

Over the AI ― AIの向こう側に(17):

至宝の人工知能 ～問題に寄り添い、最適解をそっと教えてくれる

<http://eetimes.jp/ee/articles/1711/30/news019.html>

先人たちにより開発され、磨かれてきた「至宝の最適化アルゴリズム」。本当はこれを軽々しく「AI」とは呼びたくない……い……という気持ちをぐっところえ、AI技術として解説します。「試験前の一夜漬け」「雪山遭難」「井戸堀り」の例を使って、説明していきます。繰り返しますが、最適化アルゴリズムを軽々しく「AI」という言葉で片付けたくはないんですよ、本当は。

2017年11月30日 11時30分 更新

[江端智一, EE Times Japan]



今、ちまたをにぎわせているAI(人工知能)。しかしAIは、特に新しい話題ではなく、何十年も前から隆盛と衰退を繰り返してきたテーマなのです。にもかかわらず、その実態は曖昧なまま……。本連載では、AIの栄枯盛衰を見てきた著者が、AIについてたっぷりと検証していきます。果たして”AIの彼方(かなた)”には、中堅主任研究員が夢見るような”知能”があるのでしょうか――。[⇒連載バックナンバー](#)

数学＝だまされないための便利な道具

私が学生のころ、ネパールに旅行中、現地の小料理屋で知りあった、私と同じ日本人のバックパッカーと料理をシェアしていた時の話です。

私はネパールの地ビールを楽しんでいましたが、彼はアルコールが飲めませんでした。さらに、そのバックパッカーの友人が、会食の終わりに、参加してきました。

会計の時、この3人の支払い(割り勘)は結構面倒なことになります。なにせ、アルコールを飲む人、飲まない人、食事の時間などを勘案して、全員がそこそ納得できる金額にしなければな



画像はイメージです

らないからです。

もちろん異国を旅する日本人同士、どんな割り勘の金額に決まったところで、誰も文句は言わないでしょう(むしろ、多めに支払うことを申し出る人の方が多いのではないのでしょうか)。

で、その時、私が何をしたか ― レシートの金額を確認した後で、そのレシートの裏に、連立方程式を書き始めたのです。

私と旅人2人の支払金額を、それぞれ未知数 x, y, z として、アルコール有無とテーブル着席時間をざっくり係数とした、単純な線形方程式を書いただけですが、

― 飲み会の割り勘に、連立方程式を使った人を初めて見た

と、2人の旅人は、相当驚いていたようです。

私は、こういう(飲み会などで使える)数字ネタをいくつか持っています。

その1: 学生のころ、異性の友人に誘われてフラフラと、マルチ商法ビジネスを行うとある会社の集会に参加させられたことがあります。その集会で、『そのビジネスモデルは数回、階層を回しただけで、世界の全人口を越えて破綻するのですが、最下位層から開始する私は、どうやら利益を得られるのですか?』と指摘して、事実上、その集会をぶち壊したことがありますし、

その2: ガス給湯器のセールスマンに対して、減価償却期間の計算結果を見せて、『では、御社の製品を購入することによって、この計算結果を越える金銭メリットを提示してください』と口にする前に、逃げていかれたことがありますし、

その3: 太陽光パネル^{*)}に関して、東京都の優遇措置の話を出してくるセールスに対して、『そんな優遇措置が、今後本当に30年間も続くと思います? 都民全員が、その優遇を受けたら、東京都は財政破綻しますよね』と疑問を口にしただけで、その会社は、二度とわが家を訪問してこなくなりました。

*) 関連記事: [「電力という不思議なインフラ\(前編\)～太陽光発電だけで生きていけるか?～」](#)

まあ、つまり、何と言うか、

― 数字や数学は、人にだまされない人生を過す上で、便利な道具

と言えそうです。

以前、わが国が「ゆとり教育」という試みを行った時、「二次方程式などは社会へ出て何の役にも立たないので、このようなものは追放すべきだ」という、奥方(作家)の意見を間に受けて、わが国の数学力の低下に最大限の貢献をした作家がいました^{*)}。

