What is MACHINE LEARNING?

Prof. Dan A. Simovici

UMB

1. A Formal Model

2. Empirical Risk Minimization (ERM)

3. ERM with Inductive Bias

4. An Example : Regression

What is Machine Learning?

Machine learning (ML) studies the construction and analysis of algorithms that learn from data.

- ML algorithms construct models starting from samples of data and use these models to make predictions or decisions.

- ML and its applied counterpart, data mining, mostly deal with problems that present difficulties in formulating algorithms that can be readily translated into programs, due to their complexity.

- ML techniques tend to avoid the difficulties of standard problem solving techniques where a complete understanding of data is required at the beginning of the problem solving process.

Typical ML Activities

Example

- finding diagnosis for patients starting with a series of their symptoms;
- determining credit worthiness of customers based on their demographics and credit history;
- document classification based on their main topic;
- speech recognition;
- computational biology applications.

Supervised Learning

Often ML aims to compute a label for each analyzed piece of data that depends of the characteristics of data.

The general approach known as supervised learning is to begin with a number of labelled examples (where answers are known or are provided by a supervisor) known as training set.

The goal is to generate an algorithm that computes the function that gives the labels of remaining examples.

Unsupervised Learning

In unsupervised learning the challenge is to identify the structure that is hidden in data, e.g. identifying groups of data such that strong similarity exists between objects that belong to the same group and also, that objects that belong to different groups are sufficiently distinct.

This activity is known as clustering and it is a typical example of unsupervised learning.

The term "unsupervised" refers to the fact that this type of learning does not require operator intervention. Other machine learning activities of this type include outlier identification, density estimation, etc.

Semi-supervised Learning

An intermediate type of activity, referred as semi-supervised learning requires a limited involvement of the operator.

For example, in the case of clustering, this may allow the operator to specify pairs of objects that must belong to the same group and pairs of objects that may not belong to the same group.

Quality of the Learning Process

The quality of the learning process is assessed through its capability for *generalization*, that is, the capacity of the produced algorithm for computing correct labels for yet unseen examples.

- the correct behavior of an algorithm relative to the training data is no guarantee, in general, for its generalization prowess;
- sometimes in the pursuit of a perfect fit of the learning algorithm to the training data leads to *overfitting*; this term describes the situation when the algorithm acts correctly on the training data but is unable to predict unseen data;
- in an extreme case, a *rote learner* will memorize the labels of its training data and nothing else. Such a learner will be perfectly accurate on its training data but lack completely any generalization capability.

Active and Reinforcement Learning

- A machine learning algorithm can achieve greater accuracy with fewer training examples if it is allowed to choose the data from which it learns, that is, to apply active learning.
  An active learner may pose queries soliciting a human operator to label a data instance. Since unlabelled data is abundant and, in many cases, easily obtained there are good reasons to use this learning paradigm.
- Reinforcement learning is a machine-learning paradigm inspired by psychology which emphasizes learning by an agent from its direct interaction with the data in order to attain certain goals of learning e.g. accuracy of label prediction.
  The framework of this type of learning makes use of states and actions of an agent, and their rewards, and deals with uncertainty and nondeterminism.

The Learner's Input

- **The domain set:** $\mathcal{X}$ consists of the objects that we wish to label; usually objects are represented by a vector of features. We refer to these objects as instances.

- **The label set:** $\mathcal{Y}$ is generally a finite set, e.g. $\{0, 1\}$ or $\{-1, 1\}$.
- **Training data:** $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ is a finite sequence of pairs in $\mathcal{X} \times \mathcal{Y}$, that is, a sequence of labelled objects. Each pair (x_i, y_i) is a training example.

The Learner's Output

The learner is required to produce a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ starting from

$$f(x_1) = y_1, f(x_2) = y_2, \ldots, f(x_n) = y_n,$$

as provided by the training data S.

