Computational Tractability of Searching for Optimal Regularities

Ran Eilat[†]

Abstract

Aragones et al. (2005) introduce a model in which a decision maker, who is facing a large knowledgebase, attempts to formulate regularities, or functional rules, that explain an arbitrary variable he bears in mind. Among all possible regularities, the decision maker is looking for the one that is "optimal" from his subjective point of view. Aragones et al. analyze a specific example of optimality criteria and argue that finding such optimal regularity might be a difficult task for a computer, let alone for a human decision maker. Using a slightly generalized version of their model, this work proposes a framework for defining what optimal regularities are, and derives sufficient conditions under which formulating such regularities is computationally difficult, but also sufficient conditions under which this formulation is computationally easy.

1. Introduction

Aragones, Gilboa, Postlewaite, and Schmeidler (2005, hereafter referred to as *AGPS*) discuss the phenomenon of "fact-free learning" in the context of human reasoning and decision theory. They introduce a decision maker who is facing a large knowledgebase, from which he would like to extract empirical regularities to describe the data. After a detailed examination of the knowledgebase, the decision maker formulates a regularity that is optimal from his *subjective* point of view. The term "fact free learning" is used to describe the phenomenon that, without acquiring any new factual knowledge, the decision maker may learn something new by noticing a regularity, which describes the data *better* than the one he had found before[1]. The main result of AGPS is that finding a "best" regularity is a computationally hard problem. Consequently, they argue that the computational complexity of the problem may explain fact free learning: the reason that people do not find the "best" regularities to describe the data in the first place is that this task is computationally intractable.

[†] Ran Eilat, Tel-Aviv University, E-Mail: eilatran@post.tau.ac.il

[1] More generally, fact-free learning would also occur if the decision maker never thought about the problem explicitly, until a certain regularity is pointed out to her. Here we assume that the decision maker is already aware of the learning problem, did her best in finding a regularity that describes the data, but may still be surprised to find that another regularity is a better way to summarize evidence.

To make the above claim well-defined it is required to specify how one models (i) regularities in databases; (ii) "best" regularity; and (iii) a computationally "hard" or "easy" problem. AGPS model regularities by linear regression equations, a choice which is rather natural to economics. They define "hard" problems by the notion of NP-Completeness, which is the most standard definition of high computational complexity in computer science. An *"easy"* problem is then defined to be any problem that is not *hard* (see Appendix B for definitions and short explanation of the terms). However, there are many ways in which one may define the "best" regularity. In this paper we show that the qualitative result of AGPS depends on this definition.

Specifically, AGPS assume that the decision maker prefers regression equations that are, other things being equal, as accurate and as simple as possible. They use R^2 as a measure of accuracy, and the number of explanatory variables as a measure of the complexity of the regression equation. To capture the tradeoff between the two criteria, the decision maker is ascribed a utility function *u(accuracy, complexity)*, increasing in the first argument and decreasing in the second, and the "*optimal regularity*" is that whose accuracy and complexity properties maximize *u*. AGPS prove that this utility function is hard to maximize.

It is not entirely obvious, however, that the number of explanatory variables used in a regression equation is the most natural measure of its complexity. For instance, Lasso, proposed by Tibshirani (1996), is a popular statistical technique for model selection, which penalizes the statistician for using high absolute values of the regression coefficients. That is, the measure of complexity implicit in Lasso is $\sum_{j=1}^{m}|b_j|$ where $b_{j=1..m}$ are the coefficients in the regression. (Specifically, Lasso prescribes to choose the highest R^2 given a certain constraint on this sum) This measure depends on the coefficients $b_{j=1..m}$ in a continuous way, as opposed to the number of coefficients, which is obviously a discontinuous function (around $b_j = 0$ for each *j*). As such, one may consider the Lasso constraint a more reasonable measure of the complexity of a regression, and then wonder whether the results of AGPS would hold for such measure[2].

Incidentally, it will follow from our analysis that Lasso is not subject to the same critique, that is, that it does not give rise to "hard" problems. Thus, if one takes Lasso as a reasonable criterion for model selection by a decision maker, it is not obvious that fact-free learning, to the extent that it exists, follows from computational complexity arguments. As it turns out, however, the continuity of the measure of complexity (with respect to the coefficients of the regression) is not necessarily the crux of the issue.

[2] The example of LASSO was pointed out to AGPS by Douglas Bernheim.

We assume that complexity entails a *burden* or a *cost* incurred by the decision maker, and consider a family of cost functions that are additive across the different coefficients. Specifically, suppose that the cost imposed by a model with $b_{j=1..m}$ is $\sum_{j=1}^{m} \varphi(b_j)$ where $\varphi : \Re \to \Re$ is symmetric around 0. Under some mild assumptions we show the following results: (I) if φ is weakly convex on $\Re_+$, then finding the model that maximizes R^2 given a bound on the cost is a problem that can be solved within polynomial time, that is, an *easy* problem. On the other hand - (II) if φ is weakly concave on R_+ (but not linear) then the same problem becomes *NP-Hard*, and, (III) if φ is non-decreasing on R_+ and discontinuous at 0 the problem is, again, in *NP-Hard*.

Note that we generalize the result of AGPS, but we also provide other results that limit its implications. To see that results (II) and (III) generalize that of AGPS, observe that AGPS use the number of variables as a measure of complexity. That is, they use a function φ that equals 1 for any non-zero value, and 0 at zero. We show that *each* of two properties of this function suffices for the conclusion that the decision maker faces a *hard* problem: first, this φ is weakly concave on $\Re_+$, and, second, it is discontinuous at 0 . On the other hand, our first result limits the implications of AGPS's result by showing that there are many reasonable cost functions for which the problem will be easy. Specifically, if the chosen φ is weakly convex on $\Re_+$, as in the example of Lasso, there is an efficient algorithm for finding the "best" regularity. Roughly stated, the robustness of the result of AGPS depends on whether one believes that a concave or a discontinuous function are more reasonable than a convex one.

One may further question the result of AGPS by wondering whether linear regression is a convincing model of the way individuals generate regularities, as well as by considering the validity of NP-Completeness as an intuitive measure of "hardness". These questions are beyond the scope of this paper.[3]

The rest of this paper is organized as follows: Section 2 lays out the framework and defines the optimization problems; Section 3 proposes sufficient conditions under which those problems are *hard* and sufficient conditions under which those problems are *easy*, in terms of computational complexity; Section 4 concludes. Proofs can be found in the Appendix A. Short review of the concepts of computational complexity can be found in appendix B.

[3] AGPS consider these questions and attempt to justify their modeling choices in these regards. We find that these modeling choices are more standard and less questionable than the measurement of complexity by the number of variables (with non-zero coefficients).