*) 上記のフレーズをコピーして検索すれば、その作家はすぐに分かりますので、御興味のある

方はどうぞ。

『こういう奴がいるから、自分の借金の金利の計算もできないまま、自己破産していく人間が断たないのだ』と思ったことを、よく覚えています。

□

ところで、人工知能(AI)技術のベースとなる数学が、高校数学で十分であることは、意外に知られていません。

本当です。AI技術のサンプルプログラムを、暇さえあれば、片っぱしから試し続けているこの私が言うのですから、間違いありません。

かなり乱暴に言えば、数列、標準偏差、微分積分の考え方が分かっていたら十分で、数式の暗記すら不要(というか有害)です。考え方さえ分かっていたら、実際の数字の計算は、電卓、「エクセル」、パソコンにやらせれば良いのです。

しかし、これ(「考え方」の理解)が、多くの人にとって容易なことではないことも、私はよく分かっているつもりです*)。

*) 私は「英語」で同じ境遇を痛感しています。「[「海外で仕事をしたい」なんて一言も言っていない!](#)」でご紹介している通り……。

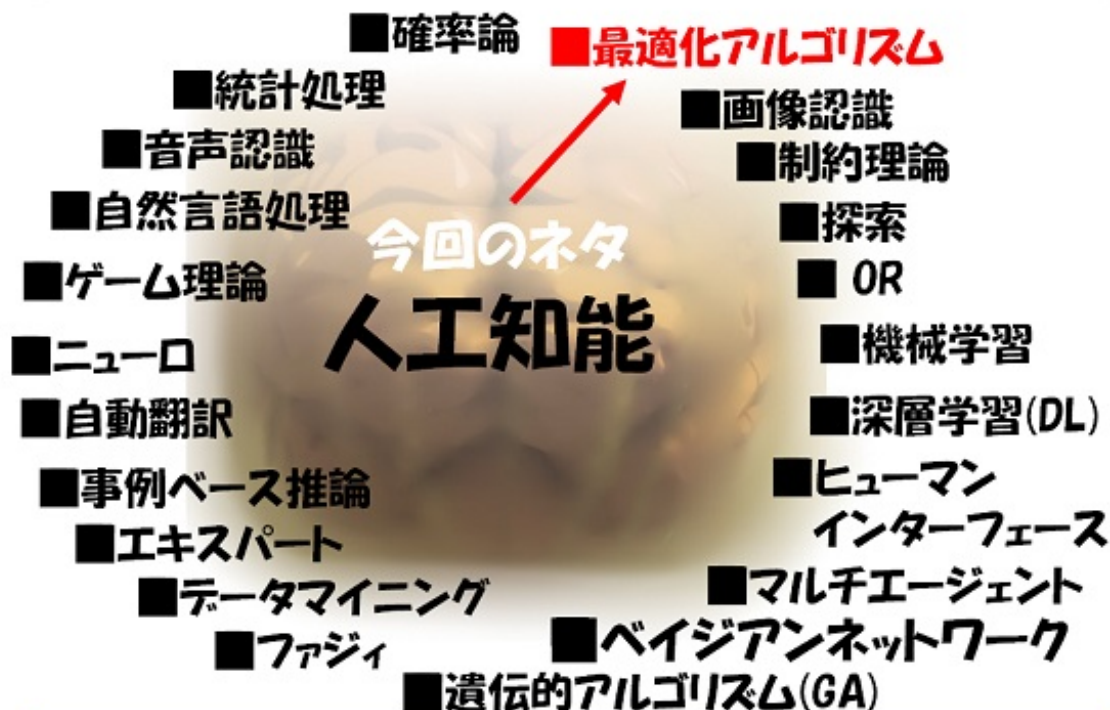
そもそも、義務教育や高等教育の数学の授業って、この私ですら「ちょっとした拷問」だと感じるくらい、退屈ですから。

ですから、今ここに、私は高らかに声を上げて、数学教育の改革を提言し……(以下省略(参照:[著者のブログ](#)))

