most commonly adhered rules for good case selection can be set up as a penalized learning vector in some weighting of the data space. This makes the selection of cases isomorphic to the machine learning problem of selecting prototypes. This allows a formal and rigorous insight into questions of good case study design, and implementation: the case selection process can be automated, differing selection mechanisms and criteria can be directly compared as learning vectors, and when there is a declared objective for the case study analysis, an optimal learning vector can be selected.

Moreover, the literature, and regularity conditions of the supervised learning framework can be directly applied to case study analysis, with important implications. The combining of case studies across different authors can be clearly seen as an ensemble learning problem, which brings different advice than common methods of pooling expert authors. Finally, treating case studies as a machine learning problem brings a

*Presented at the Summer Meetings of the Society for Political Methodology, Princeton, July 2011. My thanks for useful comments are due to Matthew Blackwell, Doug Lemke, David Nickerson, Phil Schrodt, Maya Sen, Teppei Yamamoto.

†Lecturer, The Pennsylvania State University, Department of Political Science; tercer@psu.edu

different light to the long debated differences between good case study design and good statistical treatments of data, as there are recognized differences between statistical inference and machine learning that are not common to the current debate framed as depth or breadth, or "large N" and "small N" studies.

1 The analysis of cases

This paper discusses the selection of cases for analysis in case study designs. The attempt is to provide the strongest foundation for case study practitioners, and a rigorous, replicable framework for selecting cases. Secondarily, the methodology proposed creates a new point of dialog between quantitative and qualitative researchers about the hallmarks of case study research.

The definition of good case study analysis remains elusive, and the objectives of this means of analysis varies widely across applications and authors. Epistemological disagreements abound, both among practitioners of the case analysis, as well as between proponents of case studies and advocates of statistical inference. Almost any definition of a case study would not include some implementation of a researcher who sees their own work in the case study tradition. Flyvbjerg concludes that there is no rigorous way to define a case study and one is "better off staying with commonsensical definitions like that from Webster's."(2011, p301-2) Even within political science, and among practitioners of case studies, "all subfields draw on the case study as a method of analysis, but with different interpretations of what it means and how to do it." (Yanow et al., 2008) Faced with this hurdle, nevertheless, this paper attempts to engage with the concept of the case study, while leaving the implementation as shielded by a black box, and bracketing the question of the case study's primary value.

The axioms, or terms of engagement by which this paper proceeds are *fidelity* and *agnosticism*. By fidelity, I mean that no methods violate, overturn, or argue against common best practices in case study research. I wish to remain faithful to the case study, as commonly understood by its advocates, instructors and practitioners. To the extent that these writers disagree, I attempt to be flexible enough to include different common approaches as possibilities. This is done by working within a minimal framework contained in the intersection of the understanding of the case study from many perspectives. Although the methods described are quantitative (in the elementary, most direct sense that they rely on numerical measurements) there is nothing in the development below that attempts to stretch case study analysis over the skeleton of statistical reasoning, or assume that case study research is "as-if"-statistical research conducted at an intuitive level.

By agnosticism, I mean that I attempt to avoid any judgment about what is the correct purpose or objective of case studies, the best implementation, and the resulting social science value. I set forward that there is a benevolent and powerful case study method for social science, but I don't know what it is. There are valuable, insightful roles for case study research, I just don't claim to know what they are. I don't want to narrowly define or

highlight a particular best practice for case study research partially for fear of being dismissed by case study researchers who disagree with that interpretation.[1] Additionally, I want to attempt to bracket the debate between quantitative and qualitative practitioners, because I don't wish to lose either from my audience. Regardless of the correct implementation and purpose of case analysis, it is broadly recognized that the selection of cases is instrumental for case study insight. If the world is complex, with many patterns to be seen, but all the cases studied are essentially clones of each other, then at best, much that is interesting will be overlooked, and at worst no leverage will exist to tease apart the possible explanations for the system studied. Formalization of the case selection process may allow insight into the advantages of different selection methods, while the replicability and transparency that comes with following formalized selection methods increases assurance that results are not driven by intentional or accidental selection problems.

2 A framework to case studies

While I can not define what a correct study design is, I will describe its abstract form. This abstract form is all of the structure that is necessary to proceed, and indeed directly guides the entire argument of the rest of this paper. This proposed definition should be at the intersection of every interpretation of case study methods. Anyone that recognizes this structure as the timber of their understanding of the good case study, should feel comfortable with all that follows.

There exists a set of all possible observations that could potentially be chosen as cases. Each of these observations has a *type*, that is the focus of interest of the analysis. In what follows, for simplicity I will assume this type is dichotomous, (although this is in no way necessary, and I will discuss generalizations beyond this in section 5.4). This type could be variously described as a "dependent variable" in causal models, or an "outcome" where forecasting is the chief concern, or a "left hand side" variable or an output of the system.

In addition to a type, each observation has some measured characteristics. These might be referred to as "independent" or "right hand side" variables, or "predictors"; in computer science, imaging and machine learning these are often referred to as "inputs" or "features". If the existing measured variables sufficiently explained the observation's type, then we would have no need to proceed with case studies.

Instead, a motivating tenet of case study research is that the available and easily measurable data is merely *superficial*. To properly understand an observation it is necessary to drill deep and explore, contemplate and winnow an enormous amount of information. This information that is required to truely understand an observation's type is difficult to collect, resource intensive, and may require specialized expertise to discover or interpret. Moreover, this is information in the broadest definition, meaning only that it be possible to be communicated, even if this information might be effectively non-quantifiable. Eventually, correct

[1] Hawking quips that each additional equation halves your remaining audience, and perhaps the same is true with every concrete declarative statement you make defining what a case study is.

understanding of the cases identifies and isolates the *thick* information that brings about the observation's type.