This function is known as a

- a predictor, or
- a hypothesis, or
- a classifier

A Data Generation Model

Assumptions:

- data has a probability distribution function $\mathcal{D}$;
- the learner ignores the probability distribution function $\mathcal{D}$;
- there exists some correct labelling function

$$f : \mathcal{X} \longrightarrow \mathcal{Y}$$

which we seek to determine knowing that $f(x_i) = y_i$ for $1 \leqslant i \leqslant n$.

Measures of Success

Definition

The true error of a prediction rule $h : \mathcal{X} \longrightarrow \mathcal{Y}$ is

$$L_{(\mathcal{D},f)}(h) = \mathcal{D}(\{x \mid h(x) \neq f(x)\}).$$

$L_{(\mathcal{D},f)}(h)$ is a *loss measure*: it evaluates the probability that the prediction rule $h(x)$ will produce a result distinct from the labelling function f.

The error is measured with respect to probability distribution $\mathcal{D}$ and the correct labelling function f.

Alternative terms used for $L_{(\mathcal{D},f)}(h)$:

- generalization error;
- risk;
- the true error of h.

The true error $L_{(\mathcal{D},f)}(h)$ of h is not known to the learner because $\mathcal{D}$ and f are unknown.

The Training Error

The empirical error of a predictor h can be considered only in relation with a labelled sample.

Definition

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a labelled sample. The empirical error of a predictor h on sample S is the number

$$L_S(h) = \frac{|\{i \mid 1 \leqslant i \leqslant m, h(x_i) \neq y_i\}|}{m}$$

An alternative name for $L_S(h)$ is empirical risk.

Empirical Risk Minimization (ERM)

ERM is an approach that seeks a predictor that minimizes the training error $L_S(h)$.

Let h be the predictor defined as $h(x_i) = y_i$ for all $x_i \in S$ and $h(x_i) = k$, where k does not label any object in S. The empirical error will be 0 but h will fail miserably on unseen data. This phenomenon is called overfitting: designing a predictor to fit the sample.

Inductive Bias

ERM can lead to overfitting. Therefore, we seek supplementary conditions that ensure that ERM will not overfit (conditions under which a predictor with good performance on the training data will have good performance on unseen data).

Common solution:

> *Use a restricted hypothesis class $\mathcal{H}$ chosen in advance, that is before seeing the data*

This approach is known as the inductive bias.

- For a given class $\mathcal{H}$ (known as hypothesis class) and a training sample S the hypothesis
$$h = \text{ERM}_{\mathcal{H}}(S)$$
uses the ERM rule to chose a predictor $h \in \mathcal{H}$ with the lowest possible error over S.

- Both large $L_S(h)$ values and strong inductive bias are negative; the question is achieve a balance between these factors.

- Let $\text{argmin}_{h\in\mathcal{H}} L_S(h)$ be the set of hypothesis in $\mathcal{H}$ that achieve the minimum values of $L_S(h)$. This approach aims to have $\text{ERM}_{\mathcal{H}}(S) \in \text{argmin}_{h\in\mathcal{H}} L_S(h)$.

Definition

h_S is the hypothesis that results from applying $\text{ERM}_{\mathcal{H}}$ to the sample S, namely
$$h_S \in \text{argmin}_{h\in\mathcal{H}} L_S(h).$$

Finite Hypothesis Classes

A simple inductive bias: class $\mathcal{H}$ is finite.

Definition

The Realizability Assumption: There exists $h^* \in \mathcal{H}$ such that the true error of a prediction rule $h : \mathcal{X} \longrightarrow \mathcal{Y}$ is

$$L_{(\mathcal{D},f)}(h^*) = 0.$$

This implies the existence of h^* such that with probability 1 over random samples S, where the instances of S are sampled according to $\mathcal{D}$ and are labelled by f we have $L_S(h^*) = 0$.

Realizability assumption implies that for every ERM hypothesis we have $L_S(h_S)$ with probability 1.