2. Framework

Let $Y \in R^n$ be a column vector denoting an arbitrary economic variable with *n* observations. Assume that a decision maker (henceforth DM) wishes to formulate a rule, in the form of linear regression, which explains the variance of Y using the variance of some other explanatory variables. The explanatory variables are chosen from a large knowledgebase, denoted Π, containing m variables $\{X_1,\ldots X_m\}$ with *n* observations each, formally: $\Pi \in R^{nXm}$, $\Pi = (X_1,\ldots X_m)$, $X_i \in R^n$.

Observe that, since rules are assumed to have the structure of a linear regression, each rule can be fully described by a vector of coefficients $b \in R^m$. Given a vector *b*, let $R^2(\Pi, Y, b)$ be a function that calculates the goodness of fit statistic when Y is regressed on Π and the elements in the vector *b* are used as coefficients[4]. For simplicity of notations let us fix Π and Y throughout the discussion and refer to this function *as if* it depends on *b* only, that is - $R^2(b): \Re^m \to [0,1]$. Henceforth $R^2(b)$ is adopted as the measure for (in-sample) accuracy of any vector $b \in R^m$. We assume also that rules can be characterized by the (subjective) complexity that the DM ascribes to them. We define a *cost* function $\Theta(b): R^m \to R_+$ that determines the measure of complexity that is ascribed to any vector $b \in R^m$. We consider this measure of cost to be additive across the different coefficients. Specifically, suppose that $\Theta(b) = \sum_{j=1}^m \varphi(b_j)$, where b_j are elements of $b \in R^m$ and $\varphi: R \to R_+$ is a non-decreasing function on $[0,\infty)$ and symmetric around 0 (that is, $\varphi(b_j) = \varphi(-b_j)$). We also assume that Θ can be evaluated within constant time and memory space[5].

The objective of the DM is to find a regularity *b* that maximizes the accuracy measure subject to a given cost. Formally,

Problem 1 (Const-Opt): *Given* Y, Π, Θ and $C \in \Re_+$ *find* b^* *such that*

$$b^* = Argmax_{b \in \Re^M}\ R^2(b) \quad s.t. \quad \Theta(b) \le C$$

[4] Formally, $R^2(b) = 1 - e^T e / \sum_{i=1}^n (y_i - \bar{y})^2$, where $e = Y - \Pi b$, y_i is the ith element of *Y*, and $\bar{y}$ is the average of the elements in *Y*.

[5] In other words – we assume that the time and memory required for calculating the cost for any given vector *b* are independent of *b* itself and are finite. This is a very weak technical assumption.

In the economic literature such an optimization problem is known as *constrained regression*. In a quite different form, yet sharing the same solution, it is sometime also referred to as *penalized regression*[6]. Our goal is to propose sufficient conditions under which the solution of *Const-Opt* is "hard" and sufficient conditions under which this solution is "easy", in terms of computational complexity.

At this point it is already quite evident that the explicit functional form of Θ plays a major role in our complexity analysis and that its mathematical characteristics are the core of our proofs and results. The economic literature proposes several models with an explicit Θ, suggesting quite diverse range of interpretations for this constraint. Most of the models can be classifies into two crude categories of interpretation – statistical constraints and cognitive constraints. The class of statistical constraints consists of models in which a penalty on the size of the coefficients is imposed in order to achieve some desirable statistical properties such as a lower variance for predicted values or to reduce the effect of co-linearity in the variables (see a discussion in Hastie *et al.* 2001). For example, Frank and Friedman (1993) introduced the bridge regression, in which the objective is to find a vector $\hat{b}^{bridge}$ such that:

$$\hat{b}^{bridge} = \underset{b}{\operatorname{argmin}}(Y\text{-}\Pi b)^T(Y\text{-}\Pi b) \qquad s.t. \qquad \sum_{j=1}^{m} |b_j|^{\gamma} \leq const$$

Where $\gamma \geq 0$ and b_j is the j^{th} element of *b*. The bridge regression uses the cost function to shrink the coefficients of the regression by imposing a penalty on their size. Under several conditions this shrinkage biases the results but lowers the MSE. Some well known special cases of the bridge include the Ridge regression (Hoerl and Kennard (1970)) where $\gamma = 2$ and the Lasso regression in which $\gamma = 1$. The interest in using the last is constantly growing, mainly since it tends to yield a sparse coefficients vector (depending on the size of *const*), i.e. $\hat{b}^{LASSO}$ typically has relatively few non-zero coefficients (In contrast, ridge regression typically yields $\hat{b}^{RIDGE}$ with all coefficient non-zero). In a way, Lasso uses a continuous procedure to identify and select the most relevant variables in the dataset. It thus suffers less from high variance than discrete selection procedures (that are discussed later). As far as we know, there are no known practical applications for $0 < \gamma < 1$. Our results below show that for these values of γ, the constrained optimization problem is in NP-Complete. This is a

[6] In other disciplines of research it may also be known as *regularized approximation*. See, for example, Boyd and Vandenberghe (2004), Chapter 6, for discussion of engineering-oriented problems.

plausible explanation for the absence of model that use these values of γ. On the contrary, for several years there are well known algorithms that solve both Lasso and Ridge regression efficiently (See, for example, Efron *et al*. 2004 or Osborne, Presnell and Turlach, 1999 for Lasso algorithms and every standard convex optimization algorithm for the Ridge). Also note that solving the degenerate case of $\Theta(b) = const$, that is, the cost is constant and does not depend on the elements of *b*, is equivalent to solving Ordinary Least Squares, which has various polynomial time algorithms for solution (QR decomposition and Cholesky decomposition are only two examples).

The class of cognitive constraints is more subtle and consists of models that try to capture various aspects of cognitive limitations of human perception. For example, consider a DM who, other things being equal, prefers rules with fewer explanatory variables. In their paper, Aragones *et al.* (2005) suggest three explanations for such preference: first, people tend to have more faith in the robustness of relationships that use fewer variables than in those that use more. Second, when fewer variables are involved, people might find it easier to make up explanations for regularity in the data, and third, the more explanatory variables involved, the more chances that the data will not be fully available to the DM. In this case the cost is higher as more variables are involved in the regression. Using the same notation as above, AGPS's problem can be formalized as

$$\hat{b}^{AGPS} = \underset{b}{\operatorname{argmin}}(Y\text{-}\Pi b)^T(Y\text{-}\Pi b) \quad s.t. \quad \textstyle\sum_{j=1}^{m}\varphi(b_j) \le const \quad where \quad \varphi(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x \ne 0 \end{cases}$$

This technique, which is usually referred to as *subset-selection* or *regressors-selection* in the literature, is also a well-studied econometric method, sometimes motivated by pure statistical incentives. AGPS prove that this problem is hard to solve in terms of computational complexity.