先人たちが築き上げた「至宝の最適化アルゴリズム」

こんにちは、江端智一です。今回は、「最適化アルゴリズム」についてお話致します。

“人工知能”という技術は存在しない



(原則として)関連のない個別の技術

また、毎度のことですが、「『最適化アルゴリズム』が“AI技術”なのかどうか」については、今回も『江端AIドクトリン』に基づいて私が ―― と言いたいところなのですが、実は、今回ばかりはどうしても“AI技術”と呼びたくないのです。

―― 先人たちが築き上げてきた「至宝の最適化アルゴリズム」を、軽々しく「AI技術」など呼んで欲しくない

という思い入れが、私にはあるからです。

奥村先生の「[C言語による最新アルゴリズム辞典](#)」(奥村晴彦、1991年 技術評論社)は、今なお、私が、オフィス(研究所)の中を20mも歩けば、誰かのデスクの上に1冊は発見できるくらいの名著です。

大学入試で使うような数式の方程式なんぞ使わなくても、「最適なアルゴリズムを使えば、コンピュータの力を使って解が得られる」ということを、魂のレベルで教えてくれた本で、「私のバイブル」といっても過言ではありません(今でも使っています)。

が、まあ、私に“最適化アルゴリズム”に思い入れがあったとしても、このままでは連載を進められないので、“最適化アルゴリズム”を“AI技術”と見なすことにします(不本意ですが)。

第3次AIブームをけん引する「強化学習」や「深層学習」

今回の第3次AIブームにおきましては、強化学習や深層学習が、そのブームをけん引しています。私は、その理由の1つに「神秘性」があると思っています。

私は学生のころ、逆伝搬学習の効率の悪さに（文字通り）『地獄を見た』のですが、深層学習（ディープラーニング）は、この問題を改善して、今やいろいろな分野でニューラルネットワークを使えるようにした、という点において革命的だったと言えます。

また、強化学習は、「結果」だけを使って、そのオペレーション全体を再評価するという方式を使って、「何も教えなくても、放っておけば、勝手に学習していく」というプロセスが、世間の人々の度肝を抜きました。

その度肝を抜かせた映像の1つがこちらの[「ブロック崩し」のゲーム映像](#)です。

強化学習で、コンピュータに教える知識は、ゲームオーバーになった時の得点のみで、ジョイスティックを「どのように動かせば良いのか」は一切教えません。コンピュータは、「良い得点」を取った時の動かし方が——その理由は知らんが——とにかく「良い動かし方」と評価された動きを学習するだけです。

これは、「新人の指導を押しつけられた指導員」にとって、夢の具現化と言えます。



なぜなら、強化学習の方法を、新人教育に当てはめれば、「馬鹿野郎!」と「よくやった!」の2フレーズを使うだけで指導ができるわけですから、こんなに楽チンな方法はありません。

しかし、この職人気質（かたぎ）の手法は、現代のエンジニア教育においては、完全に否定されております。あまりにも効率が悪過ぎるからです。

ニューラルネットワークが、解決したい問題に対して適切な構成を込めているかどうかは、運とセンスにかかっており、しかも超高速な計算デバイス（GPUなど）がないと、全く使いものにならない、という特徴があります。

また、ゲームを使った強化学習による学習プロセスの映像では、強化学習が簡単に完了しているかのように見えますが、実際には、膨大な回数のゲームを続けています。仮に数百、数千回ものゲームをやらせ続けたら、1回1分の時間がかかるとして、5000回で83時間、約10営業日かかることになります）。

強化学習とは——もし、指導員が新人にゲームの攻略法を教えるというカリキュラム（どんなカリキュラムだ）があったと仮定した時、「ゲームのルールは教えん。ゲームの操作方法も教えん。とにかく、表示される点数を高くする方法を、自分で見つけ出せ」といって、新人に、2週間の時間を与えて、ひたすら5000回ゲームを繰り返させる——そういう学習方法なのです。

