A Multicriteria Data Fitting Model with Sparsity and Entropy

Boreland Bryson

University of Guelph, Canada

bboreland@uoguelph.ca

Herb Kunze

University of Guelph, Canada

hkunze@uoguelph.ca

Davide La Torre

SKEMA Business School, France

davide.latorre@skema.edu

Danilo Liuzzi

University of Milan, Italy

danilo.liuzzi@unimi.it

A Generalized Multicriteria Data Fitting Model with Sparsity and Entropy

Abstract

We present a general data-fitting model which involves different and conflicting criteria. Our model integrates an abstract data fitting term with the entropy and the sparsity of the set of unknown parameters. This model can be analyzed by means of Multiple Criteria Decision Making techniques. We then propose four computational examples to validate the model in practical contexts. In the first one, we apply this algorithm to the obtain a forecasting of the US GDP by means of fractal operators. In the second and in the forth one, we use this approach to the problem of handwritten digit recognition via logistic regression and neural network. Finally in the fourth example we employ this methodology to forecast the US GDP by means of a modified neural network-based model.

1 Introduction

Multiple Criteria Decision Making (briefly MCDM) is a branch of Operations Research and Decision Making which considers decision making models involving multiple and, in general, conflicting criteria such as cost, satisfaction, profit, accuracy, quality and many others. Decision making problems with multiple criteria are more complex to be analyzed but, however, they lead to more informed and better decisions. Since the early 1960s, the number of authors who have provided advances in this field has been growingly and a variety of approaches and methods have been developed for their application in an array of disciplines, ranging from economics to engineering, from finance to management, and many others. In this paper we propose a general data-fitting model which involves three different criteria, namely the data fitting term, the entropy, and the sparsity. The model can be analyzed using different MCDM techniques: however, in this paper, for our computational studies we focus on the scalarization approach which allows to reduce the model complexity by taking into account a weighted combination of the different criteria. The paper is organized as follows: Section 2 recalls the basic formulation of an MCDM model. Sections 3.1 3.2 3.3 present the three criteria involved in our model formulation, while 3 is devoded to the MCDM model presentation. shows several numerical applications to different areas including fractal image compression using IFSM, handwritten digit recognition via logistic regression and neural network.

2 Basics on Multiple Criteria Decision Making

The aim of this section is to recall some basic facts in Multiple Criteria Decision Making (MCDM). Given a vector-valued map $J : X \subset \mathbb{R}^n \to \mathbb{R}^p$, any finite-dimensional MCDM problem can be written:

$$\max_{x \in X} J(x). \tag{1}$$

As usual, here we suppose that $\mathbb{R}^p$ is ordered by the Pareto cone $\mathbb{R}^p_+$. A point $x \in X$ is said to be Pareto optimal or efficient if $J(x)$ is one of the maximal elements of the set of achievable values in $J(X)$. Thus a point x is Pareto optimal if it is feasible and, for any possible $x' \in X$, $J(x) \leq_{\mathbb{R}^p_+} J(x')$ implies $x = x'$. In a more synthetic way, a point $x \in X$ is said to be Pareto optimal if $(J(x) + \mathbb{R}^p_+) \cap J(X) = \{J(x)\}$.

Among the different techniques to reduce an MCDM problem to a single criterion model there is, for sure, the scalarization technique. Using a scalarization technique, a multiple objective model can be reduced to a single criterion problem by summing up all criteria with different weights. The weights in front of each criterion express the relative importance of that criterion for the Decision Maker. By using this approach, more precisely, by scalarization, an MOP model boils down to:

$$\max_{x \in X} \sum_{i=1}^{p} \eta_i J_i(x), \tag{2}$$

where β is a vector taking values in the interior of $\mathbb{R}^p_+$, namely $\beta \in \text{int}(\mathbb{R}^p_+)$. The equivalence between the scalarized problem and the original MOP problem is complete if the J_i are linear and, by varying η_i, it is possible to obtain different

Pareto optimal points. In the other cases linear scalarization provides only partial results. Other scalarization methods can be found in the literature that can also be used for non-convex problems. Scalarization can also be applied to problems in which the ordering cone is different than the Pareto one. In this case, one has to rely on the elements of the dual cone to scalarize the multicriteria problem.

3 The MCDM Model

The model we are interested in involves the following criteria:

- the Data Fitting Error DFE which describes the accuracy of the approximation;
- the Entropy ENT which models the amount of information carried by the parameters' model;
- the Sparsity SP which describes the complexity of the solution in terms of number of elements in the basis to be utilized to approximate the target.

It is worth noticing that these three criteria are, in general, conflicting. It is clear that a reduction of the sparsity criterion SP, i.e. a reduction of the number of non-zero parameters involved in the model, will negatively affect the DFE as fewer elements in the basis are available to construct the solution. To observe that the Entropy ENT and the Sparsity SP criteria are also conflicting, let us take a simple example where X is a random variable with only two possible outcomes x_1 and x_2 with probabilities p and $1 - p$, respectively. It is clear that if p increases, and then $1 - p$ decreases, x_1 gets more and more likely to happen. This would

produce a decrement in $ENT(X)$ while the sparsity of the vector (x_1, x_2) would increase (see also [15] for a nice discussion on the importance of the concepts of entropy and sparsity). The following subsections describe the three criteria with more details.

3.1 The Data Fitting Term

Given two normed spaces X and Y and a compact set of parameters $\Lambda \subset \mathbb{R}^p$, and a set of input vectors x_i and labels y_i, $i = 1...N$, consider a black box function $f : X \times \Lambda \to Y$ consider the following data fitting/minimization problem:

$$\min_{\lambda \in \Lambda} DFE(\lambda) := \sum_{i=1}^{n} d(f(x_i, \lambda), y_i) \tag{3}$$

As we can see from this formulation the problem is reduced to the minimization of the function $DFE(\lambda)$ over the parameters' space Λ. Depending on the specific function form of f, the function DFE can show different mathematical properties.