We proceed by defining the "accuracy-cost feasible-set", that is the set of *(r,c)* pairs for which there exists a vector *b* whose accuracy measure is (weakly) higher than *r* and cost measure is (weakly) lower than *c*. Formally, let Γ denote the "accuracy-cost feasible-set", i.e.:

$$\Gamma = \Gamma(Y, \Pi, \Theta) \equiv \{(r,c) \mid \quad \exists\, b \in R^m \quad R^2(b) \ge r \quad and \quad \Theta(b) \le c\}$$

We can now ask the question whether a specific accuracy-cost measure is feasible in a given regression setup, formally:

Problem 2 (Feasibility): *Given* $Y, \Pi, \Theta, r \in [0,1]$ *and* $c \in R_+$ determine whether $(r,c) \in \Gamma$

As we shall see, the complexity analysis of problem *Feasibility* will turn out to have a crucial part in the proof for hardness of problem *Const-Opt*, as the following remark suggests:

Remark: Given any Π, Y and Θ, ***if*** *Feasibility* is computationally hard ***then*** *Const-Opt* is computationally hard.

The proof is trivial. Recall that according to our definitions every problem that is not hard is said to be easy. Thus, the remark is equivalent to the statement that ***if*** *Const-Opt* is easy ***then*** *Feasibility* is easy, which follows directly from the definitions.

3. Computational Complexity Results

We will now specify conditions that are sufficient for the feasibility problem to be hard. Consider the following properties of a real value function $\varphi : R \to R_+$:

P-1. Symmetric around *0:* $\varphi(x) = \varphi(-x)$

P-2. Weakly concave, but not linear, on $[0,\infty)$[7]

P-2 also implies that φ is non-decreasing on $[0,\infty)$. When φ is interpreted as a cost function, the first property is motivated by an implicit assumption that there is no essential difference between the "burden" imposed on the DM by a positive or negative regression-coefficients with the same absolute value. This assumption can be slightly relaxed, but it is crucial to our proof that both positive and negative coefficients are penalized. The second property suggests concavity which implies, in particular, that changing a coefficient from 0 to ε impose a marginal cost that is not smaller than the marginal cost of changing it from $\beta > 0$ to $\beta + \varepsilon$. (In fact, as will be clear from the proof, this is the only implication of concavity that is used in the theorem below.) It turns out that these properties are sufficient to imply that problem *Feasibility* is *hard*, or formally:

Theorem 1: *For every* $r \in (0,1]$, *and every function* φ *that satisfies P-1 and P-2, the following problem is NP-Complete:*

(Problem Feasibility) Given $Y, \Pi, \Theta, r \in (0,1]$ *and* $c \in R^+$ determine whether $(r,c) \in \Gamma$.

The proof can be found in appendix A. In essence the theorem argues that if the cost function is the sum of nonlinear weakly concave functions of the estimated paramters, then deciding whether an accuracy-cost pair is feasible cannot be generally done within time which is polynomial in the size of

[7] This could be relaxed such that φ is required to be non-decreasing, weakly concave, but not linear, on $[0, \overline{a}]$ for **for any** $\overline{a}$ such that $\overline{a} \in (0,\infty)$.

the data[8]. Moreover, the result is independent of the exact functional form of φ and is also independent of the required measure of accuracy (it is hard for every $r \in (0,1]$).

The domain of functions φ that yields this complexity result can be expanded further. For example, consider a DM that is facing a cost function which is discontinuous at 0. Such cost function might reflect a one-time effort needed to collect data and maintain a variable in the database. Another interpretation for such discontinuity might be a cognitive effort required to evaluate algebraic calculations with a non-zero coefficient when using regularities for prediction. Formally, consider the following properties of a real value function $\varphi : R \rightarrow R_+$:

Q-1. Symmetric around *0:* $\varphi(x) = \varphi(-x)$

Q-2. Non-decreasing on $[0,\infty)$

Q-3. Discontinuous at 0, that is - $\lim_{\varepsilon \to 0^+} \varphi(\varepsilon) > \varphi(0)$

Note that no assumptions are made about the second derivative - it might be positive, negative or it might not exist at all. It turns out, however, that these properties are also sufficient for implying that problem *Feasibility* is *hard*, or formally:

Proposition 2: *For every* $r \in (0,1]$*, and every function* φ *that satisfies Q-1,2,3, the following problem is NP-Complete:*

(Problem Feasibility) Given $Y, \Pi, \Theta, r \in (0,1]$ *and* $c \in R^+$ determine whether $(r,c) \in \Gamma$.

The proof can be found in appendix A.

An immediate consequence of Theorem 1 and Proposition 2:

Corollary 3: *for every function* φ *that satisfies P-1,2 or Q-1,2,3 the following problem is NP-Hard:*

(*Problem Const-Opt)*: *Given* Y, Π and $C \in \Re_+$ *find* b^* *such that*

$$b^* = Argmax_{b \in \Re^M} \ R^2(b) \quad s.t. \quad \Theta(b) \le C$$

Recall that AGPS's problem can be formalized as if they use a cost function $\varphi(x)$ that equals 0 at $x=0$ and 1 for $x \neq 0$. Notice that such φ satisfies both *Q-1,2,3* and *P-1,2*. Thus our result is a

[8] Note, however, that this result is stated in terms of "worst-case" analysis, as all problems in NP-Complete are. That is – there might exist a dataset for which solving the problem is polynomial.

generalization of the result of AGPS and it is clearly negative – it argues that it is sufficient for φ to be nonlinear weakly concave, or discontinuous at 0, in order to make the worst-case solution of the optimality problem non-polynomial in the size of the data. In other words, under rather weak assumptions, finding the optimal solution is generally a hard task for computers, let alone for human decision makers.

We now turn to specify conditions that are sufficient for the problem *Const-Opt* to be easy. We retain the assumption that Θ is additive across the coefficients. As an example, consider an economist that finds the Lasso constraint to be the most adequate measure of complexity, for instance due to its simplicity and continuous nature. Note that the cost function φ that is implicit in Lasso satisfies neither P-2 nor Q-3. In this case, as we have indicated before, and now we state explicitly, a polynomial algorithm for solution exists,

Proposition 4: *For every* $Y, \Pi, \Theta, r \in [0,1]$ *and* $C \in \Re_+$ *if* $\varphi(b_j) = |b_j|$ *the following problem has a polynomial time algorithm for solution:*

(*Problem Const-Opt*): *find* b^* *such that* $b^* = Argmax_{b \in \Re^M}\ R^2(b) \quad s.t. \quad \Theta(b) \le C$

Detailed algorithms for solution along with discussion about their efficiency may be found in Efron *et al*. (2004) or Osborne, Presnell and Turlach (1999). Note that this remark limits the scope of APGS results as it shows that choosing a different cost measure turns the problem to be computationally easy. It turns out, however, that this result can be further generalized for a wide range of functions φ , using algorithms of convex optimization.