- Samples are obtained by drawing values from a distribution $\mathcal{D}$ independently of each other.
- Since samples are drawn randomly from $\mathcal{D}$, the true error $L_{(\mathcal{D},f)}(h_S)$ is a random variable.
- We cannot predict with certainty that a sample S will suffice to direct the learner towards a good classifier.

Example

To predict if a papaya fruit is tasty or not but without ever tasting papayas one needs to decide which features of a papaya the prediction should be based on.

On the basis of your past experience with other fruits, two features will be used: the papaya's *color*, ranging from dark green, through orange and red to dark brown, and the papaya's *softness*, ranging from rock hard to mushy.

The input for figuring out the prediction rule is a sample of papayas that are examined for color and softness and then tasted and found out whether they were tasty or not.

Example

There is always some (small) chance that all the papayas we have happened to taste were not tasty (a non-representative sample), in spite of the fact that, say, 70% of the papayas are tasty. In such a case, $\mathrm{ERM}_{\mathcal{H}}(S)$ may be the constant function that labels every papaya as "not tasty" (and has 70% error on the true distribution of papayas).

We will therefore address the probability to sample a training set for which $L_{(D,f)}(h_S)$ is not too large. Usually, we denote the probability of getting a nonrepresentative sample by δ, and refer to $1 - \delta$ as the confidence parameter of our prediction.

Approximately Correct Predictors

Since we cannot guarantee perfect label prediction, we introduce another parameter for the quality of prediction, the accuracy parameter, denoted by ϵ.

- The probability of getting a non-representative sample is denoted by δ.
- $1 - \delta$ is the confidence parameter.
- The accuracy parameter ϵ: the event involving the true error

$$L_{(\mathcal{D},f)}(h) > \epsilon$$

is a failure of the learner.
If $L_{(\mathcal{D},f)}(h) \leqslant \epsilon$ (the true error is less than ϵ) then the output of the learner is an approximately correct predictor.

Fix f and seek an upper bound for the probability of sampling m instances that will lead to a failure of the learner.

Let $S = (x_1, \ldots, x_m)$. We would like to upper bound the probability that the true error is larger than ϵ, that is, $D^m(\{S \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\})$.

- The set $\mathcal{H}_b$ of bad hypotheses is defined as:

$$\mathcal{H}_b = \{h \in \mathcal{H} \mid L_{(\mathcal{D},f)}(h) > \epsilon\}.$$

  Note that if $h \in \mathcal{H}_b$ we have $1 - L_{(\mathcal{D},f)} < 1 - \epsilon$.

- Define the set of misleading examples:

$$M = \{S \mid \exists h \in \mathcal{H}_b, L_S(h) = 0\}.$$

  M contains those samples that produce an empirical error 0 on bad hypotheses, that is, makes a bad hypothesis look like a good hypothesis.

Note that if a sample S is such that $L_{\mathcal{D},S}(h_S) > \epsilon$, then there exists a bad hypothesis $h \in \mathcal{H}_b$ such that $L_S(h) = 0$.

Goal: to bound the probability of the event $L_{(\mathcal{D},f)}(h_S) > \epsilon$.

- The realizability assumption implies $L_S(h_S) = 0$. Therefore, the event

$$L_{(\mathcal{D},f)}(h_S) > \epsilon$$

can happen only if for some $h \in \mathcal{H}_b$ we have $L_S(h) = 0$, that is, only if $\{S \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subseteq M$.

