Homework1

1. Implement Gradient Descent

Algorithm 1 Gradient Descent Procedure

Input:
Obj. function to minimize $f(X) : R^n \rightarrow R$,
initial guess $X^0 = (x_1^0, ..., x_n^0)$,
step size η,
threshold ϵ.
Initialization: $X^t = X^0$
repeat
  Compute $f(X^t)$ and $\nabla f(X^t)$.
  Compute $X^{t+1} = X^t - \eta \nabla f(X^t)$
  $X^t = X^{t+1}$
until $\left| f(X^t) - f(X^{t+1}) \right| < \epsilon$

Algorithm 1 is a gradient descent procedure to minimize scalar functions of a vector argument. The convergence criterion of the procedure is a threshold ϵ such that the algorithm terminates when the difference in the objective function on successive steps, X^t and X^{t+1}, is less than ϵ. Table 1 shows two tests of the Gradient Descent procedure: on the quadratic bowl function (left) and on a non-convex function (right). The graphics of both function are depicted in Figure 1 (a) and (c). Figure 1 (b) and (d) has the level set of z for small values of x and y. Calculating the minimum using the gradient descent from the point $(8, 8)$, with a step $\eta = 0.25$, the procedure converges in 2 iterations to the minimum of the quadratic bowl function. And starting from $(0, 1)$ with a step $\eta = 0.5$, the procedure takes only one step to converge with the non-convex function. However, in this case the procedure converged to the local minimum $(0, 0)$ and not the global minimum (near $(1.408, 0)$) of the function. Indeed for, for most starting points with $x < 0$, the gradient descent will converge to the local minimum instead of the global minimum.

In gradient descent, the initial guess X^0, the step size η and the threshold ϵ play an essential role in the convergence of the algorithm. Table 2 shows the number of iterations of the procedure to converge and the value of the function at the final step. A very small threshold ϵ will make the final value of the function at the minimum more accurate, but the procedure will require more iterations to converge, as the last to column on the right. Equally important, a small step η will also require more iterations to converge. In the example, the number of iterations jumps from 2 to 9 when the step size η is reduced by half. However, a large value of η can prevent the procedure of convergence by making the algorithm 'jump' from one side to the other of the minimum, which is what the first column of the table shows. Finally, picking a bad initial guess will not only impact the performance of the algorithm (column 3 and 4 of Table 2) but it can also lead it to find a local minimum, as seen in the example of Table 1 (rignt).

Table 1. Analytical gradient descent of a convex function (left) and a non-convex function (right).

$f(X) =$	$x_1^2 + x_2^2$		$3x^4 - 8x^3 + 5x^2 + y^2$	
η	0.25		0.5	
ϵ	0.1		0.1	
$\nabla f(X) =$	$\begin{bmatrix} 2x \\ 2y \end{bmatrix}$		$\begin{bmatrix} 12x^3 - 24x^2 + 10x \\ 2y \end{bmatrix}$	
Iterations	X	$F(X)$	X	F(X)
0	$(8, 8)$	128	$(0, 1)$	1
1	$(4, 4)$	32	$(0, 0)$	0
2	$(0, 0)$	0	-	-

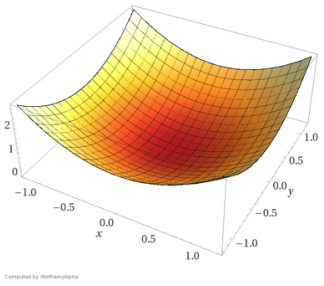
Table 2. Number of iterations for convergence and final value of the objective function for different values of the step, threshold and initial guess.

OBJECTIVE FUNCTION : $f(X) = x_1^2 + x_2^2$					
(x_1^0, x_2^0)	(4,8)	(4,8)	(4,8)	(5,11)	(5,11)
η	2	0.5	0.25	0.25	0.25
ϵ	0.001	0.001	0.001	0.001	10^{-5}
Iterations	32	2	9	10	13
$min(f(X))$	*No conv.*	0.0	3.10^{-4}	10^{-4}	2.10^{-6}

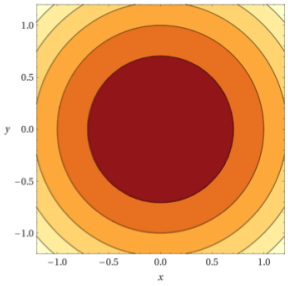
The gradient of a function at a given point can be numerically approximated using the *Finite Difference* method. The central differences approximate the n^{th} element of the gradient of a function $f(X) : R^n \rightarrow R$ by:

$$\nabla_h f_n(X) = \frac{f(X + 1/2 * h * \mathbb{1}_n) - f(X + 1/2 * h * \mathbb{1}_n)}{h} \tag{1}$$

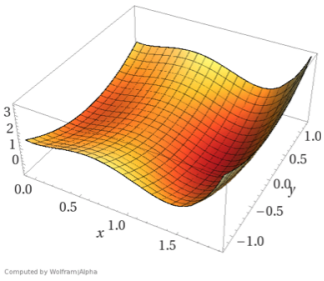
Our gradient descent code is written in python and takes as parameters the initial guess, x_0, the step size, γ, and the convergence threshold, ϵ. The convergence criteria that we