Consider the following properties of a real value function $\varphi : R \to R_+$:

S-1. Symmetric around *0:* $\varphi(x) = \varphi(-x)$

S-2. Weakly convex on $(-\infty, \infty)$

S-3. Bounded by a polynomial of β_j

S-4. Its subgradient can be calculated within polynomial time (it exists since the convexity of φ on $(-\infty, +\infty)$ implies continuity)

The motivation for S-1 is the same as for P-1. S-2 requires φ to be convex, suggesting an increasing marginal cost of coefficients (i.e. – the larger the coefficient is, the more costly it would be to increase its value). It might reflect, for example, the preference of a large number of small coefficients to a small number of large coefficients. Such preference might be motivated, for example, by statistical considerations. S-3 and S-4 are very weak technical conditions and both can be relaxed to some extent as many ad-hoc optimization techniques that exist in the literature do not require them. We claim that if φ satisfies S1-S4 then Const-opt becomes an "easy" optimization problem, and formally:

Proposition 5: *For every* $Y, \Pi, \Theta, r \in [0,1]$ *and* $C \in \Re_+$ *and every* φ *that satisfies S-1-4, the following problem has a polynomial time algorithm for solution:*

(*Problem Const-Opt)*: *find* b^* *such that* $b^* = Argmax_{b \in \Re^M} \; R^2(b) \quad s.t. \quad \Theta(b) \le C$

The polynomial time solution can be achieved using the *ellipsoid method* which is a well studied algorithm from the literature of *convex optimization* (see, for example, Ben-Tal and Nemirovski (2001), Chapter 5). This method is applicable when both the objective function and the constraint of the optimization problem are convex and "well-behaved" (i.e. *polynomially computable* and of *polynomial growth*) which is ensured by S-1 to S-4. It also requires that the problem would have a *bounded feasible set*, a property that is satisfied by *Const-Opt*, and proved in appendix A[9].

4. Discussion

Economic agents cope with a world that is immensely complex but that is, nonetheless, highly patterned. The capability of seeking patterns in data, or finding regularities in a knowledgebase, is maybe the most fundamental aptitude required from a decision maker. It goes without saying that an economic agent that lacks this ability would be able to predict, explain, and understand very little; he would therefore have no rational basis on which to choose his actions. Typically, however, the agent’s problem is the opposite one: when dealing with complex environments economic agents may recognize a large number of possible patterns, differing from each other in many parameters, and particularly in their rate of accuracy. It is then reasonable to assume that the more accurate regularity is preferred. Our work relies on the assumption that besides accuracy, the decision maker prefers his regularities to be also as simple as possible.

While preference for accuracy is quite evident, simplicity is a bit more subtle. Back in the 14th century, in the famous Occum's razor, William of Ockham suggested that, given two equally valid explanations for a phenomenon, one should embrace the less complicated one. About six hundred

[9] Though it is polynomial, the *ellipsoid method* is not the most efficient algorithm known to date. For example, if φ is also differentiable on $(-\infty,+\infty)$, then much more efficient interior point optimization techniques may be applied (see, for example, Boyd and Vandenberghe (2004), chapter 11).

years later, Albert Einstein has been attributed with the remark that "Everything should be made as simple as possible, but not simpler". In between, as well as in the past few decades, scientists and psychologists tend to agree that simplicity is a desirable property when patterns of data are considered (see, for example, Chater 1999).[10] They do not agree, however, on a unique measure for quantifying simplicity. As the above results indicate, this quantification is crucial for analyzing the tractability of the problem.

Following AGPS, this work proposed a framework in which a tradeoff between accuracy and simplicity arises. Consequently, when a decision maker attempts to find the most accurate regularity in a database, given a certain measure of simplicity, he faces a non-trivial optimization problem. In some cases, as was earlier argued, this challenge turns out to be a computationally intractable problem. We adopt the notion of cost functions to measure simplicity – the simpler regularities are, the less cost they involve. We considered three types of cost functions – concave, convex and discontinuous at 0, each implying different computational complexity. The question of which of these types is more reasonable depends on the aspect of simplicity that one wishes to capture.

Convex cost functions ensure existence of an efficient algorithm for the solution of the optimization problem. Such functions are ascribed to a decision maker who prefers a large number of small coefficients to a small number of large coefficients (e.g. Lasso). Other things being equal, such a decision maker is assumed to prefer a regression with smaller partial derivatives with respect to the variables of the regression. When human reasoning and decision making are considered, this mathematical preference is naturally translated to preference for regularities in which fluctuations in a single variable do not affect the result "too dramatically". For example, consider a decision maker who uses regularities as policy functions. That is, given many state variables he evaluates a regression and takes actions according to the result. Assume further that he favors stability, that is, he wishes to reduce the magnitude of changes in his behavior across time due to environmental changes. Such a decision maker would prefer *smoother* policy functions to more volatile ones. A convex function is then suitable to model this cost since it yields a regression with a relatively small coefficient for each variable, and thus a policy function that reacts in a relatively moderate manner to changes in the environment.

On the other hand, concave functions and functions that are discontinuous at 0 are used to model situations where, everything else being equal, the decision maker prefers a small number of relatively large coefficients to a large number of small coefficients. Concave functions are then suitable since

[10] Preference for simplicity in philosophy of science has a normative flavor, as in Occum's razor argument. In psychology, simplicity is offered as a descriptive theory of people's preferences between theories. As a descriptive theory, simplicity was mentioned by Wittgenstein (1922, 6.363) at the latest.

they propose "decreasing marginal cost" for coefficient units. Functions that are discontinuous at 0 may also be suitable if there is a "fixed cost" associated with including a variable in the regression. For example, consider a decision maker who, due to cognitive limitations, seeks to minimize memory load, that is, the number of coefficients that are different from 0. Occum's razor gives rise to this type of preferences, and it was also used as the motivation for the model proposed by AGPS. Another example is a decision maker who wishes to construct a regularity to be used for *fast and frugal* predictions, that is, predictions where accuracy is sacrificed for the sake of faster evaluation, which is achieved by using only a small number of variables in the equation.

To conclude, we believe that the implications of our work are twofold. In a purely theoretic context we proposed sufficient conditions under which a wide range of optimization problems is difficult, in terms of computational complexity. This result does not rely on any economic assumption and is therefore applicable in every discipline where constrained regression problems are used.

In the context of decision making we both generalized and outlined limitations on the result of AGPS. On the positive side, we proposed sufficient conditions under which the optimization problem is *computationally easy*. This result should be interpreted carefully, since we do not claim that such problems are necessarily easy to solve by a human decision maker. We do claim, however, that in those cases fact free learning cannot be explained by the type of arguments suggested by AGPS.

On the negative side we proposed sufficient conditions under which the optimization problem is *NP-Hard*. We do not claim that the computational implications of this result are necessarily a tight upper bound on the human cognitive capabilities. We do adopt, however, the assumption that those problems are practically infeasible for human decision makers.