3.2 The Entropy Term

The concept of entropy, as it is now used in information theory, was developed by C.E. Shannon [17]. Over the years it has been used in different areas and applications in various scientific disciplines. In his article, Shannon introduces the concept of information of a discrete random variable with no memory as a functional that quantifies the uncertainty of a random variable. The concept of entropy describes the level of information associated with an event. More precisely, the definition of Shannon's entropy [17, 7] satisfies the following properties:

- The measure is continuous and by changing the value of one of the probabilities by a very small amount should only produce a small change of the entropy;
- If all the outcomes are equally likely, then entropy should be maximal.
- If a certain outcome is a certainty, then the entropy should be zero.
- The amount of entropy should be the same independently of how the process is regarded as being divided into parts.

According to these desiderata, Shannon defines the entropy in terms of a discrete random variable X, with possible outcomes $x_1, ..., x_n$ as:

$$ENT(X) = -\sum_{i=1}^{n} p(x_i) \ln(p(x_i)) \tag{4}$$

For our purposes, this definition needs to be adapted to deal with a set of parameters, that can take both positive and negative values. For a set of parameters $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ the notion of entropy is:

$$ENT(\lambda) = -\sum_{1}^{n} \frac{|\lambda_i|}{\lambda_T} \ln \frac{|\lambda_i|}{\lambda_T} \tag{5}$$

where $\lambda_T = \sum_i |\lambda_i|$. In the sequel, rather than maximizing the entropy term - that represents the total amount of information associated with that particular combination of parameters' values - we will consider the minimization of its opposite, also known as neg-entropy. This criterion will be included in the multiple criteria model illustrated in the following Section 3.

3.3 The Sparsity Term

In literature the notion of sparsity has been widely used to reduce the complexity of a model by taking into in consideration only those parameters whose values have major impact on the solution. In other words, by adding this term we wish to determine solutions that are "simple", or more precisely *sparse*. We say that a real vector x in R^n is sparse, when most of the entries of x vanish. We also say that a vector x is s-sparse if it has at most s nonzero entries. This is equivalent to say that the ℓ_0 pseudonorm, or counting 'norm', defined as

$$\|\lambda\|_0 = \#\{i : \lambda_i \neq 0\} \tag{6}$$

is at most s. The ℓ_0-pseudonorm is a strict sparsity measure, and most optimization problems based on it are combinatorial in nature, and hence in general NP-hard. To overcome these difficulties, it is common to replace the function with relaxed variants or smooth approximations that measure and induce sparsity. One possible variant is to use the ℓ_1 norm instead, which is a convex surrogate for the ℓ_0, defined as

$$\|\lambda\|_1 = \sum_{i=1}^{n} |\lambda_i|. \tag{7}$$

It is also the best surrogate in the sense that the 1 ball is the smallest convex body containing all 1-sparse objects of the form $\pm e_i$ (see [5]). Another possibility is to replace the ℓ_0 pseudonorm with some approximation, as for instance

$$\|\lambda\|_* = \sum_{i=1}^{n} \max\{1 - e^{-\alpha\lambda_i}, 1 - e^{\alpha\lambda_i}\}, \tag{8}$$

$$\|\lambda\|_{**} = \sum_{i=1}^{n} [\max\{1 - e^{-\alpha\lambda_i}, 1 - e^{\alpha\lambda_i}\}]^2, \tag{9}$$

or

$$\|\lambda\|_{***} = \sum_{i=1}^{n} \left(1 - e^{-\alpha\lambda_i^2}\right) \tag{10}$$

for a given $\alpha > 0$. It is worth noticing that $\|\lambda\|_{**}$ is a $C^{1,1}$ or LC^1 function (continuous with Lipschitz gradient).

4 Model Implementation and Numerical Experiments

More in details, the MCDM model we are considering includes the following criteria to be optimized simultaneously:

- $DFE(\lambda)$, the Data Fitting Term to be minimized over $\lambda \in \Lambda$;
- $ENT(\lambda)$ is the Entropy, to be maximized over $\lambda \in \Lambda$;
- $SP(\lambda)$ is the Sparsity, to be minimized over $\lambda \in \Lambda$.

By introduction the neg-entropy $-ENT$, the multiple criteria model can be formulated as a minimization program (now all criteria have to be minimized) as follows:

$$\min_{\lambda \in \Lambda} (DFE(\lambda), -ENT(\lambda), SP(\lambda)). \tag{11}$$

As discussed in the previous section, this MCDM model can be transformed into a single criterion one by means, for instance, of scalarization techniques. More practically, one can construct the following single-criterion model: We scalarize

the model by introducing three different positive weights, namely η_1, η_2, η_3. The scalarized model boils down to:

$$\min_{\lambda \in \Lambda} \eta_1 DFE(\lambda) - \eta_2 ENT(\lambda) + \eta_3 SP(\lambda). \tag{12}$$

By varying the parameters' combinations, one can determine different Pareto optimal solutions. In the following sections we discuss four relevant applications of this model to different contexts.

4.1 Fractal Image Compression with IFSM

In fractal image coding based on Generalized Fractal Transforms (GFT), one seeks to approximate a target image or signal by the fixed point of a contractive fractal transform operator ([1],[2],[3],[8],[11]).

The usual formulation involves a fixed set of geometric contraction maps along with a corresponding set of greyscale maps. The inverse problem, which involves the determination of the best greyscale map parameters for a given target image, is based on the so-called "Collage Theorem," a simple consequence of Banach's fixed point theorem. Another consequence of Banach's fixed point result is that the approximation of the target image or signal can be generated by iteration of the fractal transform.

In [9] and [10], the authors showed that one can find an iterated function system with greyscale maps (IFSM) to approximate any target signal or image with arbitrary precision, and they provided a suboptimal but systematic approach for doing so.

In this numerical example we focus on the method of iterated function systems

with greyscale maps (IFSM) ([9]) which can be used to approximate a given element u of $L^2([0,1])$. We extend the approach developed in [9] and [14] by adding the entropy and the sparsity criteria to the collage error minimization.