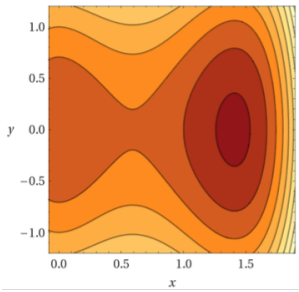
(a) Quadratic bowl in 3D



(b) Quadratic bowl in 2D



(c) Non-convex function in 3D



(d) Non-convex function in 2D

Figure 1. (a) and (b) are 3D and 2D graphics of a convex function with only one local and overall minumum. (c) and (d) are 3D and 2D graphics of a non-convex function with at least 2 local minima.

use to terminate is when the difference between successive "guesses" (call these x_n and x_{n+1} for some arbitrary n) is less than ϵ, i.e. $|x_{n+1} - x_n| < \epsilon$. Since we are only working with scalar functions, this will suffice. If we were working with vector values functions, we could instead use the two-norm.

In order to test our gradient descent, we experimented with a few different functions. We used a quadratic function: $f(X) = \sum_{i=1}^{n} x_i^2$, the Rosenbrock function: $f(x_1, x_2) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$ (as referenced in the Wikipedia page for gradient descent), and a few non convex functions to test gradient descent with multiple minima.

The choice of starting guess, x_0, the step size, γ and the convergence threshold are very import to yield the "correct" solution. In order to be confident that the algorithm has reached a solution, it's important to pick a small ϵ. For most of our testing, we used $\epsilon = 10^{-6}$. However, picking too small of an epsilon can be computationally expensive: causing the algorithm to perform many more iterations that necessary to produce a "confident" solution. The choice of starting guess, x_0 is also interesting. In order to guarantee convergence, one needs to pick an x_0 relatively close to the true solution. (In the case of more than one minima, the algorithm will converge to the minima closest to the starting point). We found that However, we found the most important criteria was the γ parameter. We usually started with $\gamma = 1$ and decreased it's value until we found a good fit. If γ is to large, the algorithm could miss the minima by jumping back and forth between the same points. If γ is too small, the algorithm may not converge as quickly...Mention numerical errors.

Table 3. Finite difference method numerical gradient v. analytical gradient for various point of the quadratic bowl function and a non-convex function for $h = 0.5$.

$f(X) =$	$x_1^2 + x_2^2$		$3x^4 - 8x^3 + 5x^2 + y^2$	
(x_1, x_2)	DIFF	ANALYTICAL	FIN. DIFF	ANALYTICAL
$(4, 8)$	$(8, 16)$	$(8, 16)$	$(426.5, 16)$	$(424, 16)$
$(1, 1)$	$(2, 2)$	$(2, 2)$	$(-1.75, 2)$	$(-2, 2)$
$(1 - \frac{1}{\sqrt{6}}, 0)$	$(1.1835, 0)$	$(1.1835, 0)$	$(-0.0562, 0)$	$(0, 0)$
$(0, 0)$	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(-0.5, 0)$

We also wrote code to approximate the gradient of a function numerically using central differences. We verified its behavior by comparing the analytical and numerical gradients in a testing suite. We used the python unittest framework and the assertAlmostEquals function to guarantee that all approximations were within 7 digits of accuracy. We tested the gradient on the same functions we used to test our gradient descent, and also a few common polynomial functions.

2. Linear Basis Function Regression

(2.4) In this instance, we used a polynomial basis function. If instead the basis functions were sine functions, $\phi_1(x) = \sin(2\pi x), ..., \phi_M(x) = \sin(2\pi M x)$ we would expect that these basis functions would insure an even better fit for the data. In fact, testing this on our data set, we found that the SEE was much smaller (insert example). However, the sine basis functions are not always a good fit. If we did not know that the data was generated from $\sin(2\pi x)$, using an oscillating basis function like the sine can cause overfitting (like we saw in the polynomial basis case of $M = 9$ on the curve fitting data set). Also, as alluded to, using a sine basis would force our data to oscillate, which may not be a good choice if our data doesn't have that shape. Examples include x^3, which has a saddle point and does not oscillate, or something with an exponential growth (like e^x)

3. Ridge Regression

4. Generalizations

4.1. Outlier and LAD

4.2. Sparsity and LASSO

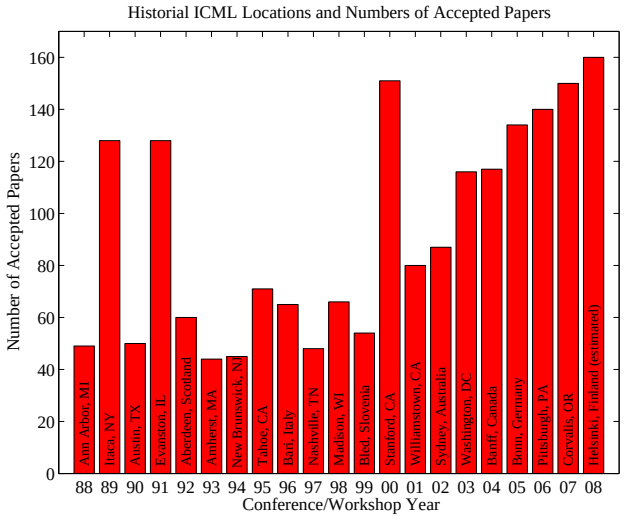


Figure 2. Historical locations and number of accepted papers for International Machine Learning Conferences (ICML 1993 – ICML 2008) and International Workshops on Machine Learning (ML 1988 – ML 1992). At the time this figure was produced, the number of accepted papers for ICML 2008 was unknown and instead estimated.