The computational complexity of certain problems does not mean that individuals always do poorly on these problems. It is quite possible that, learning from each other, economic agents can improve the solutions that they find on their own. In our case, it suffices that a single individual, such as a scientist or a political leader, point to a certain regularity, for this regularity to be adopted by many individuals who have similar cost functions. Thus, to the extent that concave or discontinuous cost functions are reasonable, computational complexity may help explain the phenomenon of fact-free learning, as well as the role of social learning.

Appendix A - Proofs

Proof of Theorem 1 and Proposition 2

Remark A.1: The proof proposed here is a generalization of a proof proposed by Aragones *et al* (2005)

Remark A.2: For a brief review of the concepts of the theory of complexity refer to appendix B.

Note that for every function $\Theta = \sum \varphi(b_i)$ and every $r \in (0,1]$ the *feasibility problem* can be stated formally as:

Problem Feasibility:

Input: (1) $\Pi \in R^{n\,X\,m}$ (2) $Y \in R^{n}$ (3) $c \in R^{+}$

Output: Is there a vector $b \in R^{m}$ such that $R^2(b) \geq r$ and $\Theta(b) \leq c$? Yes / No

Consider also the following NP-Complete decision problem (see Karp 1972):

Problem Exact Cover:

Input: (1) a finite set $S \subset R$ (2) a set $\Omega \subseteq P(S)$ (set of subsets of S)

Output: Are there pairwise disjoint subsets in Ω whose union equals *S*? Yes / No

To show that Feasibility is NP-Complete it is sufficient to show that:

1. *Feasibility* is in NP
2. There exists a polynomial reduction from *Problem Exact Cover* to *Problem Feasibility*

The first requirement is almost trivially satisfied in this context. Given an arbitrary vector $b \in R^{m}$ we have to determine whether $R^2(b) \geq r$ and $\Theta(b) \leq c$ within polynomial time. Since calculating $R^2(b)$ and $\Theta(b)$ can be done within polynomial time, then we are done.

The second requirement is a bit more complex. The concept of polynomial reduction can be thought of as a construction of a translation function F, such that F takes input to the problem of *Exact Cover* and translates it into input for problem *Feasibility:* $F:\{s,\Omega\} \rightarrow \{\Pi, Y, c\}$ in such a way that:

1. The translation algorithm is polynomial in the size of the input
2. $(Feasibility) \circ F = (Exact\ Cover)$, that is - the two functions return the same yes/no result for every input.

Construction of such *F* would lead to the result that ***if*** there exists a polynomial time algorithm for solving *Feasibility* ***then***, using the function *F*, there exists polynomial time algorithm for solving *Exact Cover*. In other words – *Feasibility* would then be at least as hard as *Exact Cover*, a problem for which there does not exists, to date, any known algorithm of solution.

In the first part of the proof we propose a construction of such F. In the second part we show that $(Feasibility) \circ F = (Exact\ Cover)$ holds.

Part I: Construction of F

<u>*Proposition A.3*</u>: for any non-decreasing real-valued φ function that satisfies either *P-1,2* or *Q-1,2,3*, there exists a constant $\bar{a} \in R_+$ such that $\varphi(0)+\varphi(\bar{a}) < \varphi\big((1-\lambda)\bar{a}\big)+\varphi(\lambda\bar{a})$ $\forall\ \lambda \in R_+ /\{0,1\}$ (Short proof follows later).

Let there be given *S* and Ω. Assume without loss of generality that $S=\{1,\ldots,s\}$ and that $\Omega=\{S_1,\ldots,S_l\}$ (where s,l are natural numbers). We construct $n=2(s+l+1)$ observations of $m=2l$ variables in the following way:

1. Let $Y \in R^{2(s+l+1)}$ be a column vector and denote its values by $(y_i)_{i\le n}$.

   Let $M \ge 0$ be a constant to be specified later

   For $i \le s+l$ set $y_i = \bar{a}$ ($\bar{a}$ is taken from the definition of *proposition A.3*)

   For $i = s+l+1$ set $y_{s+l+1} = M$

   For $i > s+l+1$ set $y_i = -y_{i-(s+l+1)}$

2. Let $\Pi \in R^{2(s+l+1)x2l}$. Observe that Π is consisted of *2l* column vectors, and denote them by $X_1,\ldots,X_l$ and $Z_1,\ldots,Z_l$. Their corresponding values will be denoted $(x_{ij})_{i\le n, j\le l}$, $(z_{ij})_{i\le n, j\le l}$.

   For $i \le s$ set $x_{ij}=1$ if $i \in S_j$ and $x_{ij}=0$ if $i \notin S_j$;

   For $i \le s$ set $z_{ij}=0$;

   For $s < i \le s+l$ set $x_{ij}=z_{ij}=1$ if $i=s+j$ and $x_{ij}=z_{ij}=0$ if $i \ne s+j$;

   For $i = s+l+1$ set $x_{s+l+1,j}=z_{s+l+1,j}=0$;

   For $i > s+l+1$ set $x_{ij}=-x_{i-(s+l+1),j}$ and $z_{ij}=-z_{i-(s+l+1),j}$;

3. Let *C* be defined in the following way: $C = l\cdot\big(\varphi(0)+\varphi(\bar{a})\big)$

EXAMPLE:

$$
\begin{array}{c|cccc:cccc|}
 & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{X_{j=1..l}}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{Z_{j=1..l}}} \\
 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
l & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hdashline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
s & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
l & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\
 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
\hdashline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{c|c|}
 & Y \\
 & \bar{a} \\
s & \bar{a} \\
 & \bar{a} \\
\hdashline
 & \bar{a} \\
l & \bar{a} \\
 & \bar{a} \\
 & \bar{a} \\
\hdashline
 & M \\
\hdashline
 & -\bar{a} \\
s & -\bar{a} \\
 & -\bar{a} \\
\hdashline
 & -\bar{a} \\
l & -\bar{a} \\
 & -\bar{a} \\
 & -\bar{a} \\
\hdashline
 & -M
\end{array}
$$

The matrices on the left are an illustrative example for the construction of Π and Y. This example shows how the construction is done for *S* that consists of 3 elements, i.e. $S = \{1, 2, 3\}$ and Ω that consist of 4 subsets of S as follows: $\Omega = \bigcup_{i=1}^{4} S_i$ and $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}$, $S_3 = \{3\}$, $S_4 = \{1\}$

Note that in this example a (unique) exact cover exists as $S_1 \cup S_3 = S$ and $S_1 \cap S_3 = \varnothing$.

Remark A.4: The bottom half of the matrix Π, as well as the bottom half of the vector Y, are the negatives of the respective top halves. This implies that each of the variables $X_1, \ldots, X_l$, $Z_1, \ldots, Z_l$ and Y has a mean of zero and therefore a linear regression of *Y* on Π yields an intercept of zero.

Remark A.5: The translation function *F* is polynomial in the size of the input.