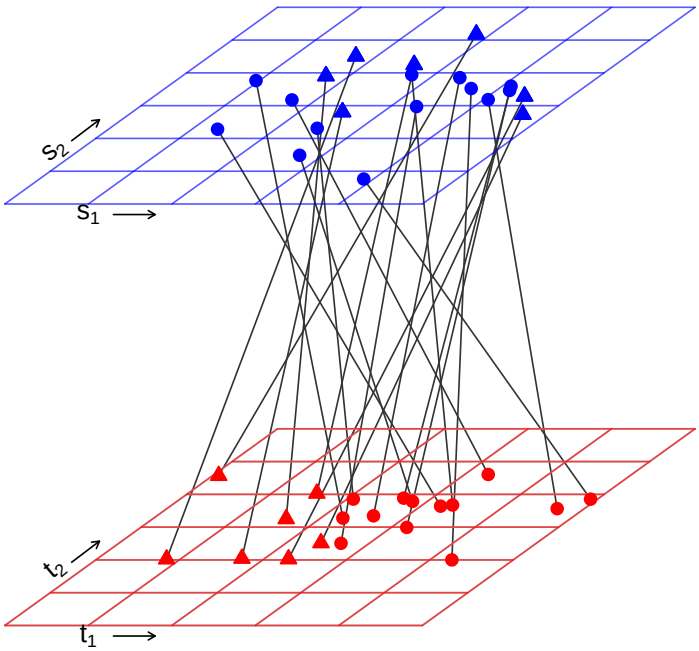
Thus the set of possible cases (observations) have type, y, some vector of superficial quantitative variables $\mathbf{s}$ and discovered thick characteristics $\mathbf{t}$. A representation of this is shown in figure 1a. Here there are two superficial variables, and all observations have a location is the two dimensional space of $\mathbf{s}$. There are similarly two thick characteristics that correctly define the observation's type. Observations have locations in both sets of dimensions and those positions are joined by a line segment. Observations whose location are close in $\mathbf{s}$ might be more distant in $\mathbf{t}$, so $\mathbf{t}$ is a probabilistic projection of $\mathbf{s}$. Between these two planes, their gazes restrained upwards, are the inhabitants of Plato's cave. Every observation also has a type, where here, types are either triangles or circles, thus here figure 1 is a representation of five dimensions.

A correct study of an individual case isolates a thick set of characteristics that lead to that observation's type. If the case study is correct, observations that have these same thick characteristics will be of the same class, because by definition the thick variables are the correct story. Here results our fundamental definition. Case studies form a correspondence between certain studied points in the dimensions of thick characteristics to the set of possible types an observation may have. The term *correspondence* is employed for twofold duty. Firstly, it has a vernacular understanding that does not imply too much adopted meaning on the nature of this relationship, and continues to be agnostic to the purpose and intent of the case analysis. Secondly, a correspondence has a technical mathematical meaning that some readers will recognize as a generalization of the function; a mapping between sets that is possibly not one-to-one.

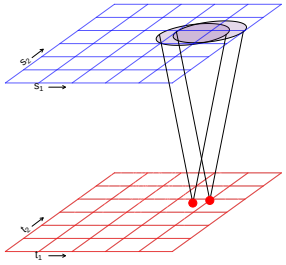
Proposition 1. *All individual case studies claim "Objects that have **these** key characteristics are mapped to **this** group."* $(c_k : \mathbf{z_k} \to \hat{y_k} \subset Y)$

Together, the set of case studies, $C = \{c_1, \cdots, c_K\}$, form an incomplete correspondence. They map only from the cases studied, thus they map only from a few specific points in $\mathbf{t}$.[2] To be usable, that is, for the case study method to provide guidance, we need to interpret the specific understandings from points in $\mathbf{t}$ to give guidelines for other locations for which an exact case study does not exist. We should expect an observation that is exactly like a studied case, except with some minuscule difference in one of the thick characteristics, to behave as that close case, and belong to the same type, as it is almost identical in the important identified characteristics. If this not true then the case study does not teach us anything. As Gerring writes "In order for a focused case study to provide insight into a broader phenomenon, it must be representative of a broader set of cases" (2007, p91). As these differences between these transductive locations and the studied case increase, we may be less certain that the guidance of the case applies to this displaced set of characteristics.

[2]That we map only from $\mathbf{t}$ gives us a simple corollary as a strict definition of what we mean by superficial, which will be implicitly useful in much of the later argument. *"In the presence of all information, additionally knowing the values of any superficial variables is not influential in any case study."* $c^* : (s_k, t_k) \to y_k; c : t_k \to y_k$, then $\mathbf{s}$ is superficial if $c = c^* \quad \forall k$



(a) *Every observation has values both in the dimensions of s and t, here paired respectively by line segments that link from t to s. The observed variables, s_1 and s_2 bear a relationship between each other, and the class of the observation (here signified by circles and triangles). However, the unmeasured variables t_1 and t_2 are those that directly lead to the class, and the position of* $\mathbf{t}$ *is related to y only through omitted variable bias.*



(b) *Here we see the probabilistic realisation of the superficial variables,* $\mathbf{s}$*, from the thick variables,* $\mathbf{t}$*. Observations close in* $\mathbf{t}$ *might be further apart in the more readily observed* $\mathbf{s}$

Figure 1: Represention of the thick and the superficial dimensions.

However, if the only information we allow ourselves for guidance are the current case studies, then for understanding what will occur at some location $\mathbf{t_j}$, we should look to the most similar case study. This leads to our second proposition about the nature of case study analysis.

Proposition 2. *Guidance for any set of thick characteristics is found by finding the closest studied case study in the space of the thick information, and following that correspondence.* $(G : t_i, C \rightarrow y_k \subset Y : t_k = \arg\min_k \|\mathbf{t_i}, \mathbf{t_k}\|)$

Guidance in case studies requires locating the closest explored case to the instance or observation which needs guidance.[3] Being able to judge distance, or similarity in $\mathbf{t}$, between an observation and the studied cases does not require that $\mathbf{t}$ be quantitative, and the distance be a quantitative metric, only that it is possible to find the minimization solution, that is, identify the closest or most similar analyzed case.

Two questions are left undefined by these proposed definitions. First, how $\mathbf{t}$ is generated or discovered once the cases have been selected. Partly this is because this is not necessary for these purposes, a continuing subject of research and debate amongst practitioners, and the immediate benefits of formalizing this are not evident for present purposes. To the extent that the process of uncovering $\mathbf{t}$ influences or requires a particular set of cases, we will consider adaptations in section 4.1. Secondly, no claim needs to be made as to what the guidance, $G : t, C \rightarrow y$ means, or can be used for.

2.1 Overview

In summary, the following is our attempt to set out a definition of the case study process, with both fidelity and agnosticism.

1. Observations have superficial values.
2. Researchers carefully select a limited set of cases for analysis.
3. By some method they are able to uncover the correct thick variables which are truly important in these cases.
4. Unexplored observations that are close in the thick characteristics to a studied case, behave like that case in important respects.

Proposition 1, its corollary, and proposition 2 define steps 3, 1 and 4, respectively. Our focus from here is the implications for step 2.