- The set of misleading examples M can be written as:

$$M = \bigcup_{h \in \mathcal{H}_b} \{S \mid L_S(h) = 0\},$$

hence

$$\mathcal{D}^m(\{S \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \leqslant \mathcal{D}^m(M) = \mathcal{D}^m(\bigcup_{h \in \mathcal{H}_b} \{S \mid L_S(h) = 0\}.$$

Elementary probability implies that

$$P(\bigcup_{i=1}^{k} A_i) \leqslant \sum_{i=1}^{k} P(A_i).$$

Therefore, by elementary probability theory we have

$$\mathcal{D}^m\Big(\bigcup_{h\in\mathcal{H}_b}\{S \mid L_S(h)=0\}\Big) \leqslant \sum_{h\in\mathcal{H}_b}\mathcal{D}^m(\{S \mid L_S(h)=0\}).$$

Fix some bad hypothesis $h \in \mathcal{H}_b$. The event $L_S(h) = 0$ is equivalent to $h(x_i) = f(x_i)$ for $1 \leqslant i \leqslant m$. Since the examples in the training sets are sampled independently and identically distributed (iid), we get

$$\begin{aligned}\mathcal{D}^m(\{S \mid L_S(h)=0\}) &= \mathcal{D}^m(\{S \mid h(x_i)=f(x_i) \text{ for } 1\leqslant i\leqslant m\})\\ &= \prod_{i=1}^{m}\mathcal{D}(\{x_i \mid h(x_i)=f(x_i)\}).\end{aligned}$$

For each individual sampling we have:

$$\mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}) = 1 - L_{(\mathcal{D},f)}(h) \leqslant 1 - \epsilon,$$

where the last inequality follows from the fact that $h \in \mathcal{H}_b$.
Note that $1 - \epsilon \leqslant e^{-\epsilon}$.
Thus,

$$\mathcal{D}^m(\{S \mid L_S(h) = 0\}) \leqslant (1-\epsilon)^m \leqslant e^{-\epsilon m}.$$

Since

$$\mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_b}\{S \mid L_S(h)=0\}\right) \leqslant \sum_{h\in\mathcal{H}_b}\mathcal{D}^m(\{S \mid L_S(h)=0\})$$

we conclude that

$$\mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_b}\{S \mid L_S(h)=0\}\right) \leqslant |\mathcal{H}_b| e^{-\epsilon m}.$$

Theorem

Let $\mathcal{H}$ be a finite hypothesis class, $\delta \in (0,1)$, $\epsilon > 0$ and let m be an integer such that

$$m \geqslant \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$$

Then, for any labelling function f and for any distribution $\mathcal{D}$ for which the realizability distribution holds, with probability at least $1-\delta$ over the choice of an iid sample S of size m, we have that for every ERM hypothesis h_S it holds that $L_{(\mathcal{D},f)}(h) \leqslant \epsilon$.

Thus, for sufficiently large m the ERM rule over a finite hypothesis class $\mathcal{H}$ will be probably (with a confidence of $1-\delta$) approximatively correct (up to an error of ϵ).

Define the class of affine functions L_d that consists of functions of the form $h_{\mathbf{w},b} : \mathbb{R}^d \longrightarrow \mathbb{R}$, where

$$h_{\mathbf{w},b}(\mathbf{x}) = (\mathbf{w}, \mathbf{x}) + b = \sum_{i=1}^{d} w_i x_i + b.$$

Note that:

- Each function $h_{\mathbf{w},b}$ is parametrized by $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.
- Each function $h_{\mathbf{w},b}$ takes as input a vector $\mathbf{x}$ and returns a scalar $(\mathbf{w}, \mathbf{x}) + b$.

Different hypotheses classes are constructed using functions from L_d

THE CLASS OF HALFSPACES:

The class of halfspaces is designed for binary classification problems, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$.

The realizability assumption means that it is possible to separate with a hyperplane all positive example from all negative examples; this is the separable case.

The ERM problem for halfspaces in the realizable case can be expressed as a linear programming problem.