Denote $R^2(b)$ as the goodness-of-fit measure (R-squared) of a vector *b* in the ordinary linear regression setup where *Y* is regressed on Π. Formally, $R^2(b) = 1 - e^T e / \sum_{i=1}^{n} (y_i - \bar{y})^2$, where $e = Y - \Pi b$, y_i is the i$^{\text{th}}$ element of Y, and $\bar{y}$ is the average of the elements in Y.

Remark A.6: The vector b that maximizes $R^2(b)$ is independent of *M* (since the values of the rows *s+l+1* and 2*(*s+l+1)* in Π are all zero).

Let $\hat{\Pi}$ be equal to Π without observations $s+l+1$ and $2(s+l+1)$. Define $\hat{R}^2(b)$ to be the goodness-of-fit measure (R-squared) of a vector *b* in the OLS regression setup where *Y* is regressed on $\hat{\Pi}$.

Lemma A.7 (R-squared-equivalence): *There exists* $\hat{M}$ *such that* $\forall b \quad \hat{R}^2(b) = 1 \quad \Leftrightarrow \quad R^2(b) \geq r$

A constructive proof, which finds such $\hat{M}$ in *O*(1), follows later.

Set $M = \hat{M}$ and observe that the following two problems are equivalent:

1. *Does there exist a vector* $b \in R^m$ *such that* $\Theta(b) \leq c$ *and* $R^2(b) \geq r$ *?*
2. *Does there exist a vector* $b \in R^m$ *such that* $\Theta(b) \leq c$ *and* $\hat{R}^2(b) = 1$ *?*

Part II: Show that $(Feasibility) \circ F = (Exact\ Cover)$

We wish to show that:

a) ***If*** the output of $(Feasibility) \circ F$ is *no* ***then*** the output of *Exact cover* is *no*

b) ***If*** the output of $(Feasibility) \circ F$ is *yes* ***then*** the output of *Exact cover* is *yes*

Or Equivalently:

S has an exact cover from Ω $\Leftrightarrow$ There exists a vector $b \in R^m$ such that $\Theta(b) \leq c$ and $R^2(b) \geq r$

This is, in turn, equivalent to (using *lemma A.7*):

S has an exact cover from Ω $\Leftrightarrow$ There exists a vector $b \in R^m$ such that $\Theta(b) \leq c$ and $\hat{R}^2(b) = 1$

Part II-a

We wish to show the following direction: *S* has an exact cover from Ω $\Rightarrow$ There exists a vector $b \in R^m$ such that $\Theta(b) \leq c$ and $\hat{R}^2(b) = 1$

Assume that such an exact cover exists. That is, assume that there is a set $J \subseteq \{1 \ldots l\}$ such that $\{S_j\}_{j \in J}$ constitutes a partition of *S*. Formally, $\bigcup_{j \in J} (S_j) = S$ and $S_j \cap S_k = \varnothing \quad \forall j,k \in J, j \neq k$.

Let $(\beta_j)_{j \leq l}$ denote the coefficients of $(X_j)_{j \leq l}$ and $(\gamma_j)_{j \leq l}$ denote the coefficients of $(Z_j)_{j \leq l}$. For $j \in J$ set $\beta_j = \bar{a}$ and $\gamma_j = 0$, and for $j \notin J$ set $\beta_j = 0$ and $\gamma_j = \bar{a}$.

For every $i \leq s$ the following equality holds:

$$\sum_{j=1}^{l} \beta_j x_{ij} + \sum_{j=1}^{l} \gamma_j z_{ij} = \sum_{j=1}^{l} \beta_j x_{ij} = \sum_{j \in J} \bar{a} x_{ij} = \bar{a} = y_i$$

The first equality holds since $z_{ij} = 0$ for every $i \leq s$. The second equality holds due to the values assigned to β_j. The third equality follows from the construction of Π since for every $i \leq s$ there exist exactly one $j \in J$ for which $x_{ij} \neq 0$ (that is the j for which $i \in S_j$).

For every $s < i \leq s + l$ the equality:

$$\sum_{j=1}^{l} \beta_j x_{ij} + \sum_{j=1}^{l} \gamma_j z_{ij} = \beta_j + \gamma_j = \bar{a} + 0 = y_i$$

follows from the construction of Π and the assignment of β_j and γ_j

It is also clear that the same arguments hold for the bottom half of $\hat{\Pi}$, Thus $\hat{R}^2(b) = 1$.

To see that $\Theta(b) \leq c$ observe that since for every $j \leq l$ either $(\beta_j = \bar{a}, \gamma_j = 0)$ or $(\beta_j = 0, \gamma_j = \bar{a})$ then for every $j \leq l$ the following equality holds

$$\varphi(\beta_j) + \varphi(\gamma_j) = \varphi(\bar{a}) + \varphi(0)$$

and therefore

$$\Theta(b) = \sum_{j=1}^{2l} \varphi(|b_j|) = \sum_{j=1}^{l} \left(\varphi(|\beta_j|) + \varphi(|\gamma_j|)\right) = l \cdot \left(\varphi(\bar{a}) + \varphi(0)\right) = c.$$

QED.

Part II-b

We wish to show the following direction: There exists a vector $b \in R^m$ such that $\Theta(b) \leq c$ and $\hat{R}^2(b) = 1 \Rightarrow$ *S* has an exact cover from Ω

Assume that there exists a vector b such that $\Theta(b) \leq c$ for which $\hat{R}_b^2 = 1$. Denote the coefficients of *X* and *Z* in the regression as $(\beta_j)_{j \leq l}$ and $(\gamma_j)_{j \leq l}$ correspondingly, as before. Note that since $\hat{R}^2(b) = 1$ then the rows $s < i \leq s + l$ in $\hat{\Pi}$ imply that for every $1 \leq j \leq l$ the equation $\beta_j + \gamma_j = \bar{a}$ holds.

Claim: $\Theta(b) \leq c$ and $\beta_j + \gamma_j = \bar{a}$ for every $1 \leq j \leq l$ imply that either $\beta_j = \bar{a}$ or $\beta_j = 0$.

Proof: According to *proposition A.3*, for every $1 \leq j \leq l$, the minimal value of $\varphi(\beta_j) + \varphi(\gamma_j)$, when $\beta_j + \gamma_j = \bar{a}$, is acquired when either $(\beta_j = 0, \gamma_j = \bar{a})$ or $(\beta_j = \bar{a}, \gamma_j = 0)$. It is clear, therefore, that the minimal value of the cost function $\Theta(b)=\sum_{j=1}^{l}\left(\varphi(\beta_j)+\varphi(\gamma_j)\right)$ can be not less than $l\cdot\left(\varphi(\overline{a})+\varphi(0)\right)$, which is exactly c. Therefore, existence of any β_j that equals neither $\overline{a}$ nor 0 contradicts $\Theta(b)\leq c$.