[3] A simple corollary of this proposition is that only observations for which $\mathbf{t}$ has been identified can serve as prototypes. That is, If $G : t_i, C \rightarrow y_k, \exists c_k \in C$, or more succinctly, *the unexamined case is not worth following*

3 Learning Vector Quantization

We shift gears now to describing some topics in machine learning, for reasons and connections we will make clear. Machine learning is a form of quantitative methods that attempts to find patterns in data, when the data is commonly large in numbers of variables, but short on theory or knowledge about these variables by which to leverage statistical models. One set of problems often addressed are methods by which to classify data, that is, partition observations based on their observed features, into sets of observations that share the same type. One very powerful framework for this task is *classification by prototypes*. The goal of these methods are to find exemplars of local patterns in the data. A set of exemplars are found that broadly cover the space of the data. (Potential) Observations are (transductively) classified by finding the nearest prototype to that location in the feature space, and assigning that location the same group label as the proximal prototype. That is, any location is nearest-neighbor matched to one of the prototypes, and classified as that type (For stricter definitions of prototype methods, look ahead to section 3.2).

Hastie et al., give such methods the fulsome praise of "*black box* prediction engines." (2009). In full:

> "Because they are highly unstructured, they typically are not useful for understanding the nature of the relationship between the features and class outcome. However, as black box prediction engines, they can be very effective, and are often among the best performers in real data problems." (p.459)

There are many methods in existence to construct and place prototypes in the space of the data, or which can be interpreted as equivalent to some prototype styled model, including k-means clustering, and k-nearest neighbor matching. Several statistical models well known to the social sciences can also be employed, particularly mixtures-of-normals models, and other latent class based techniques.

In what follows we explore the prototype model of *learning vector quantization* (LVQ), and choose this over the long possible list for three reasons. It is algorithmically cheap to implement, and is known to perform similar to most other reasonable methods in well structured problems. Its semi-parametric form is highly flexible. Finally, the algorithmic steps are simple to understand and concrete to modify. We will argue, additional points in its favor with regard to case study research further on. The steps of the algorithm are as follows:

1. Choose some number, K, of prototypes to use for each type.

2. Randomly sample, perhaps from the original data, a set of prototypes, or *codebook vectors* $\{c_1^1, \ldots c_K^1\} = C$, where the subscript identifies each individual prototype and the superscript identifies the iteration of the algorith (here 1) .

3. Randomly choose one observation in the dataset, s_i, and calculate the closest prototype c_j^t.

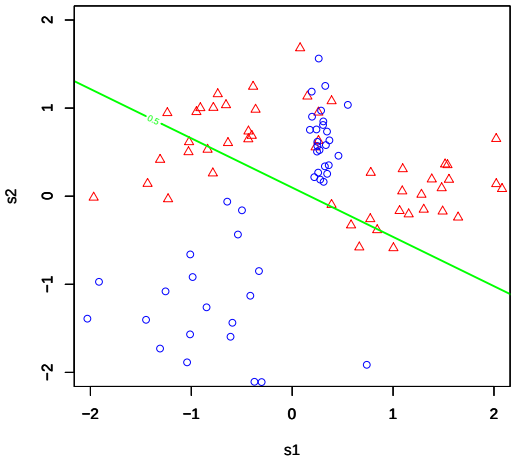
   (a) If s_i and c_j are of the same type, move the prototype closer to the observation: $c_j^{t+1} = c_j + \epsilon_t(s_i - c_j)$.

   (b) If s_i and c_j are of the opposite type, move the prototype away the observation: $c_j^{t+1} = c_j - \epsilon_t(s_i - c_j)$.

4. Repeat step 3, many times, using a decreasing sequence of ϵ_t, named the learning rate.

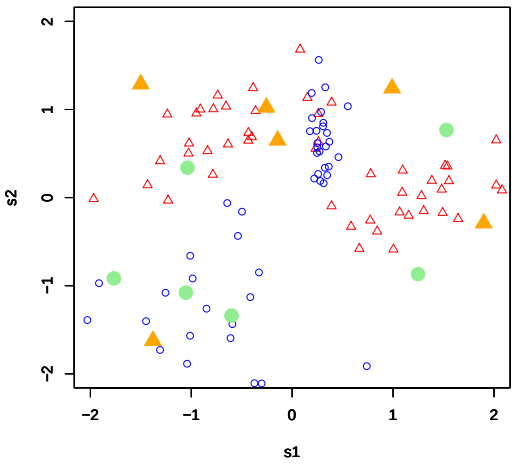
3.1 Example

To demonstrate how this method works, and to build an intuition surrounding key ideas in prototype classification and LVQ, we demonstrate some steps in the implementation. In the figure below we see one hundred observations of two types ($\triangle$, $\bigcirc$) distributed across two observable superficial dimensions, s_1 and s_2.



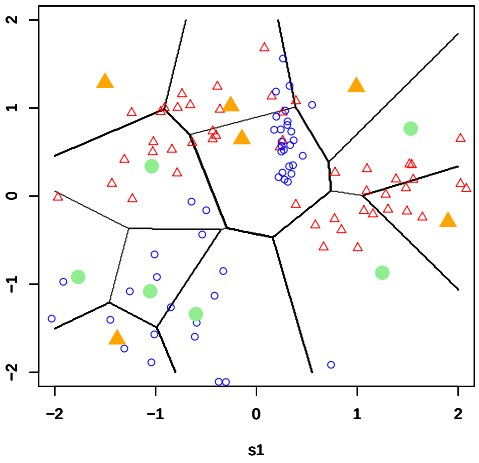
A simple parametric model to classify these observations might be to code types as 1's and 0's and use Logistic regression to predict the $\hat{y}$ with s_1 and s_2 as covariates. All points in $\mathbf{s}$ have a value of $\hat{y}$, and the green line marks the division between $\hat{y}$ above and below 0.5. Thus, this is the classification boundary that Logit would obtain; locations above (or below) this line are classified as $\triangle$ (or $\bigcirc$).

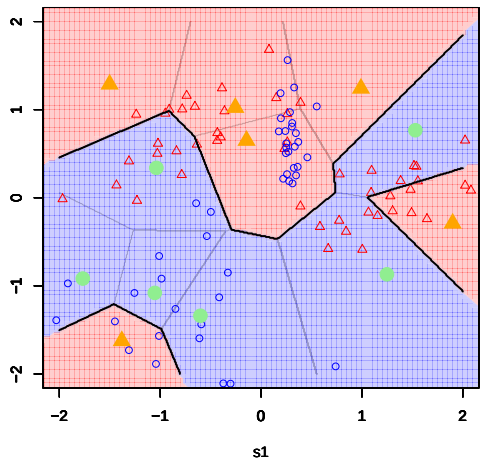
This particular statistical model does a poor job of classification for this data, misclassifying a large and central cluster of observations, and scattered observations throughout (only 63 of 100 observations would be assigned the correct type if using the classification boundary). Other, more flexible statistical models, such a two-dimensional spline, might have better performance.



Prototypes of each type are initially selected, possibly at random in the dimensions of the space, or randomly from the observations in the dataset. Here, six prototypes are used for each class of observation. For observations of type △ the prototypes are marked ▲ and similarly, ● are prototypes for observations of type ○ .