We consider the case in which $u : [0,1] \to [0,1]$ and the space

$$X = \left\{ u : [0,1] \to [0,1], u \in L^2[0,1] \right\}. \tag{13}$$

The recall that the main ingredients of an N-map IFSM on X are

1. a set of N contractive mappings $w = \{w_1, w_2, \ldots, w_N\}$, $w_i(x) : [0,1] \to [0,1]$, most often affine in form:

$$w_i(x) = s_i x + a_i, \quad 0 \leq s_i < 1, \quad i = 1, 2, \ldots, N; \tag{14}$$

2. a set of associated functions—the greyscale maps—$\phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$, $\phi_i : R \to R$. Affine maps are usually employed:

$$\phi_i(t) = \alpha_i t + \beta_i, \tag{15}$$

with the conditions

$$\alpha_i, \beta_i \in [0,1] \tag{16}$$

and

$$0 \leq \sum_{i=1}^{N} \alpha_i + \beta_i < 1. \tag{17}$$

Associated with the N-map IFSM (w, ϕ) is the *fractal transform* operator T, the

action of which on a function $u \in X$ is given by

$$(Tu)(x) = \sum_{i=1}^{N}{}' \phi_i(u(w_i^{-1}(x))), \tag{18}$$

where the prime means that the sum operates on all those terms for which w_i^{-1} is defined. It is easy to prove ([9]) that $T : X \to X$ and for any $u, v \in X$ we have

$$d_2(Tu, Tv) \leq C d_2(u, v) \tag{19}$$

where

$$C = \sum_{i=1}^{N} s_i^{\frac{1}{2}} \alpha_i. \tag{20}$$

When $C < 1$, then T is contractive on X, implying the existence of a unique fixed point $\bar{u} \in X$ such that $\bar{u} = T\bar{u}$.

The inverse problem associated with IFSM can, in principle, be solved to arbitrary accuracy, using a procedure defined in Forte and Vrscay (1995). The squared collage distance function associated with an N-map IFSM may be written as a quadratic form,

$$\Delta_N^2(z) = z^T A z + b^T z + c, \tag{21}$$

where $z = (\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N)$. The maps w_k are chosen from an infinite set W of fixed affine contraction maps on $[0, 1]$ which satisfy the following assumptions. We say that a set of maps $w_1, w_2, ..., w_N$ generates a dense and nonoverlapping family A of subsets of I (with respect to the Lebesgue measure m) if for every $\epsilon > 0$ andevery $B \subset I$ there exists a finite set of integers i_k, $i_k \geq 1$, $1 \leq k \leq N$, such

that

1. $A = \cup_{k=1}^{N} w_{i_k}(I) \subset B$,

2. $m(B \backslash A) < \epsilon$, and

3. $m(w_{i_k}(I) \cap w_{i_l}(I)) = 0$ if $k \neq l$,

where m denotes Lebesgue measure. If we define by W^N the set

$$W^N = \{w_1, \ldots w_N\} \tag{22}$$

be the N truncations of w. Let $\Phi^N = \{\phi_1, \ldots, \phi_N\}$ be the N-vector of affine grey level maps. Let Ω be a compact subset of set R^{2N} which describes the set of all possible constraints and let z_N be the solution of the previous quadratic optimization problem over Ω. Let $\Delta^2_{N,min} = \Delta^2_N(z_N)$. In [9] it was proved that $\Delta^2_{N,min} \to 0$ when $N \to \infty$. Using the Collage Theorem, the inverse problem may be solved to arbitrary accuracy. A practical choice for the contraction maps w on $X = [0, 1]$ is

$$w_{ij}(x) = 2^{-i}(x + j - 1),\ i = 1, 2, \ldots, M,\ j = 1, 2, \ldots, 2^i,$$

where

$$N = \sum_{i=1}^{M} 2^i.$$

The following examples illustrate that adding small-weighted entropy and sparsity constraints can lead to a better fixed point approximation.

For $j = 1, \ldots, N$, we introduce the family of IFS maps

$$\begin{cases} w_i^j(x) &= \frac{1}{2^i}x + (j-1)\frac{1}{2^i} \\ \phi_i^j(t) &= \alpha_i^j t + \beta_i^j \end{cases}, \; i = 1, \ldots, 2^j,$$

which for each j defines an associated IFSM map

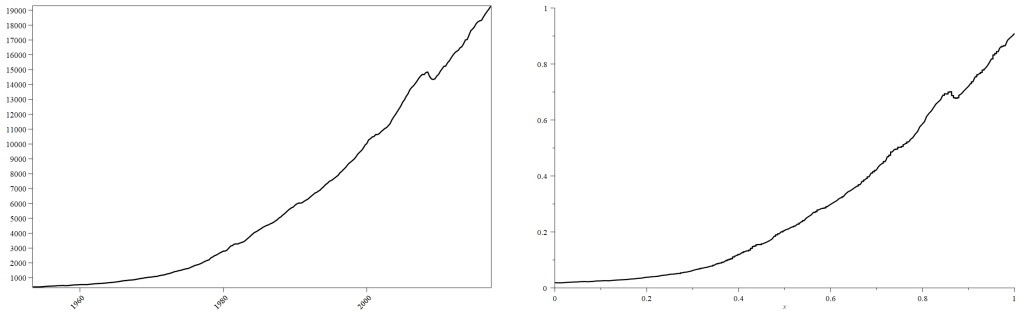
$$(T_j u)(x) = \sum_{i=1}^{2^j}{}' \phi_i^j(u((w_i^j)^{-1}(x))).$$

The map T_j assembles 2^j shrunken and adjusted copies of $u(x)$, each supported on an interval of width $\frac{1}{2^j}$, of the function u. For fixed j, the domains of the maps w_i^j only overlap at the endpoints of their domains. On the other hand, any point $x \in [0, 1]$ that is not a multiple of $\frac{1}{2^m}$ for some m appears in the domain of exactly N of the maps in the family, once per member of the family, offering a sort of map redundancy. We define the combined (contractive) map

$$(Tu)(x) = \sum_{j=1}^{N}(T_j u)(x),$$

and consider the associated squared collage distance $DFE(z)$, where z is the vector of parameters α_i^j and β_i^j. In this example, we explore the scalarized optimization problem (12).