Let $S = \{(\mathbf{x}_i, y_i) \mid 1 \leqslant i \leqslant m\}$ be a training set of size m. Since we assume the realizable case, an ERM predictor should have zero errors on the training set. Thus, we are looking for $\mathbf{w} \in \mathbb{R}^d$ such that

$$\text{sign}((\mathbf{w}, \mathbf{x}_i)) = y_i \text{ for } 1 \leqslant i \leqslant m.$$

Equivalently, we are seeking $\mathbf{w}$ such that

$$y_i(\mathbf{w}, \mathbf{x}_i) > 0 \text{ for } 1 \leqslant i \leqslant m.$$

Since we assume realizability such a vector $\mathbf{w}$ must exist. Let $\mathbf{w}^*$ be one. Define

$$\gamma = \min_i y_i(\mathbf{w}^*, \mathbf{x}_i) \text{ and } \tilde{\mathbf{w}} = \frac{1}{\gamma}\mathbf{w}^*.$$

For all i we have

$$y_i(\tilde{\mathbf{w}}, \mathbf{x}_i) = \frac{1}{\gamma} y_i(\mathbf{w}^*, \mathbf{x}_i) \geqslant 1.$$

Thus, there exists a vector $\mathbf{w}$ that is an ERM predictor.

Linear Regression

$\mathcal{X}$ is a subset of $\mathbb{R}^d$ and the label $\mathcal{Y}$ is a subset of $\mathbb{R}$.
Goal is to learn a linear function $h : \mathbb{R}^d \longrightarrow \mathbb{R}$ that best approximates the relationship between variables (e.g., predicting the weight of a baby as a function of age and weight at birth).
The hypothesis class is

$$\mathcal{H}_{reg} = L_d = \{h : \mathbb{R}^d \longrightarrow \mathbb{R}, h(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b\}.$$

A loss function for linear regression could ve $\ell(h, (\mathbf{x}, y)) = (h(\mathbf{x}) - y)^2$ (the squared loss).

Least Squares

Least squares is an algorithm that solves the ERM problem for the linear regression predictors with respect to the squared loss.
The ERM problem starts with a traing set S and seeks to find w, where

$$\mathbf{w} = \text{argmin}_{\mathbf{w}} L_S(h) = \text{argmin}_{bfw} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w}'\mathbf{x}_i - y_i)^2.$$

The minimum condition amount to $\frac{\partial L_S(h)}{\partial w_p} = 0$ for $1 \leqslant p \leqslant n$.

We have

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(w'\mathbf{x}_i - y_i)^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}(w_1 x_{1i} + \cdots + w_m x_{mi} - y_i)^2.$$

Therefore,

$$\frac{\partial L_S(h)}{\partial w_p} = \frac{2}{m}\sum_{i=1}^{m}(w_1 x_{1i} + \cdots + w_m x_{mi} - y_i)x_{pi}$$

$$= \frac{2}{m}\left(\sum_{i=1}^{m}(w_1 x_{1i} x_{pi} + \cdots + w_m x_{mi} x_{pi} - y_i x_{pi})\right) = 0,$$

which implies

$$\sum_{i=1}^{m}(w_1 x_{1i} x_{pi} + \cdots + w_m x_{mi} x_{pi} - y_i x_{pi}) = 0,$$

for every p, $1 \leqslant p \leqslant n$.

The last equalities can be written in compact form as $XX'\mathbf{w} = X\mathbf{y}$, which requires solving the equation $A\mathbf{w} = \mathbf{b}$, where $A = XX'$ and $\mathbf{b} = X\mathbf{y}$.

Note that:

- if A is invertible, the solution to the ERM problem is $\mathbf{w} = A^{-1}\mathbf{b}$;
- $\mathbf{b} = X\mathbf{y}$ is a linear combination of the columns of X.

If the linear system $A\mathbf{w} = \mathbf{b}$ has no solution, the "next best thing" is to find a vector $\mathbf{c} \in \mathbb{R}^n$ such that $\| A\mathbf{c} - \mathbf{b} \|_2 \leqslant \| A\mathbf{w} - \mathbf{b} \|_2$ for every $\mathbf{w} \in \mathbb{R}^n$, an approach known as *the least square method*.