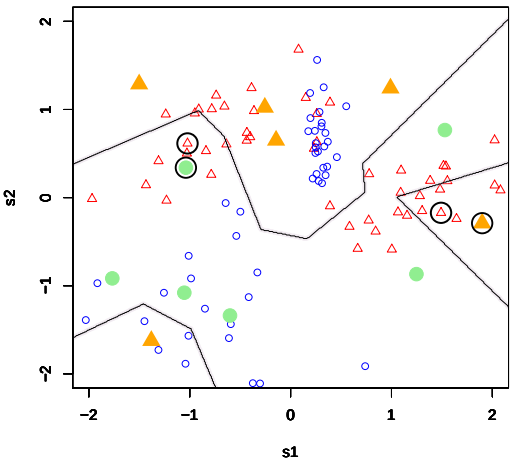
Any point in the space, $\mathbf{s}$, is classified by finding the closest prototype to that point, and assigning it to the group of that prototype. That is, all points are nearest-neighbor matched to the closest prototype. Every prototype has a surrounding neighborhood, named a *Voronoi cell*, in which it is the closest prototype, and thus everything in that region would be classified according to that prototype. The boundaries of the Voronoi cell are composed of segments of the hyperplanes that are equidistant between prototypes. The set of all cells is called the Voronoi tesselation.

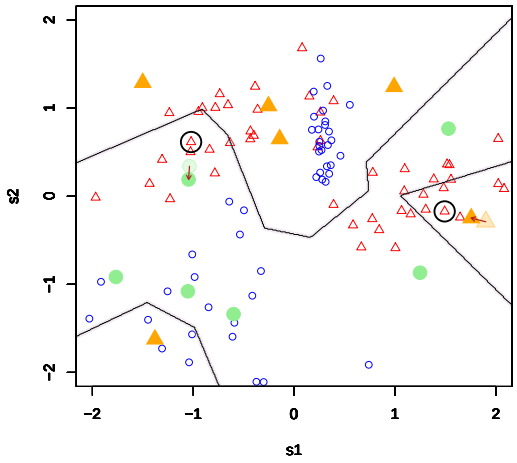




Joining all the Voronoi cells that map to the same type forms a classification boundary between types. Using the current locations of prototypes, the region currently classified as type ○ is shaded blue, and the region for type △ is shaded red. In the left, there is one region classified for one type, and three discontinuous regions for the other. With this initial distribution of prototypes there are many observations that would currently be misclassified.

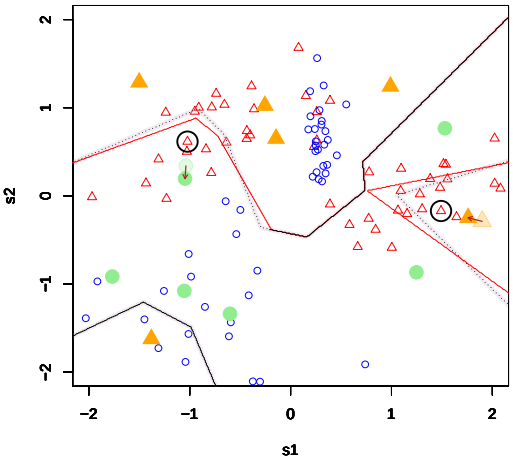
In each iteration of the algorithm, one observation is selected at random. Here we are demonstrating two possible iterations with the two observations selected circled to highlight. For both of these, the nearest prototype has been determined and is also denoted. In the left example, the nearest prototype would have incorrectly classified the selected observation as the opposite type. In the right example, the nearest prototype would have correctly classified the observation in question.





The closest prototype is moved towards or away from the selected observation depending on whether correctly or incorrectly classifies that observation. The movement, of some distance ϵ, shown by the vector $\rightarrow$, starts at some reasonable value, and decreases in length as the iterations increase.

Movement in the prototypes correspondingly results in movement in the classification boundaries; here the newly changed boundaries are marked in red. Some boundaries have not changed (noted in black) as these boundaries are determined by prototypes that themselves have not moved.



After ten thousand iterations the prototypes (large symbols) have settled over the data (small, open symbols). Positions in the parameter space are classified by nearest neighbor matching to the closest prototype. Prototypes are drawn towards local regions with many observations of matching type, and repelled from regions where the prototypes is misclassifying. Each iteration resembles one component of a Monte Carlo integration of the attractive force of all the data on the prototypes.

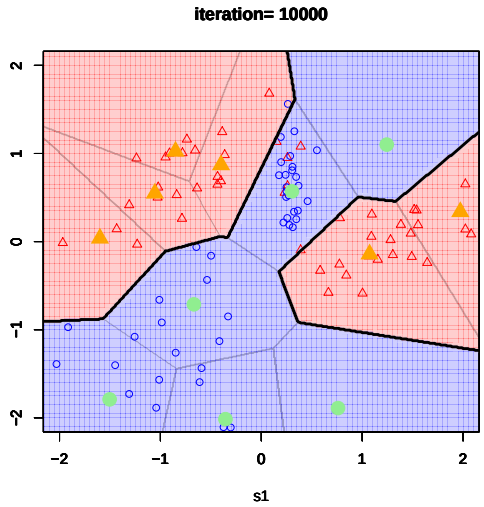


Figure 2 shows the path these prototypes take over the ten thousand iterations. The darkest image shows the final location of each prototype, the red line the path it took over the iterations, and the ghosted images trailing behind are the locations after each thousand iterations. Prototypes that were initialized poorly (intentionally for pedagogical purposes) can end up moving substantially. The resulting decision boundaries classify the data well (95 of 100 observations are correctly classified), where the right edge of the top left cluster of △'s overlap the central grouping of ○'s . One of the prototypes (bottom right) has only one observation in its Voronoi cell.

3.2 Summary

In summary, prototype methods provide a powerful and flexible means of classifying quantitative data. Prototypes form exemplars that cover the space of the observed data, and identify groups or clusters of observations of the same type that have similar features. Each prototype, m_k, is an representation of an idealized member of that feature-cluster-type, relating observed features $\mathbf{s}$ to type, y:

Definition 1. $m_k : \mathbf{s_k} \rightarrow \hat{y_k} \subset Y$

Nearest neighbor matching to the prototypes results in winner-takes-all, hard-edged coding decisions. Given a set of prototypes, any point in the space can be classified according to the type of the closest prototype.

Definition 2. $L : s_i, M \rightarrow y_k \subset Y : s_k = \arg\min_k \|\mathbf{s_i}, \mathbf{s_k}\|$

Prototypes themselves are black boxes. There is no expository *why* with regards to how it is that a local cluster of observations share the same type y, the prototype simply. While the prototypes provide no exploratory insight, they are easy to interpret, as they are stylized observations with a set of values in the feature space. They tell an easily relatable and interpretable story about typical observations in the observed data.