We use the quarterly GDP data for the United States, freely available online at `https://datahub.io/core/gdp-usdata`. The dataset runs from July 1, 1947, through to April 4, 2017, for a total of 282 data values. Following the construction presented in the previous discussion, it is convenient to work with 256 data values, so we work with most recent 256 data values, the interpolation of which is pre-



(a) (b)

Figure 1: (a) Quarterly GDP data for the United States, (b) rescaled to the function u on $[0, 1]^2$.

sented graphically as the function $u(t)$ in Figure 1(a). The theoretical formulation requires that $u : [0, 1] \to [0, 1]$, so we map the 256 quarterly measurement dates to the values $t = \frac{i}{256}$, $i = 0, \ldots, 255$, and, letting we scale the GDP range values of u, dividing by 1.1 times maximal GDP value in the data set. The result is the rescaled graph in Figure 1(b). We set $M = 5$, which means we have a total of 124 parameters in the optimization problem. We use the nonlinear program solver in Maplesoft's Maple to solve (12).

Table 1 shows the results for the example. The table reports the values of η_2 and η_3, the entropy and sparsity weights, respectively; the values of the collage distance ($DFE(z)$); the value of the entropy ($ENT(z)$); the sparsity outcome, given as the number of nonzero parameter values ($SP(z)$); and, finally, $\|v - \bar{u}\|_2$, the L^2 distance between the target $u(t)$ and the resulting fixed point approximation $\bar{u} = T\bar{u}$. Although values in the table are presented with various numbers of decimal places, the computations were run with 100 digits of floating point precision.

The first row of Table 1 gives the results without any entropy or sparsity constraints: the resulting solution uses 74 parameters. In the next six rows, the en-

tropy constraint enters the objective function. The middle four of these six rows show for which the L^2 error in the fixed point approximation improves thanks to the addition of this tiny bit of entropy. Many values of $\eta_2 \in [6.58 \times 10^{-6}, 7.21 \times 10^{-6}]$ induce a fixed point with such an improvement, but there also values of η_2 in this interval for which the fixed point approximation error worsens compared to the case $\eta_2 = 0$. Notice, as well, that the value of $ENT(z)$ increases when $\eta_2 \neq 0$, as we would expect, and, interestingly, the number of nonzero parameters also increases. The next seven rows in the Table explore the effect of introducing

η_2 ($\times 10^{-6}$)	η_3 ($\times 10^{-7}$)	$DFE(z)$	$ENT(z)$	$SP(z)$	$\|v - \bar{u}\|_2$
0	0	0.1594678	17.26794	74	0.00010417211319
6.57	0	0.1595238	27.41607	83	0.00010417305383
6.58	0	0.1595269	27.42275	83	0.00010416094993
6.69	0	0.1595286	27.42698	84	0.00010416223506
7.00	0	0.1595336	27.43881	84	0.00010416601317
7.21	0	0.1595371	27.44676	84	0.00010416870439
7.22	0	0.1595358	27.44398	84	0.00010419814398
0	0.11	0.1594678	13.50961	53	0.00010417205581
0	0.22	0.1594678	13.46549	53	0.00010417218565
0	1.54	0.1594682	13.07825	51	0.00010416990711
0	1.98	0.1594685	13.26600	50	0.00010416168674
0	2.42	0.1594687	13.13613	50	0.00010416639150
0	3.74	0.1594701	12.68110	49	0.00010416826871
0	22.80	0.1595283	9.59571	35	0.00010418056198
7.00	1.54	0.1595240	27.41526	83	0.00010415433247
6.69	1.98	0.1595171	27.39696	83	0.00010414801427

Table 1: Results of the ISFM Example, using United States GDP data.

sparsity into the objective function, with no entropy. In all seven rows, we see a meaningful decrease in the number of nonzero parameters compared to the case when the sparsity weight, η_3, is zero. When η_3 increases, we see that the number of nonzero parameters decreases, and, despite this decrease, we see a number of

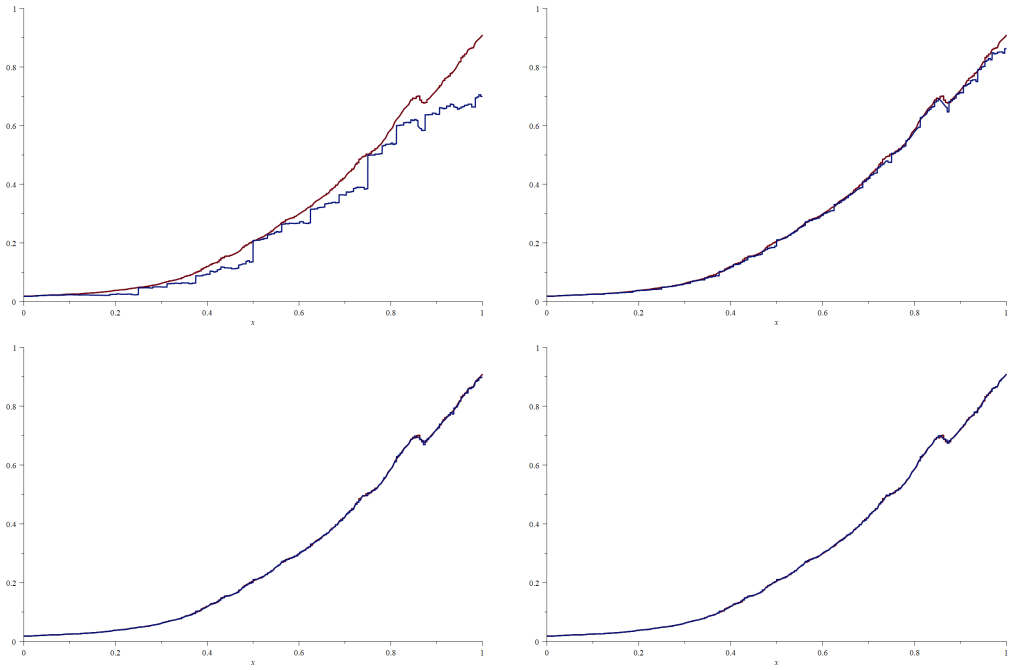


Figure 2: (top-to-bottom,left-to-right) Iterations 2, 4, 6 and 10 of the IFSM operator corresponding to the final row of Table 1.