4.3. Figures

You may want to include figures in the paper to help readers visualize your approach and your results. Such artwork should be centered, legible, and separated from the text. Lines should be dark and at least 0.5 points thick for purposes of reproduction, and text should not appear on a gray background.

Label all distinct components of each figure. If the figure takes the form of a graph, then give a name for each axis and include a legend that briefly describes each curve. Do not include a title inside the figure; instead, the caption should serve this function.

Number figures sequentially, placing the figure number and caption *after* the graphics, with at least 0.1 inches of space before the caption and 0.1 inches after it, as in Figure 2. The figure caption should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left. You may float figures to the top or bottom of a column, and you may set wide figures across both columns (use the environment `figure*` in LaTeX), but always place two-column figures at the top or bottom of the page.

Algorithm 2 Bubble Sort

Input: data x_i, size m
repeat
  Initialize $noChange = true.$
  for $i = 1$ **to** $m - 1$ **do**
    if $x_i > x_{i+1}$ **then**
      Swap x_i and x_{i+1}
      $noChange = false$
    end if
  end for
until $noChange$ is $true$

Table 4. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

DATA SET	NAIVE	FLEXIBLE	BETTER?
BREAST	95.9± 0.2	96.7± 0.2	√
CLEVELAND	83.3± 0.6	80.0± 0.6	×
GLASS2	61.9± 1.4	83.8± 0.7	√
CREDIT	74.8± 0.5	78.3± 0.6	
HORSE	73.3± 0.9	69.7± 1.0	×
META	67.1± 0.6	76.5± 0.5	√
PIMA	75.1± 0.6	73.9± 0.5	
VEHICLE	44.9± 0.6	61.5± 0.4	√

4.4. Algorithms

If you are using LaTeX, please use the "algorithm" and "algorithmic" environments to format pseudocode. These require the corresponding stylefiles, algorithm.sty and algorithmic.sty, which are supplied with this package. Algorithm 2 shows an example.

4.5. Tables

You may also want to include tables that summarize material. Like figures, these should be centered, legible, and numbered consecutively. However, place the title *above* the table with at least 0.1 inches of space before the title and the same after it, as in Table 4. The table title should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left.

Tables contain textual material that can be typeset, as contrasted with figures, which contain graphical material that must be drawn. Specify the contents of each row and column in the table's topmost row. Again, you may float tables to a column's top or bottom, and set wide tables across both columns, but place two-column tables at the top or bottom of the page.

4.6. Citations and References

Please use APA reference format regardless of your formatter or word processor. If you rely on the LaTeX bibliographic facility, use `natbib.sty` and `icml2015.bst` included in the style-file package to obtain this format.

Citations within the text should include the authors' last names and year. If the authors' names are included in the sentence, place only the year in parentheses, for example when referencing Arthur Samuel's pioneering work (**?**). Otherwise place the entire reference in parentheses with the authors and year separated by a comma (**?**). List multiple references separated by semicolons (**???**). Use the 'et al.' construct only for citations with three or more authors or after listing all authors to a publication in an earlier reference (**?**).

Authors should cite their own work in the third person in the initial version of their paper submitted for blind review. Please refer to Section **??** for detailed instructions on how to cite your own papers.

Use an unnumbered first-level section heading for the references, and use a hanging indent style, with the first line of the reference flush against the left margin and subsequent lines indented by 10 points. The references at the end of this document give examples for journal articles (**?**), conference publications (**?**), book chapters (**?**), books (**?**), edited volumes (**?**), technical reports (**?**), and dissertations (**?**).

Alphabetize references by the surnames of the first authors, with single author entries preceding multiple author entries. Order references for the same authors by year of publication, with the earliest first. Make sure that each reference includes all relevant information (e.g., page numbers).

4.7. Software and Data

We strongly encourage the publication of software and data with the camera-ready version of the paper whenever appropriate. This can be done by including a URL in the camera-ready copy. However, do not include URLs that reveal your institution or identity in your submission for review. Instead, provide an anonymous URL or upload the material as "Supplementary Material" into the CMT reviewing system. Note that reviewers are not required to look a this material when writing their review.

Acknowledgments

Do not include acknowledgements in the initial version of the paper submitted for blind review.

If a paper is accepted, the final camera-ready version can (and probably should) include acknowledgements. In this case, please place such acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.