We will refer to the triple $(A, \mathbf{w}, \mathbf{b})$ as an *instance of the least square problem*.

Note that $A\mathbf{w} \in \mathbf{range}(A)$ for any $\mathbf{w} \in \mathbb{R}^n$. Thus, solving this problem amounts to finding a vector $\mathbf{w}$ in the subspace $\mathbf{range}(A)$ such that $A\mathbf{w}$ is as close to $\mathbf{b}$ as possible.

Let $A \in \mathbb{R}^{m \times n}$ be a full-rank matrix such that $m > n$, so the rank of A is n. The symmetric square matrix $A'A \in \mathbb{R}^{n \times n}$ has the same rank n as the matrix A. Therefore, the system $(A'A)\mathbf{w} = A'\mathbf{w}$ a unique solution $\mathbf{s}$. Moreover, $A'A$ is positive definite because $\mathbf{w}'A'A\mathbf{w} = (A\mathbf{w})'A\mathbf{w} = \| A\mathbf{w} \|_2^2 > 0$ for $A\mathbf{w} \neq \mathbf{0}$.

Theorem

Let $A \in \mathbb{R}^{m \times n}$ be a full-rank matrix such that $m > n$ and let $\mathrm{b} \in \mathbb{R}^m$. The unique solution of the system $(A'A)\mathrm{w} = A'\mathrm{b}$ equals the projection of the vector b on the subspace $\mathrm{range}(A)$.

Proof

The n columns of the matrix $A = (\mathbf{v}_1 \ \cdots \ \mathbf{v}_n)$ constitute a basis of the subspace $\mathbf{range}(A)$. Therefore, we seek the projection $\mathbf{c}$ of $\mathbf{b}$ on $\mathbf{range}(A)$ as a linear combination $\mathbf{c} = A\mathbf{t}$, which allows us to reduce this problem to a minimization of the function

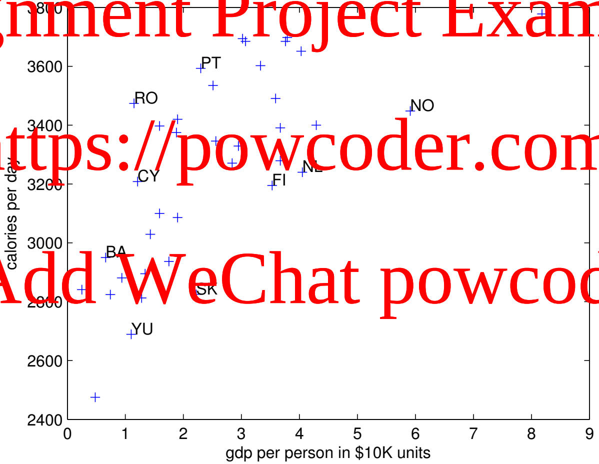
$$\begin{aligned} f(\mathbf{t}) &= \parallel A\mathbf{t} - \mathbf{b} \parallel_2^2 \\ &= (A\mathbf{t} - \mathbf{b})'(A\mathbf{t} - \mathbf{b}) = (\mathbf{t}'A' - \mathbf{b}')(A\mathbf{t} - \mathbf{b}) \\ &= \mathbf{t}'A'A\mathbf{t} - \mathbf{b}'A\mathbf{t} - \mathbf{t}'A'\mathbf{b} + \mathbf{b}'\mathbf{b}. \end{aligned}$$

The necessary condition for the minimum is

$$(\nabla f)(\mathbf{t}) = 2A'A\mathbf{t} - 2A'\mathbf{b} = 0,$$

which implies $A'A\mathbf{t} = A'\mathbf{b}$.
The linear system $(A'A)\mathbf{t} = A'\mathbf{b}$ is known as the *system of normal equations* of A and $\mathbf{b}$.