Define $\hat{J}$ to be a set such that $\hat{J}=\{j\mid b_j=\overline{a}\}$. $\hat{R}^2(b)=1$ implies that for every $i\leq s$ the equality $\sum_{j=1}^{l}\left(x_{ij}\beta_j\right)=\overline{a}$ holds. Therefore, for every $i\leq s$ it is also true that $\sum_{j\in\hat{J}}\left(x_{ij}\cdot\overline{a}\right)=\overline{a}$ which, in turn, implies that for every $i\leq s$ there exists exactly one $j\in\hat{J}$ for which $x_{ij}=1$. Recall that $\hat{\Pi}$ was defined such that $x_{ij}=1$ if $i\in S_j$ and therefore we conclude that **for every** $i\leq s$ there exists **exactly one** $j\in\hat{J}$ for which $i\in S_j$. Thus, $\{S_j\}_{j\in\hat{J}}$ is well defined and constitutes an exact cover of S. QED.

Proof of Proposition A.3

First note that for every function φ that satisfies *P-1,2* or *Q-1,2,3* there exists $\overline{a}\in(0,\infty)$ such that the line segment from $(0,\varphi(0))$ to $(\overline{a},\varphi(\overline{a}))$ lies strictly beneath the graph of φ on the interval $(0,\overline{a})$. If *P-2* is satisfied than this property is true by the definition of concavity, and if *Q-2,3* are satisfied then this property is true since $\varphi(0)$ is a global minimum.

We wish to show that for such $\overline{a}$ the following inequality holds:

$$\varphi(0)+\varphi(\overline{a})<\varphi((1-\lambda)\overline{a})+\varphi(\lambda\overline{a})\quad\forall\ \lambda\in R_+/\{0,1\}.$$

If $\lambda<1$:

Let $f(x)=((\varphi(\overline{a})-\varphi(0))/\overline{a})*x+\varphi(0)$, that is the equation that represents the line segment that joins $\varphi(0)$ and $\varphi(\overline{a})$. $f(x)$ is linear and therefore - $f(0)+f(\overline{a})=f(\lambda\overline{a})+f[(1-\lambda)\overline{a}]$. Note that $f(0)=\varphi(0),\quad f(\overline{a})=\varphi(\overline{a})$ and that $f(x)<\varphi(x)\ \ \forall x\in(0,\overline{a})$, thus

$\varphi(0)+\varphi(\overline{a})<\varphi((1-\lambda)\overline{a})+\varphi(\lambda\overline{a})\ \ \forall\ \lambda\in(0,1)$.

If $\lambda>1$:

Both *P-1,2* and *Q-1,2,3* imply that φ has a unique minimum at 0. Since in both cases φ is non-decreasing then $\varphi(\lambda\overline{a})\geq\varphi(\overline{a})$ and since it is also symmetric then $\varphi((1-\lambda)\overline{a})=\varphi((\lambda-1)\overline{a})>\varphi(0)$. Summing up both inequalities yield: $\varphi(0)+\varphi(\overline{a})<\varphi((1-\lambda)\overline{a})+\varphi(\lambda\overline{a})$.

QED.

Proof of Lemma A.7 (R-squared-equivalence)

Using the notations defined above, we wish to find $\hat{M}$ *such that* $\forall b \quad \hat{R}^2(b)=1 \quad \Leftrightarrow \quad R^2(b)\geq r$

Denote $E\hat{S}S$ and $T\hat{S}S$ the explained sum of squares and the total variance of *Y*, respectively, of the regression of *Y* on $\hat{\Pi}$ and denote *ESS* and *TSS* as the variances of the regression of *Y* on Π. Thus, $R(b)^2 = ESS/TSS$ and $\hat{R}^2(b) = E\hat{S}S/T\hat{S}S$. Observe that $T\hat{S}S = 2(s+l)\overline{a}^2$ and $TSS = 2(s+l)\overline{a}^2 + 2M^2$. Also, $E\hat{S}S = ESS$ and is independent of *M*.

Define $\hat{M} = \overline{a} * \sqrt{(s+l)} * \sqrt{(1-r)/r}$. Note that $\hat{M}$ is well defined for every $r \in (0,1]$, and that it can be calculated in *O*(1).

(Direction only if): Suppose $\hat{R}^2(b)=1$. Therefore $TS\hat{S} = E\hat{S}S(= ESS)$ and $R^2(b)$ is then given in the equation

$$R_b^2 = \frac{T\hat{S}S}{TSS} = \frac{2(s+l)\overline{a}^2}{2(s+l)\overline{a}^2 + 2M^2} = \frac{\overline{a}^2(s+l)}{\overline{a}^2(s+l) + \overline{a}^2(s+l)(1-r)/r} = r$$

QED.

(Direction if): Suppose $r \leq R^2(b)$, that is

$$r \leq R_b^2 = \frac{ESS}{TSS} = \frac{E\hat{S}S}{TSS} = \hat{R}_b^2 * \frac{T\hat{S}S}{TSS} = \hat{R}_b^2 * \frac{2\overline{a}^2(s+l)}{2\overline{a}^2(s+l)+2M^2} = \frac{\hat{R}_b^2 * \overline{a}^2(s+l)}{\overline{a}^2(s+l)+\overline{a}^2(s+l)(1-r)/r} = \hat{R}_b^2 * r$$

Eliminate r from both sides to get $1 \leq \hat{R}^2(b)$, which, in turn, implies a strict equality, that is $\hat{R}^2(b)=1$, QED.

***Proof of Proposition** 7* **(The feasible set of *Const-Opt* is bounded)**

Consider the Const-Opt problem:

Given Y, Π, Θ and $C \in \Re_+$ *find* b^* *such that* $b^* = Argmax_{b \in \Re^M} \; R^2(b) \quad s.t. \quad \Theta(b) \leq C$.

Calculate b^{OLS}, which is the vector of coefficients that maximizes the unconstrained least squares problem, that is $b^{OLS} = Argmax_{b \in \Re^M} \; R^2(b)$ and set $Q \equiv \sum_{j=1}^{m} \varphi(b_j^{OLS})$. Recall that $\varphi : R \to R_+$ is symmetric and convex (and thus continuous) and therefore either:

a. $\varphi(b_j)$ is strictly increasing in b_j

b. $\varphi(b_j)$ is constant.

If $\varphi(b_j)$ is constant then the problem is degenerated and can be solved using standard polynomial OLS techniques. On the other hand, if $\varphi(b_j)$ is strictly increasing in b_j then φ^{-1} is well defined. Set $C\_BOUND \equiv \varphi^{-1}(Q)$.

Note that since the unconstrained least squares coefficients attain the highest R-squared then no optimal solution can exceed its cost. Thus C_BOUND is the upper bound for every element in any optimal b^*. Let V_T be a vector with *m* elements, each equals to C_BOUND, and observe that the feasible set is then bounded within an Euclidean ball, centered at the origin with radius of the L_2 norm $\left\|V_T\right\|_2$. QED.