η_3 values that give an improvement in the fixed point approximation, compared to row one. For illustration, the final of these seven rows, shows that we can drive the number of nonzero parameters to below half the corresponding number in Table 1. The final two rows of the Table show that combining a small amount of entropy and sparsity in the objective function can generate parameter values that induces an additional decrease in the fixed point approximation error.

Figure 2 displays some iterates of the IFSM operator T corresponding to the final row of the table.

4.2 Handwritten Digit Recognition via Logistic Regression

In this example we apply the theoretical framework described above to a handwritten digit recognition system. We use a logistic regression model for a one-vs-all classification problem. The prediction will be the label that has the largest output among all the possible one-vs-all classifications, that is the largest value of the hypothesis function. We build on a widely known and freely available example of logistic regression discussed in Andrew Ng course on Machine Learning at Coursera. We use a subset of the MNIST database of handwritten digits. [1] Some examples of the digits stored in the database are represented in Figure 3: each of the handwritten digits consists of a 20x20 pixels grayscale picture.

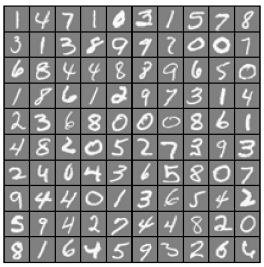


Figure 3: Handwritten digits from the MNIST database

In the remainder of this section we implement a data fitting model based on a scalarization of the problem 11, following the approach of **Model 1**. For the accuracy of the approximation we use a standard logistic regression cost function:

$$\begin{aligned} DFE(\lambda) &= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\, log(h_\lambda(x^{(i)})) - \\ &\quad (1-y^{(i)})\, log(1-h_\lambda(x^{(i)}))] \end{aligned} \tag{23}$$

[1] Available at http://yann.lecun.com/exdb/mnist/

where m is the number of training examples, $x^{(i)}$ corresponds to the i^{th} example's vector of features and $y^{(i)}$ to the associated label (with $y = 0..9$). The function $h_\lambda(x^{(i)}$ embodies the usual hypothesis of the logistic regression:

$$h_\lambda(x^{(i)}) = \frac{1}{1 + e^{\lambda^T x^{(i)}}} \tag{24}$$

In our case study we have $m = 5000$ training examples, while the dimension of the feature vector $x^{(i)}$ is $n = 400$ (20x20 pixels), to which the bias term x_0^i is added. We use the definition of the entropy in 5:

$$ENT(\lambda) = -\sum_{j=1}^{n} \frac{|\lambda_j|}{\lambda_T} \ln \frac{|\lambda_j|}{\lambda_T} \tag{25}$$

where

$$\lambda_T = \sum_{j=1}^{n} |\lambda_j| \tag{26}$$

It is important to notice that we have excluded the coefficient of the bias term λ_0 from the calculation of the entropy, mimicking the procedure applied to regularization terms. In defining the sparsity contribution to the objective function 11, we have used the last option of section 3.3, desribed in equation10:

$$SP(\lambda) = \sum_{i=1}^{n} \left(1 - e^{-\alpha\lambda_i^2}\right) \tag{27}$$

The sparsity measure does not take into account λ_0, the contribution associated to the bias term x_0^i. The training phase is made of ten one-vs-all classifications.

The results are shown in Table 2 for different values of the relative weights η_2 and η_3 . The Accuracy refers to the percentage of correct predictions in the training

η_2	η_3	$ENT(\lambda)$	$SP(\lambda)/n$	Accuracy
0	0	78.449521	0.487567	94.78
1×10^{-6}	0	79.148402	0.502829	94.92
5×10^{-3}	0	81.077755	0.504458	94.94
1×10^{-2}	0	82.271153	0.515114	95.10
1×10^{-1}	0	97.071693	0.595306	94.98
0	5×10^{-3}	77.499648	0.469070	94.62
0	5×10^{-2}	71.823349	0.458263	94.80
0	1×10^{-1}	66.323675	0.396736	94.50
1×10^{-2}	1×10^{-6}	83.078820	0.524375	95.12
1×10^{-2}	1.001×10^{-6}	82.879650	0.536819	95.20

Table 2: $ENT(\lambda)$, $SP(\lambda)/n$ and Accuracy on the training set.

set, while $SP(\lambda)/n$ refers to the average of the sparsity measure 10 taken over the ten one-vs-all classifications and divided by n, so to be comparable across different machine learning algorithms. The parameter α in 10 has been set to $\alpha = 10$. An inspection of Table 2 corroborates the conclusions of the previous section. With respect to the benchmark scenario ($\eta_2 = 0$ and $\eta_3 = 0$), a carefully chosen entropy contribution ameliorates the accuracy, as shown in rows 2 to 5. The same applies after the introduction of the sparsity term: it is possible to slightly increase the accuracy while reducing $SP(\lambda)/n$, as shown in rows 6 to 8. The last two rows shows that an additional gain in accuracy is attainable with a combination of entropy and sparsity corrections.

4.3 Handwritten digits recognition with Neural Network

In this section we propose an example of handwritten digit recognition via a neural network. We use the same database of the previous section and we build on the same widely known Coursera's example. The network's architecture consists of

three layers , as shown in Figure 4: one input layer, one hidden layer and the output layer. The input layers has $n = 400$ units plus the bias term, the hidden layer has $H = 10$ units plus bias, the output layer has $K = 10$ units, corresponding to the 10 digits (labels). We proceed as in **Model 1**. The cost function is:

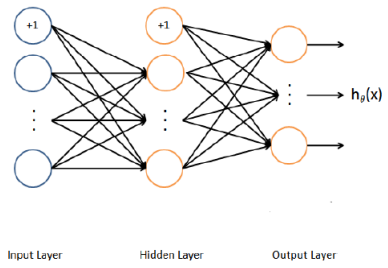


Figure 4:

$$DFE(\lambda) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}[-y_k^{(i)}\, log((h_\lambda(x^{(i)}))_k) - (1-y_k^{(i)})\, log(1-(h_\lambda(x^{(i)}))_k)] \tag{28}$$

where the hypothesis $(h_\lambda(x^{(i)}))_k$ is obtained through forward propagation: each unit or perceptron in the second and third layer processes the linear combination of its incoming signals via a sigmoid function. The index $k = 1..K$ represent the k^{th} label. As for the entropy, we follow the definition given in the previous section, eq. 25, adapted to the network architecture:

$$ENT(\lambda) = -\sum_{j=1}^{H}\sum_{i=1}^{n}\frac{\left|\lambda_{ji}^{(1)}\right|}{\lambda_T}\ln\frac{\left|\lambda_{ji}^{(1)}\right|}{\lambda_T} + \tag{29}$$

$$-\sum_{k=1}^{K}\sum_{j=1}^{H}\frac{\left|\lambda_{kj}^{(2)}\right|}{\lambda_T}\ln\frac{\left|\lambda_{kj}^{(2)}\right|}{\lambda_T} \tag{30}$$

where

$$\lambda_T = \max\left(\lambda_T^{(1)}, \lambda_T^{(2)}\right), \tag{31}$$

with

$$\lambda_T^{(1)} = \max\left|\lambda_{ji}^{(1)}\right| \tag{32}$$

$$\lambda_T^{(2)} = \max\left|\lambda_{kj}^{(2)}\right| \tag{33}$$

The matrices $\lambda^{(1)}$ and $\lambda^{(2)}$ represent the forward propagation from layer 1 to layer 2 and from layer 2 to layer 3 respectively. The bias terms, corresponding to the indeces $i = 0$ and $j = 0$, have been excluded from the calculations.The sparsity measure follows the definition in 10:

$$SP(\lambda) = \sum_{j=1}^{H}\sum_{i=1}^{n}\left(1 - e^{-\alpha\left(\lambda_{ji}^{(1)}\right)^2}\right) + \sum_{k=1}^{K}\sum_{j=1}^{H}\left(1 - e^{-\alpha\left(\lambda_{kj}^{(2)}\right)^2}\right) \tag{34}$$

The results of the training are shown in Table 3 for different combinations of the weights η_2 and η_3.

η_2	η_3	$ENT(\lambda)$	$SP(\lambda)/n$	Accuracy
0	0	461.1917	0.4766	97.88
1×10^{-6}	0	463.0571	0.4832	97.94
0	1×10^{-4}	455.4616	0.4759	97.90
1×10^{-6}	1×10^{-4}	467.2392	0.5091	98.38

Table 3: $ENT(\lambda)$, $SP(\lambda)/n$ and Accuracy on the training set.

The conclusions are consistent with those obtained in the logistic regression

experiment. The entropy contribution improves on the benchmark scenario (row 2 wrt row 1) in terms off accuracy. $SP(\lambda)/n$ increases slightly the accuracy while reducing the number of nonzero elements. A carefully crafted combination of ENT and SP creates an additional increase in accuracy.

4.4 Time Series Forecasting with a Deep Neural Network

The model is a deep neural network consisting of 32 input neurons (1 for bias, 30 for historical data points, 1 for current data point), a hidden layer containing 8 fully connected neurons, and a single output neuron. The network architecture is one variation of Figure 4. The input layer and hidden layer use a rectified linear unit activation function while the output neuron uses the Adam optimizer with a mean squared error cost function, $DFE(w) = L(y(w) - y^*)$, where $y(w)$ is the predicted value, w is the network weights, and y^* is the actual data value, as the base case cost function. The use of a rectified linear unit (ReLU) activation function, defined by $f(x) = \max\{x, 0\}$, helps the network to converge quickly. The Adam optimization algorithm [12] is a modified version of the stochastic gradient descent method that is computationally efficient and has little memory requirement.

As in Section 4.1, we use the quarterly GDP data from the United States, starting from 1947. The network forecasts the GDP based on the previous 30 data points; for example, 2015 Q3 GDP would be predicted using data points from 2008 Q1 through 2015 Q2. The training data is made up of 70% of the complete dataset and the remaining 30% is test data. The model is trained over 100 epochs with batch sizes of 30. For clarity, for each epoch the model trains the network,

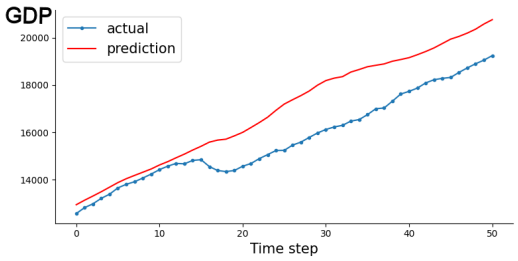


Figure 5: Training results for the base case: Time steps 15-20 correspond to the recession in 2008-09.

adjusting values of the weights using 30 of the data points; the following epoch will contain 30 different data points from which to train; and this procedure occurs 100 times.

When looking at the testing data in Figure 1(a), we see a dip in the value of GDP in the years of 2008 and 2009, during the great recession. We expect that this dip in the data will cause issues for the training of the network and, hence, the predictions it provides. Sure enough, the trained network in the dase case, with cost function $DFE(w)$, produces a prediction which begins to have errors around the years of 2008-09. The predictions instead continuing the trend of a steady increase seen in the training data in prior years. See Figure 5.

As discussed in Section 3.3, the ℓ_1 norm can serve as a proxy sparsity term, $SP(w) = \ell_1(w)$. Using $\eta_3 = 0.0001$, we see in Figure 6(a) that we have improved upon the error found in the base case of Figure 5. Indeed, around 2014 (time step 40), the network's predictions lie very close to the actual values. The results with $\eta_3 = 0.1$, in Figure 6(b), show a slightly poorer match between prediction and actual results at the earlier time steps, but a modest improvement over the base case for later time steps. To give a sense of the speed of convergence of the network, we also present graphs of the loss or cost function versus epoch for these