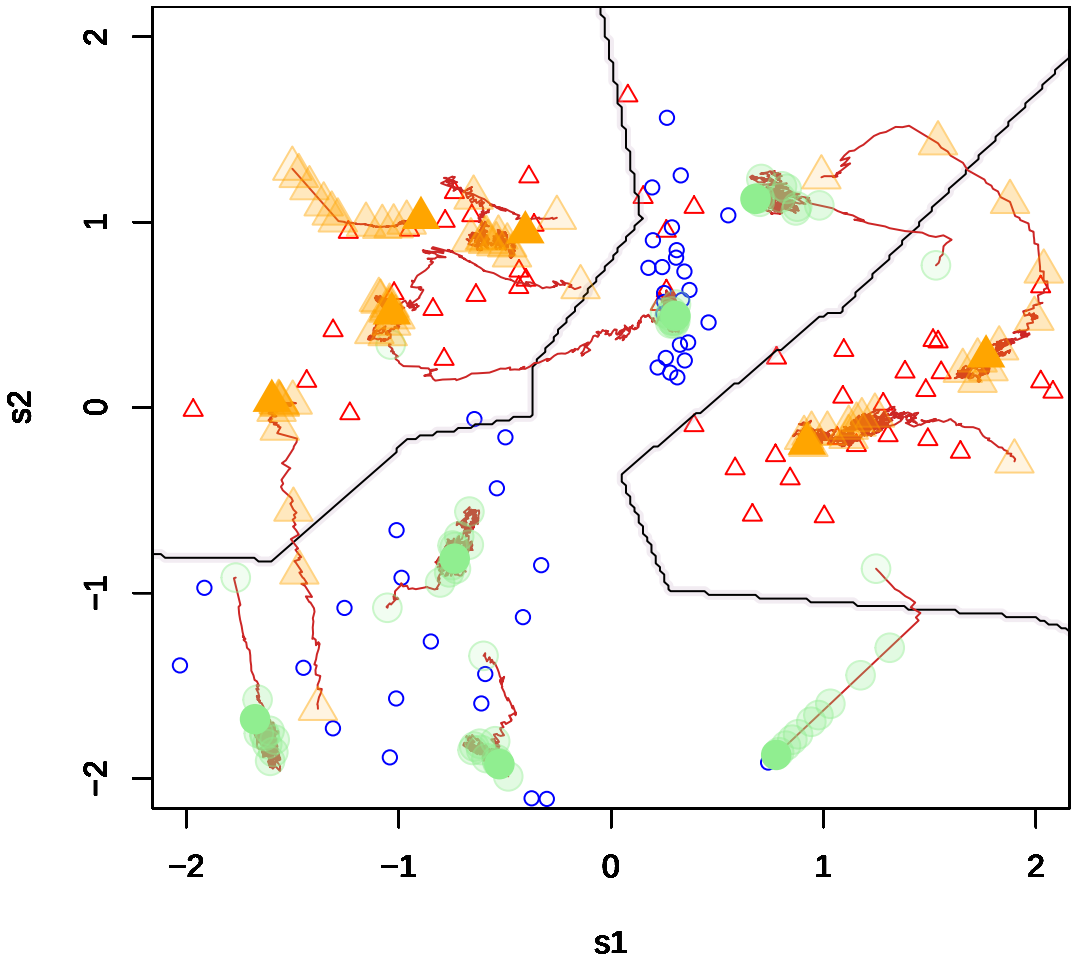


Figure 2: *The movement of the prototypes through the observed space. The darkest symbols represent the final positions of the codebook vectors, the red lines the path they took, and the ghosted images the locations after each thousand iterations.*

3.3 Classification versus Inference

Prototype methods provide a powerful means of classification of quantitative data, without adopting assumptions required by statistical modeling. This is not to state that prototype methods are superior to statistical methods, but that a smaller set of axioms provide a more limited, less ambitious–if sometimes more easily flexible–set of tools. Fundamentally, the goals and hallmarks are very different.

Statistical methods, somewhat inadvertently, classify observations; statistical models estimate underlying densities and their uncertainty, and these densities can generate posterior probabilities that can be used to create decision boundaries.

The most appropriate statistical model might not be the best classifier among the set of statistical models, as classification is not the objective function that the estimator is conditioned to maximize; oftentimes statistical modelers will remove variables that are excellent at classification because they are inappropriate for that model. Statistical models have additional objectives that pure classification ignores, particularly calculation of uncertainty.

However, classification models do not require the statistical assumptions that are conventionally seen as central to the disagreements between statistically oriented and case study scholars, or conventionally, "large N" and "small N." Prototype methods require the presence of cross-case quantitative data, which everyone acknowledges exists, without making claims about its centrality to inference.

4 Learning Vectors for Case Selection

Our argument is rooted in the nearly identical structure of the proposed abstract form of the case study (proposition 1, propositon 2) and the definitions of the methods of prototypes (definitions 1 & 2). These are reiterated in table 1.

Proposition 1 gives our minimalist definition for what an individual case study achieves, a mapping from a set of characteristics that define the important aspects of a case, $\mathbf{t_i}$ to the corresponding associated type y_i. In prototype based methods, by definition 1, every prototype performs the role of an exemplar, an idealized template representative of a cluster or grouping of similar observations. The difference here, is that any learning algorithm (or statistical estimator) to uncover prototypes is going to iterate over the observed, quantified information available for every observation. For example, LVQ requires the positions of all the observations in the feature space $\mathbf{s}$. The case study is instead going to map from the thick information,$\mathbf{t}$.

Proposition 2 gives our minimalist definition for how to learn from a case study. Observations similar in the important variables to any case study c_j are given the same type as the case study analyzed. Similarity is determined by distance in the thick information. Whether

Proposition 1 $\mathrm{m}_k : \mathbf{s_k} \rightarrow \hat{y_k} \subset Y$	Definition 1 $\mathrm{c}_k : \mathbf{t_k} \rightarrow \hat{y_k} \subset Y$
Proposition 2 $L : s_i, M \rightarrow y_k \subset Y : s_k = \arg\min_k \lVert \mathbf{s_i}, \mathbf{s_k} \rVert$	Definition 2 $G : t_i, C \rightarrow y_k \subset Y : t_k = \arg\min_k \lVert \mathbf{t_i}, \mathbf{t_k} \rVert$

Table 1: Propositions and definitions for case analysis and classification by prototypes.

this distance is quantifiably measured, judged by experts, or hermeneutically interpreted, is not fundamental. What is key, is that case studies provide guidance to a small set of fixed known points whose correspondence between $\mathbf{t}$ and y is known, just as prototype methods provide transductive inferences by nearest neighbor matching in the space of the prototypes.