Appendix B – concepts from the theory of computational complexity

The following appendix gives a short introduction to the theory of computational complexity of algorithms. It is not intended to be used as an extensive survey of the definitions and results of the field, but rather as an orientation guide to the unfamiliar reader. Consequently we might have abused notation or use informal definitions in order to allow readability and clarity of the concepts. For extensive review of this field the reader may refer for one of many available resources and textbooks (for example see Garey and Johnson 1979).

Generally speaking, computer science deals with finding solutions to problems that are defined in some formal language. The method of finding a solution to a problem is usually called an *algorithm*. The theory of computational complexity deals with classifying problems into complexity sets according to the algorithms that are used for their solution. Not all problems have an algorithm that solves them; such problems are called *undecidable*. Our discussion, however, focuses on those problems for which a solution algorithm exists; such problems are called *decidable*.

Given an algorithm, we are interested in the amount of time it requires for evaluation, where time is usually approximated by the number of the steps it takes to find the solution. The *time complexity function* expresses the time requirements by giving, for each possible input length, the largest amount of time needed by the algorithm to solve a problem instance of this size. Denote the *input size* as *n* and the *time complexity function* as *f(n)*.

The *"Big O"* is a mathematical notation used to describe the asymptotic behavior of functions. Its purpose is to characterize functions' behavior for very large inputs in a simple but rigorous way that enables comparison to other functions. More precisely, the symbol *O* is used to describe an asymptotic upper bound for the magnitude of a function in terms of another, usually simpler, function. Formally, a function *f(n)* is $O(g(n))$ whenever there exists a constant *c* such that $|f(n)| \leq c \cdot |g(n)|$ for all values $n \geq n_0$. A *polynomial time algorithm* is defined to be one whose time complexity function is $O(p(n))$ for some polynomial function *p*. An *intractable problem* is defined to be one that does not have any polynomial time algorithm that solves it. Simple calculations show that finding a solution to intractable problems may take very long time even for modern computers that perform millions of operations per second. Note that while they are hard to solve, intractable problems are still decidable, since there exists an algorithm that solves them.

The theory of computational complexity classifies the universe of decidable problems into several subsets. The subset *P* (stands for *polynomial*) consists of all the decision problems that can be solved by a polynomial time algorithm. For example – the question whether, given a vector of numbers, there exist two numbers whose sum exceeds a given constant (solvable in $O(n^2)$, where *n* is the size of the vector). The subset *NP* (stands for Nondeterministic Polynomial) consists of all the decision problems for which there exists a polynomial time algorithm that *verifies* a solution (but not necessarily finds it). That is, *given* a solution the algorithm verifies whether it is correct within polynomial time, without worrying about how hard it might be to find this solution. For example, suppose that *p* is a very large number that has only two factors, both are prime, *r* and *q*. The problem is to find *r* and *q* when only *p* is given. It turns out that the problem is very hard to solve when *p* is large. However, given *r* and *q* one can easily verify whether they constitute a correct solution by calculating *r**q* and checking whether it equals *p*.

It is easy to see that $P \subseteq NP$. However, the question of whether $P = NP$ is one of the most important open questions in computer science these days. It is hard to overestimate the implications of a proof that would show that this equality holds. However, no such proof has been proposed to date and given the current state of knowledge it seems that the converse is true (that is - $P \subset NP$). Thus, if $P \neq NP$ then the set NP / P is not empty, and there are problems whose solution cannot be found easily, but could be verified so.

Let Q_1 and Q_2 denote two problems with a yes-no answer. We say that Q_1 is *reducible* to Q_2 if we can use Q_2 to solve Q_1. That is – if we can find a transformation *f* for which $f(Q_2)$ and Q_1 return the same result for *every* input. If such a function *f* exists then it is called a *reduction* and denoted by $Q_1 \propto Q_2$. If, in addition, *f* is polynomial then the reduction is called a *polynomial reduction*. Intuitively, in this case solving Q_1 cannot be harder than solving Q_2.

A problem is said to be in the set of *NP-Complete* if (i) it is in NP (ii) all other problems in NP can be reduced to it. In a sense, the problems in the class of NP-Complete are the hardest problems of NP. If one succeeds in finding an algorithm that solves one of those problems using a polynomial time algorithm, then all problems in NP can be solved within polynomial time, thus P=NP. The first problem that was identified to be in NP-Complete is called Satisfiability (a result that is well known as "Cook's theorem"). During the years of research many more problems have been identified to be in this class, for a partial list see Garey (1979). It follows that showing that a problem is in NP-Complete indicates that it is at least as hard as many other problems that many scientist over many years could not find an efficient (polynomial) algorithm for their solutions. Practically, In order to show that a new problem is in NP-Complete one has to show that (i) it is in NP (ii) it has a *polynomial reduction* from at least one problem that is already known to be in NP-Complete. Problems that satisfy only the second property are said to be in the class of *NP-Hard*. That is – *NP-Hard* problems are reducible from all the problems in *NP*, but are not necessarily themselves in *NP*.

References

- **Aragones, E. Gilboa, I., Postlewaite, A. and Schmeidler, D (**2005)., "Fact-Free Learning", *The American Economic Review*, Vol. 95, No. 5
- **Ben-Tal, A. and Nemirovsky A.,** (2001) *Lectures on Modern Convex Optimization*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
- **Boyd, S. and Vandenberghe, L.** (2004)**,** *Convex Optimization***,** Cambridge University Press, Cambridge, UK
- **Chater, N.**, (1999) "The search for simplicity: A fundamental cognitive principle?" *Quarterly Journal of Experimental Psychology*, 52A, 273-302
- **Efron, B., Hastie, T., Johnstone, I. and Tibshirani R.,** (2004) "Least Angle Regression", *Annals of Statistics*, 32, no. 2 (2004), 407–499.
- **Garey, M. and Johnson, D.,** (1979), *Computers and Intractability*, Freeman, New York.
- **Hastie, T., Tibshirani, R., Friedman, J.** (2001), *The Elements of Statistical Learning*, New York, NY:Springer
- **Hoerl, A. and Kennard, W.,** (1970), "Ridge Regression: Biased Estimation for Nonorthogonal Problems", *Technometrics*, Vol. 12, No. 1
- **Karp, R.**, (1972), "Reducibility Among Combinatorial Problems.", *Complexity of Computer Computations*, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y.. New York: Plenum, p.85-103.
- **Osborne, M.R., Presnell, B. and Turlach, B.A**. (2000). "On the LASSO and its dual", *Journal of Computational and Graphical Statistics* **9**(2): 319-337.
- **Tibshirani**, **R.** (1996), "Regression shrinkage and selection via the Lasso", *Journal of Royal. Statistical Society.* B. 58, 267-288