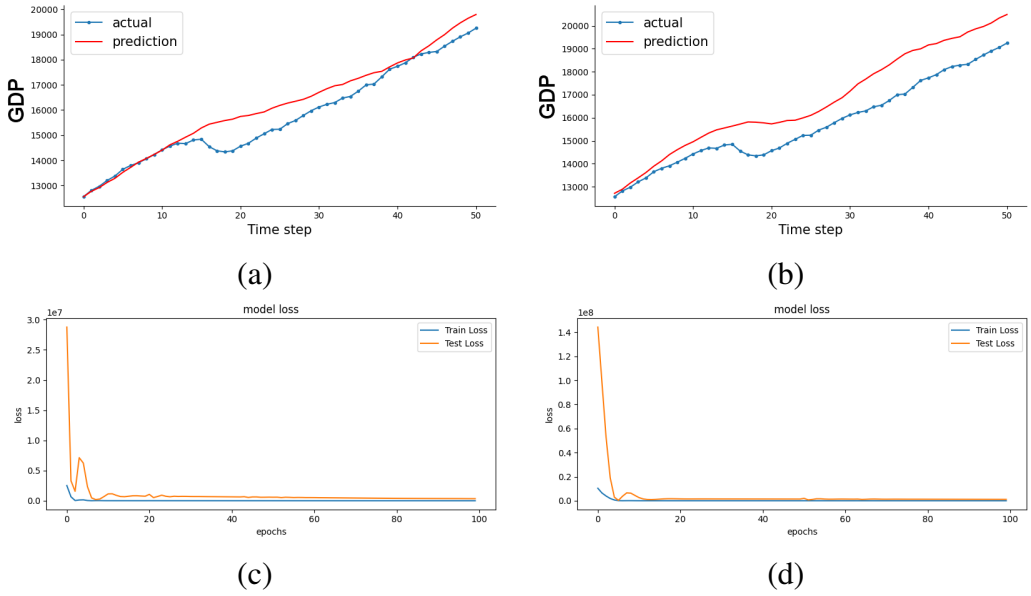


Figure 6: Training results with $\eta_2 = 0$, $SP(w) = \ell_1(w)$, and (a) $\eta_3 = 0.0001$ and (b) $\eta_3 = 0.1$. The corresponding cost function graphs are in (c) and (d).

two cases in Figure 6(c)-(d). No entropy term is introduce as yet ($\eta_2 = 0$).

We note as well that the ℓ_2 norm can serve as a proxy sparsity term, $SP(w) = \ell_2(w)$. For comparison purposes, we consider the same cases we did for the ℓ_1 norm, $\eta_3 = 0.0001$ and $\eta_3 = 0.1$, and present the corresponding graphs in Figure 7. Although the addition of this term improves the predictions compared to the base case, the improvement is less than the corresponding case with $SP(w) = \ell_1(w)$.

We introduce an entropy term $ENT(w)$ of the form in (30). In the earlier exploration, we never saw a network weight grow beyond the value of 10, so we set $\lambda_T = 10$ to hopefully ensure that the terms $|w_i|/\lambda_T < 1$. In the absence of a sparsity constraint ($\eta_3 = 0$), we consider $\eta_2 = 0.001$ to find that the error improves compared to the base case. In particular, around time step 30, there is a notable decrease in the error. The situation improves modestly when we increase

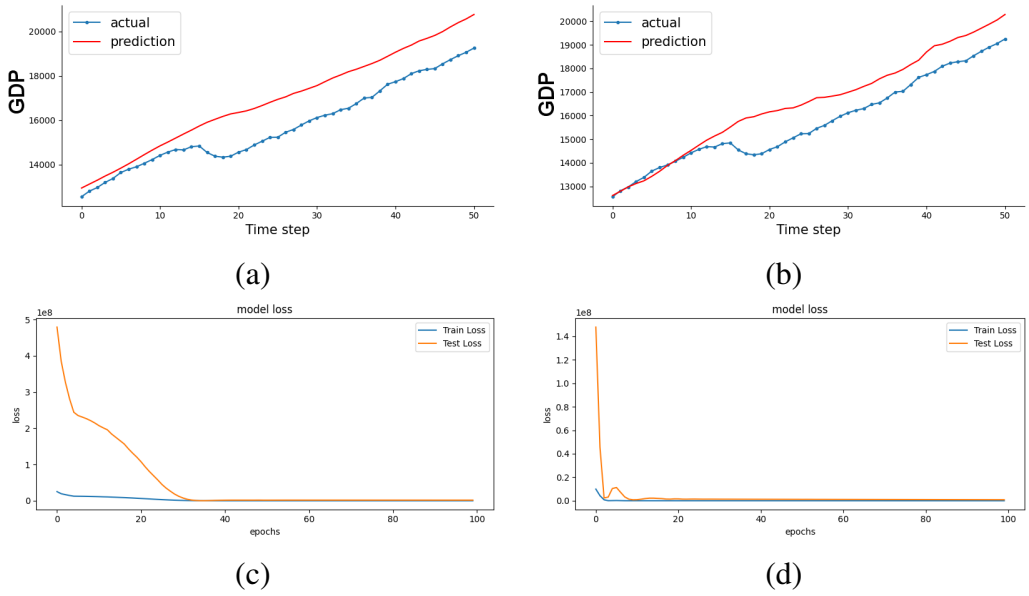
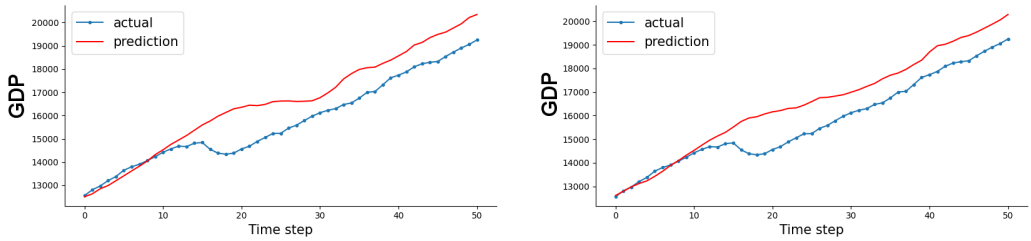


Figure 7: Training results with $\eta_2 = 0$, $SP(w) = \ell_1(w)$, and (a) $\eta_3 = 0.0001$ and (b) $\eta_3 = 0.1$. The corresponding cost function graphs are in (c) and (d).

the coefficient to $\eta_2 = 0.1$. See Figure 8(a)-(b).

Finally, we introduce both entropy and sparsity terms into the formulation. In this case, we use $SP(w) = \ell_1(w) + \ell_2(w)$. Figure 9 presents the predictions for four values of (η_2, η_3). In general, the error obtained by including the two constraints is lower than the corresponding cases of the individual constraints with the same η_i value. The choices $(\eta_2, \eta_3) = (0.001, 0.0001)$ and $(\eta_2, \eta_3) = (0.1, 0.1)$ produce very good results.