Therefore, if we could, somehow, run the LVQ algorithm in the dimensions of the thick information, we would have case studies. Of course, we do not know what the thick variables are without the methods of case analysis. And then, even when the dimensions are uncovered, we might not have the values of $\mathbf{t_i}$ for all observations, but only for those that have been intensely studied.

Given a set a cases for analysis, in addition to whatever else they perform, propositions 1 and 2 state that the framework of case studies is isomorphic to the classification of a feature space by the method of prototypes. Case studies might do very much in addition to this, but minimally, and across all implementations, case studies classify the feature space of the uncovered thick dimensions. Case studies that do not classify by prototypes do not meet propositions 1 and 2. Case studies that do not classify by prototypes do not provide any guidance.

However, this is not to say that case studies could be replaced by a prototype generating machine learning algorithm. Case studies reveal the dimensions of $\mathbf{t}$. Machine learning algorithms, dependent on cross-case quantitative information, are constrained to operate on the space of $\mathbf{s}$. However, if the cases are poorly selected, the case study will have poorly distributed prototypes in $\mathbf{t}$ and do a poor job of classification and guidance. Prototypes, that cover $\mathbf{s}$ and classify well are the best distribution of prototypes we can initially select with no additional initial information.

If it is possible to classify y with $\mathbf{s}$, there must be some correspondence between $\mathbf{s}$ and $\mathbf{t}$. This could be that $\mathbf{s}$ are poorly measured versions of $\mathbf{t}$ and suffers a lack of construct validity, or that the relations between $\mathbf{s}$ and $\mathbf{y}$ is all omited variable bias, or that $\mathbf{s}$ is post-treatment, while $\mathbf{t}$ is treatment. These are all arguments proposed by case study researchers for the reasons while relationships found from statistical models between $\mathbf{s}$ and y are naive. Any of these could be true, and they would allow $\mathbf{s}$ to provide an estimate of the structure of $\mathbf{t}$ even if $\mathbf{t}$ is yet to be revealed.

Prototype methods are considered a "black box" method of classification because they give little exploratory insight into the behavior of individual variables, or indeed any parametric estimates. Each prototype behaves in an unexplorable fashion in a particular point in space, and classifies a surrounding territory. However, this is appropriate for building a foundation for case studies, as it the case analysis that will provide the explanation of the behavior in the vicinity of the prototype. The manner in which they are a black box directly leaves the *just so* story to explain the prototype's behavior to interpretive methods.

Structure in $\mathbf{t}$ will result in noiser structure in $\mathbf{s}$ if $\mathbf{t}$ directly brings about y and $\mathbf{s}$ weakly predicts y. It is the job of the case study to discover the informational dimensions of $\mathbf{t}$, while the actual case studies have nothing to say about which cases should have been selected. Prototype methods purport to discover all the available structure from the superficial quantitative data $\mathbf{s}$, by finding the best distribution of prototypes, all the while remaining silent

about why any prototype behaves as it does.

In summary, a good selection of cases should do the best job possible classifying the dependent variables from the superficial data, because case studies are themselves a classification by prototypes of a related, but more crucial, set of dimensions.

4.1 Implementation and extension

Our argument is that observations selected by prototype methods to classify **s** provide the best set of observations for analysis of case studies and exploration of **t**. Strictly, the prototypes themselves are not observations, but locations in the space of quantitative data, which have an assigned type. If the set of subjects is malleable such that an observation of any type can be tracked down for any location, such as with subjects from a large pool relative to a small number of variables, then a case study can be found that exactly matches each of the prototypes generated. More commonly, an observation very close to each prototype will need to suffice. This is no different from simple, unidimensional quantitative selection rules that propose using an observation at the mean of some key variable, when no individual observation is guaranteed to be at precisely that point.

To what extent does this differ from current best practices for selecting samples for case analysis? Seawright and Gerring set out seven different methods commonly used for selecting a sample in case study research (Seawright and Gerring, 2008, also Gerring 2007, ch.5), of which the most commonly implemented are the *diverse*, *typical*, *extreme*, *most similar* and *most different*.[4]

Seawright and Gerring's discussion focuses on one dimension of variables to select from. *Diverse* cases "are likely to be representative of the full variation in the population" while *Typical* examples are "typical examples of some cross-case relationship." and either located together in the center of distributions or small in the magnitude of the residual of some fitted model. On the other hand, *Extreme* examples are as unalike as possible and represent the extreme possible values of the observed variables.

The key characteristic of prototypes selected by the conventional implementation of LVQ are that they blanket the space of the features. In that sense, they are diverse by construction.

Typical models, if selected by residuals, could unwittingly also be extreme, if selected by being on the edge of the data space: This is particularly likely if the model generating the residuals is non-linear. There is no straightforward "model" from which to generate residuals in the LVQ approach, but parametrically, observations that are typical, and well fitted, are observations that are in the center of (possibly multidimensional) patterns of data. If the data space is made of many clusters of data, a sufficiently flexible parametric model would find observations at the centers of these clusters to be well fitted and have small residual. We take typicality then, to mean, in the center of clusters of similiar observations. Similarly, extreme observations, are observations at the edges and boundaries of any clear patterns.

[4]Sekhon (2004) traces the origin and popularity of these techniques to Mills' work *A System of Logic* and his *method of agreement* and *direct* and *indirect* versions of the *method of difference*.

Several different changes to the LVQ algorithm have been proposed to achieve additional desirable characteristics in the prototypes. Partially this has been facilitated by the transparency and ease of implementation of the algorithm. For example, an adaptation called LVQ2 pays more attention to forming correct boundaries by selecting only misclassified observations and adjusting both the nearest (incorrectly classifying) prototype, and the nearest prototype of the correct class. LVQ3 achieves increased stability in the prototype locations by using weights on the prototype movements based on their distance to the decision boundary (Kohonen 1990).