We represent (using the function `plot` of `ML`), the number of calories consumed by a person per day vs. the gross national product per person in European countries



ccode	gdp	cal	ccode	gdp	cal
'AL'	0.74	2824.00	'IT'	3.07	3685.00
'AT'	4.03	3651.00	'LV'	[illegible].43	302[illegible].00
'BY'	1.34	2895.00	'LT'	[illegible].59	3397.00
'BE'	3.79	3698.00	'LU'	8.18	3778.00
'BA'	0.66	2950.00	'MK'	0.94	2881.00
'BG'	1.28	2813.00	'MT'	2.51	3535.00
'HR'	1.75	2937.00	'MD'	0.25	2841.00
'CY'	1.21	3208.00	'NL'	4.05	3240.00
'CZ'	2.56	3346.00	'NO'	5.91	3448.00
'DK'	[illegible].67	3391.00	'PL'	1.88	3375.00
'EE'	[illegible].90	3086.00	'PT'	2.30	3593.00
'FI'	3.53	3195.00	'RO'	1.15	3474.00
'FR'	3.33	3602.00	'RU'	1.59	3100.00
'GE'	0.48	2475.00	'YU'	1.10	2689.00
'DE'	3.59	3491.00	'SK'	2.22	2825.00
'GR'	3.02	3694.00	'SI'	2.84	3271.00
'HU'	1.00	3420.00	'ES'	2.95	3[illegible]29.00
'IS'	3.67	3279.00	'CH'	4.29	3400.00
'IE'	3.76	3685.00			

We seek to approximate the calorie intake as a linear function of the gdp of the form

$$\text{cal} = r_1 + r_2\,\text{gdp}.$$

This amounts to solving a linear system that consists of 37 equations and two unknowns:

$$\begin{aligned} r_1 + 0.74 r_2 &= 2824 \\ &\vdots \\ r_1 + 4.29 r_2 &= 3400 \end{aligned}$$

and, clearly such a system is inconsistent.

We augment the data sample matrix by a column that consists of 1s to accommodate a constant term r_1; thus, we work with the data sample matrix $B \in \mathbb{R}^{37\times 2}$ given by

$$B = \begin{pmatrix} 1 & 0.74 \\ \vdots & \vdots \\ 1 & 4.29 \end{pmatrix}$$

whose second column consists of the countries' gross domestic products in \$10K units. The matrix

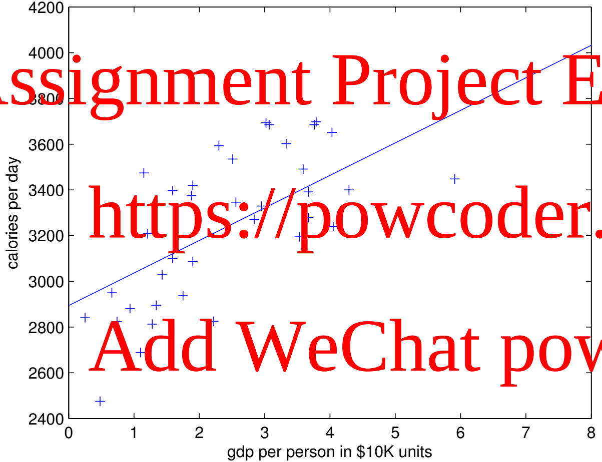
$C = B'B$ is

$$C = \begin{pmatrix} 37.0000 & 94.4600 \\ 94.4600 & 333.6592 \end{pmatrix}.$$

Solving the normal system using the `ML` statement $\mathbf{w} = C \setminus (B' * b)$ yields

$$\mathbf{w} = \begin{pmatrix} 2894.2 \\ 142.3 \end{pmatrix},$$

so the regression line is $cal = 142.3 * gdp + 2894.2$.


4200
4000
3600
3400
3200
3000
2800
2600
2400
calories per day
0
1
2
3
4
5
6
7
8
gdp per person in $10K units