Once again, we observe that adding a "small amount" of sparsity or entropy to the base objective function gives a noticeable improvement in the function of the network, and adding appropriate amounts of both types of constraints can lead to a further improvement.



(a) (b)

Figure 8: Training results with $\eta_3 = 0$, $ENT(w)$ term, and (a) $\eta_2 = 0.001$ and (b) $\eta_2 = 0.1$.

References

[1] M.F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster, "Solution of an inverse problem for fractals and other sets", *Proc Nat Acad Sci USA*, Vol. 83, pp. 1975–1977, 1985.

[2] M.F. Barnsley, *Fractals everywhere*, New York: Academic Press, 1989.

[3] M.F. Barnsley and S. Demko S, "Iterated function systems and the global construction of fractals", *Proc Roy Soc London Ser A*, Vol. 399, pp. 243–75, 1985.

[4] M.I. Berenguer, H. Kunze, D. La Torre, and M. Ruiz Galán, "Galerkin method for constrained variational equations and a collage-based approach to related inverse problems", *J. Comput. Appl. Math.*, Vol. 292, pp. 67–75, 2016.

[5] E. J. Candès, "Mathematics of sparsity (and a few other things)", *Proceedings of the International Congress of Mathematicians*, Seoul, South Korea, 2014.

[6] L.C. Evans, *Partial Differential Equations*, Graduate Studies in Mathematics, American Mathematical Society, 2010.

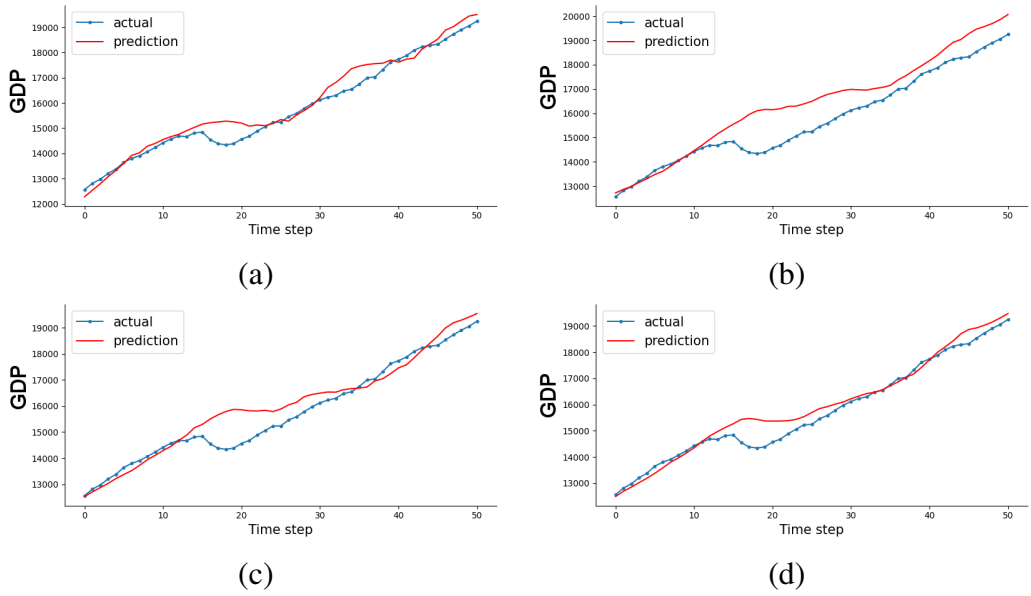


Figure 9: Training results with $SP(w) = \ell_1(w) + \ell_2(w)$ and (a) $(\eta_2, \eta_3) = (0.001, 0.0001)$ (b) $(\eta_2, \eta_3) = (0.1, 0.0001)$, (c) $(\eta_2, \eta_3) = (0.001, 0.1)$, and (d) $(\eta_2, \eta_3) = (0.1, 0.1)$.

[7] F. Flores Camachoa, N. Ulloa Lugob, and H. Covarrubias Martıneza, "The concept of entropy, from its origins to teachers", *Revista Mexicana de Fısica E*, Vol. 61, pp. 69–80, 2015.

[8] Y. Fisher, *Fractal image compression, theory and application*, New York: Springer-Verlag, 1995.

[9] B. Forte and E.R. Vrscay, "Solving the inverse problem for function and image approximation using iterated function systems", *Dynamics of Continuous, Discrete and Impulsive Systems*, Vol. 1(2), 1995.

[10] B. Forte and E.R. Vrscay, "Theory of generalized fractal transforms", *Fractal Image Encoding and Analysis NATO ASI Series* (Y. Fisher Ed.), Vol. 159, 1999.

[11] M. Ghazel, G.H. Freeman, and Vrscay ER, "Fractal image denoising", *IEEE Trans Image Proc*, Vol. 12 (12), pp. 1560–78, 2003.

[12] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, 2015. `arXiv:1412.6980`

[13] H. Kunze, D. La Torre, F. Mendivil, and E.R. Vrscay, *Fractal-based methods in analysis*, Springer, 2012.

[14] D. La Torre and E.R. Vrscay, "Fractal-based measure approximation with entropy maximization and sparsity constraints", *AIP Conf. Proc.*, Vol. 1443, pp. 63–71, 2012.

[15] G. Pastor, I. Mora-Jimenez, R. Jantti, and A.J. Caamano, "Mathematics of Sparsity and Entropy: Axioms, Core Functions and Sparse Recovery", *Proceedings of the Tenth International Symposium in Wireless Communication Systems (ISWCS 2013)*, 2013.

[16] Y. Sawaragi, H. Nakayama, T. Tanino, *Theory of multiobjective optimization*, Academic Press Inc., 1985.

[17] C.E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27(3), pp. 379–423, 1948.

[18] A.N. Tychonoff and N.Y. Arsenin, *Solution of Ill-posed Problems*, Washington: Winston and Sons, 1977.

[19] C.R. Vogel, *Computational Methods for Inverse Problems*, SIAM, New York, 2002.