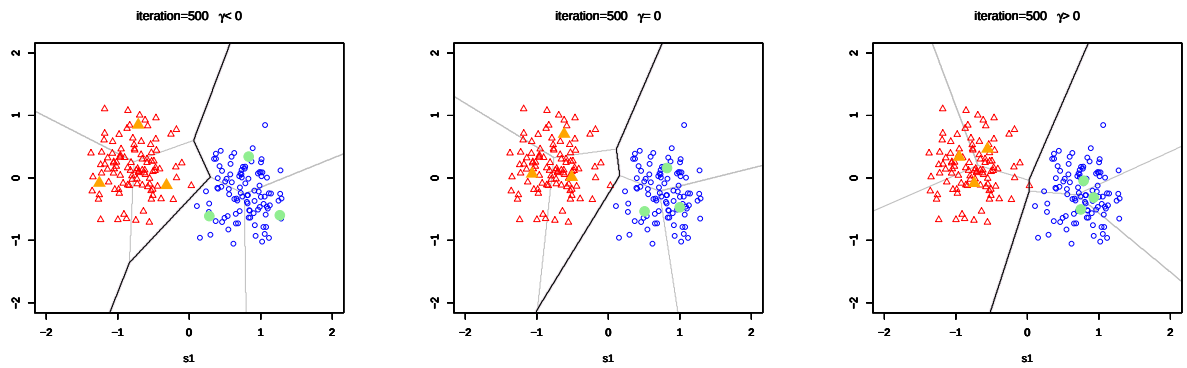
As an advance on LVQ, in fidelity to the ideas surrounding typical and extreme case selection, we propose and demonstrate an algorithmic addition that can incorporate additional desirable properties of prototype cases as understood by these *desiderata*. We add the concept of prototype *charge* as an additional step in the LVQ algorithm. When any prototype, c_k, is selected for movement (because the randomly selected observation, s_i was contained in that prototypes Voronoi cell), we assume that prototype experiences an attraction to all other prototypes of the same type, that decreases as the square of distance. The total movement of the prototype now becomes:

$$c_k^{t+1} = c_k^t + \text{sign}\, \epsilon_t(s_i - c_k) + \sum_{i=1}^{K} \delta_i \frac{c_i - c_k}{\|c_i, c_k\|^2}$$

The central term is the movement in the prototype, as before, based on the direction of the randomly selected observation, the learning rate ϵ and where the operator sign $\in \{+1, \text{-}1\}$ depends on whether s_i is correctly classified. The rightmost term is the addition to the algorithm. The key value δ_i expresses the attraction that any prototype experiences towards another prototype. We might assume, for example, that $\delta_i \in \{0, \gamma\}$ is a small positive number γ if i and k are of the same type, and zero otherwise. The numerator expresses the vector direction of any other prototype to which our prototype might be attracted, while the numerator diminishes this attraction at a quadratic rate, as analogous, for example, to charged ions experiencing attractions to each other in a electrodynamic system.

When γ is positive, prototypes of the same class, are attracted to each other. When γ is negative, prototypes of the same class are repelled from each other. When $\gamma = 0$ the right term drops out and we have returned exactly to the original LVQ algorithm. Properly chosen, γ gives us a continously defined parameter that allows us to move from simple LVQ, which achieves a *diverse* selection of prototypes, to either group of prototypes that are more *typical* or more *extreme*. Examples of these are shown in figure 3.

The groupings and clusters of data still attract the prototypes, and the center term still acts as a Monte Carlo integration of the force exerted on the prototypes by the observations. We likely want the data to dominate the choice of prototypes, even with this refinement, thus care should be taken in the choice of γ. Too large a positive value will be stronger than the attraction of the data, and collapse all the prototypes of the same type onto the same location. Too large a negative value will repel the prototypes to such an extent they may reside outside the bounds of the observed data.



(a) *Prototypes of like type repel each other, resulting in samples more extreme in the distribution.*

(b) *The standard LVQ prototypes, which are nested in the adaptation at $\gamma = 0$ and thus and thus diverse prototypes towards the center of mass of the constituent observations of each Voronoi cell.*

(c) *Prototypes of like type attract each other, resulting in sample more typical and similar in the distribution.*

Figure 3: Adaptations of the LVQ algorithm for implementing a continuous tuning parameter for *typical* to *extreme* sample selection.

5 The Lens of the Machine Learning Framework

The framework of classification by prototypes as a method of case selection, together with the characterization of case analysis as minimally a classification problem has implications for many long standing questions about appropriate case study practices, including sampling conditional on variables, the pooling of results across studies, and the use of case studies to explain "deviant" observations.

5.1 Conditional Sampling

Whether or how to sample conditional on other variables is a continuing point of debate among case study practitioners. "Selecting on the dependent variable" triggers instinctual fear of dangers that are well understood (for example, Geddes 1990). King and Powell (2008) tie some of these conditional selection issues to recent topics in statistical methodology. In the event that some type memberships are very rare, they recommend intentionally oversampling by the dependent variable, following the logic of rare-events models which sample on the oversample rare outcomes for efficiency purposes, and then analytically correct the bias this creates in estimators (King and Zeng, 2001). They also point to the argument in Imai et al. (2008) to consider stratification techniques, and following that work, perhaps blocking techniques for case selection. However, they strongly warn against doing both, with the more nuanced advice "If you select cases based on the values of both your dependent and explanatory variables, nothing is left to learn from the data, so you must be sure not to select on both." (p.24) Prototype methods, as an instance of supervised learning, clearly

violate this application of statistical reasoning, although in a complex fashion; Selection on y is implicit by choosing the number of prototypes of each value of y, while selection on the "explanatory variables" is exactly what the prototype positioning is achieving. Prototypes are not a form of blocking, but viewed as a form of intentional selection on the covariates they are close to stratification or maybe even closer to *raking* as a form of establishing distributions on the feature space.

5.2 Ensembles

Commonplace arguments state that as the number of analyzed cases accumulate through continued study, across perhaps different authors, these cases can be treated analogously to additional observations in a statistical model (In the extreme, the findings of each case cumulate and the "small N" framework starts to turn into a "large N" statistical problem). To the extent that case analysis is a form of prototype classification, this intuition of additivity at the level of the observation is clearly incorrect. If a set of prototypes have been constructed to accurately construct decision boundaries and classify data, the prototypes have been arranged in careful balance in the quantitative space. If for example, a series of prototypes have been selected and currently do a good job of classifying some features, and suddenly and unilaterally an additional prototype is dropped into the space, all the surrounding boundaries change and classification performance may drop precipitously. For example, in the last figure from the demonstration of LVQ convergence, if an additional prototype of either type were suddenly deposited near the classification border, that prototype–with no other related prototype to counter it– would suddenly include much of its Voronoi cell beyond the current classification boundary, and miss classify previously well classified observations, as in figure 4. Prototypes can not be imported that are not in equilibrium with the other prototypes (the number of prototypes in the model could certainly be increased, but then entirely new equilibria would result for all prototypes). Instead, the correct method to combine classification results across studies is the method of *ensemble learning*.

Ensembles allow multiple models to make multiple, self-contained, predictions, and then these results are combined. In the simplest constructions, especially where it is hard to judge relative model quality, the ensemble might be simply be the *committee method* where each model that can make a classification of some particular point "casts one vote" and plurality rules when models disagree. If the relative merits of models can be judged –by quantitative measures of performance or subjective judgements of quality, by preferences for parsimony, by or other objective functions or information criteria– then the vote of any model can be weighted by its measured value or performance. In the extreme, where the model performance can be calculated or approximated as a Bayes factor, then ensemble methods reduce to Bayesian model averaging.

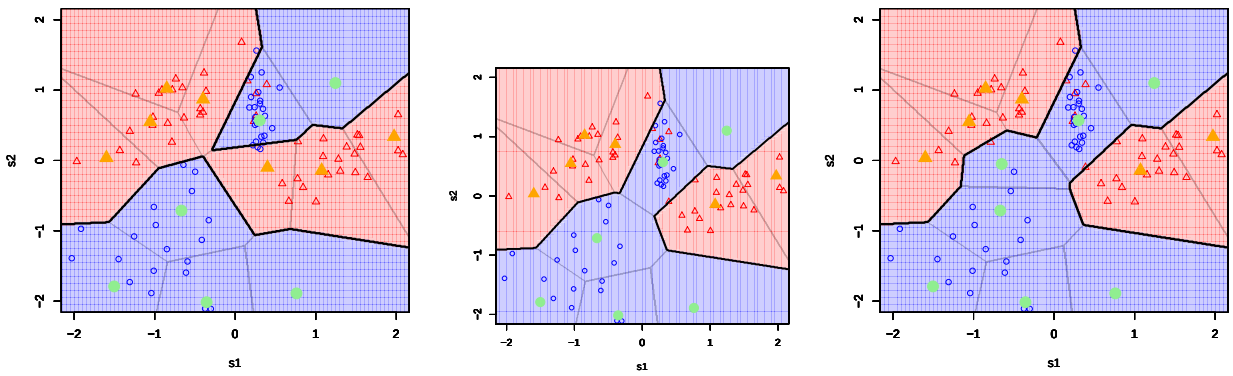


Figure 4: The center figure shows the correct classification boundaries as previously shown in equilbrium, while the left and right figures show how the classification boundaries are preturbed by adding one additional case out of equilibrium with the LVQ algorithm, near the boundary. Additional cases, out of equilibrium, *decrease* the correctly classified rate.

5.3 Boosting

Perhaps the most powerful extension of ensemble methods is the technique of *boosting*. This technique runs a iterative, progressively dependent, series of classifiers on the same features. Any observations that are misclassified in the first model are given extra weight in the next model, while observations that are correctly classified are down weighted. Sequentially, observations that are very easy to classify are progressively downweighted, such that the attention of the next model in the sequence is to classify the "difficult" observations that have been repeatedly misclassified. Early models in the sequence correctly classify the "simple" observations and latter model the "difficult" ones, and the entire sequence of models is used as committee.

This is a seemingly unconventional method to classify data, intentionally building a sequence of models that disagree with each other, and that weight the data in different fashions. However, this approach seems closely related to the case selection method of choosing *deviant* cases (Seawright and Gerring 2008). In this method of selection, some initial model is run (the assumption is generally that this is a simple, exploratory parameteric model) and then deviant observations that differ drastically from the model predictions (perhaps measured by large absolute residual) are selected as cases for analysis that may have novel insights. Boosting seems strongly tied to this idea for case selection. There are no residuals in classification by prototype, but misclassified observations in the first classification model, would be the observations given increased weight, thus more attraction by the prototypes, in the next model. Deviant case selection, might be interpreted as a short sequence, intuitive, even folk-implementation of boosting.[5]

[5]Where *folk* is used as a jargon of praise, for a mathematical method that was widely understood to work, but could not be proved correct until long after it was accepted as likely true.

6 Conclusion

The selection of cases is critical to case study analysis, but the rules currently implemented are predominantly intuitive, and often unidimensional. We show that case selection is fundamentally a quantitative problem, although it is not necessarily statistical. A formalization is achieved by showing the equivalency of case selection to the machine learning task of classification by prototypes, and this formalization can be implemented without harm to the competing rules for good case study design, or current intuitive understandings of sample selection, while generalizing easily to many quantitative dimensions in a rigorous, replicable, transparent fashion. Conceptualizing case analysis as *minimally* a problem of classification by prototypes–irregardless of whatever else case analysis may legitimately pursue and achieve– brings a different vantage and point of dialog to questions of longstanding debate between qualitative and quantitative approaches.

References

Flyvbjerg, Bent. 2011. "Case Study," in Norman K. Denzin and Yvonna S. Lincoln, eds., *The Sage Handbook of Qualitative Research*, 4th Edition. 2011. Sage: Thousand Oaks, CA. Chapter 17, pp. 301-316.

Geddes, Barbara. 1990. *How the Cases You Choose Affect the Answers You Get: Selection Bias in Comparative Politics.* Political Analysis (1990) 2 (1): 131-150. doi: 10.1093/pan/2.1.131

George, Alexander L., and Andrew Bennett. 2005. *Case studies and theory development in the social sciences.* MIT Press: Cambridge, MA.

Gerring, John. 2007. *Case Study Research: Principles and Practices.* Cambridge University Press: Cambridge.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition).* Springer: New York.

Imai, Kosuke, Gary King and Elizabeth Stuart. 2008. "Misunderstandings Among Experimentalists and Observationalists about Causal Inference." *Journal of the Royal Statistical Society, Series A* 171, part 2:481502.

King, Gary, and Eleanor Neff Powell. 2008. "How Not to Lie Without Statistics." Working Paper, Copy at http://j.mp/lYToFR

King, Gary, and Langche Zeng. 2001. "Logistic Regression in Rare Events Data" *Political Analysis.* 9 (2): 137-163.

Kononen, Teuvo. 1990. "Improved Versions of Learning Vector Quantization." *Neural Networks* pp545-550.

Kononen, Teuvo. 1988. "Learning Vector Quantization." *Neural Networks 1*, Supplement 1, p303

Plümper, Thomas, Vera E. Troeger, and Eric Neumayer. 2010. "Case Selection and Causal Inference in Qualitative Research" Working Paper, Copy at: http://ssrn.com/abstract=1439868

Yanow, D., Schwartz-Shea P. and Freitas, M.J. (2008). "Case Study Research in Political Science." In A.J. Mills, G. Durepos and E. Wiebe (Eds.), *Encyclopedia of Case Study Research* Sage Publications

Seawright, Jason and John Gerring. 2008. "Case Selection Techniques in Case Study Research: A Menu of Qualitative and Quantitative Options." *Political Research Quarterly* 61, 294-308.

Sekhon, Jasjeet S. 2004. "Quality Meets Quantity: Case Studies, Conditional Probability, and Counterfactuals". *Perspectives on Politics* 2, 281-293.