このケースなら、指導員が、たった一言「ボールに追従してバーを動かせ」とだけ指導すれば

、このカリキュラムは完了です(所要時間5秒)。

深層学習や強化学習は、なぜ「そのように動いたか」を説明できない

仕事の現場においては、仕事の背景、ルール、ノウハウを、ロジカルに説明し、問題があればその問題点を明かにして、次のプロセスで改善させる ―― そういう学習方法の方が、優れているに決まっています ―― たとえ「神秘性」はなくとも。

実際、深層学習や強化学習は、制御分野、特に、超高度なリアルタイム性や安全性を要求される分野では、現在のところ、全く利用されていませんし、今後も利用される可能性は低いでしょう。

理由は明快です。深層学習や強化学習は、なぜ「そのように動いたか」を説明できないからです。本連載の別の回「[困惑する人工知能 ～1秒間の演算の説明に100年かかる!?](#)」を読んでもいただければ、その理由が分かるかと思います*)。人間だって、自分の行動をきちんと説明できる訳ではありませんが、それでも、(たとえ“後付け”であったとしても)「言い訳」を語れなければ、社会の構成員としては認められません。

*) 現在、この問題に対して、“Explainable Artificial Intelligence=説明可能な人工知能”の研究が盛り上がっています。

個人的な意見ですが、ニューラルネットワークや強化学習で、知識を習得したAI技術を搭載したジェットコースターなんぞ、私は絶対に乗りません。

しかし、最適化アルゴリズムによって制御されているジェットコースターなら、乗車しても良いと思っています。たとえ私がそのジェットコースターで事故死しても、家族にその理由を説明してもらえと思えるからです。

もっとも、私の家族が、私を殺した「最適化アルゴリズム」の内容や、そのベースとなる数学のコンセプトを理解できるかどうかは、別問題ですが。

「最適化アルゴリズム」とは、つまり何なのか

さて、「最適化アルゴリズム」を、私なりに定義すると、

- 酷く複雑で大規模で面倒くさい問題に対して、その解答を導くもので
- コンピュータプログラムとして記述可能なもので
- (理屈の上ではともかく) 現実にはコンピュータの計算能力の助けなくしては使えないものの

です。

今回、この「最適化アルゴリズム」をカテゴリーで分類しようとしたのですが、表を作っても、あまり面白くありませんでした*)。

＊) 線形計画、整数計画、非線型計画とか、離散系とか連続系とか、そんなところです。

ですので、今回は、「最適化アルゴリズム」の著名なアプリケーションを羅列してみました。

最適化アルゴリズムの適用例

問題分類で見るのはやめた、江端が知っているもの
(一部はやってみたもの)だけを羅列

問題	イメージ	目的
(1) ナップザック問題	(後述)	(後述)
(2) 施設配置問題	部品工場を、どこに新築するか	工場建設費用と工場輸送の費用を最小にすること
(3) 巡回セールスマン問題	複数のお客を、どの順番で訪問するか	最短時間、最小費用で、全部のお客を回ること
(4) 乗務員スケジューリング問題	空港を移動し続けるパイロットやCAの勤務をどう管理するか	無駄なく航空機に載せて、かつ休暇も組み込むこと
(5) 看護婦スケジューリング問題	24時間体制で、患者を受け入れ可能なスケジュールを、どう作るか	ベテランと新人をバランス良く組み合わせ、夜勤や日勤、休日にもバランスよく組み立てること
(6) 切り出し/詰め込み問題	1枚の金版から、部品を、どう切りだすか	色々な形状の部品を切り出す。切り出しの残りカスを最小にすること

どの問題も、先人の成果があり、
今なお、研究されている

一見すると、「これって、そんなに難しい問題か?」と思えるかもしれませんが、皆さんも、これ

までの人生で同じような問題で苦労してきたはずです。

- 時間配分を無視して、つまらんスピーチをダラダラと続ける上司によって狂わされる、発表会タイムスケジュールの復旧方法
- 結婚披露宴に、一度も出会ったことがない親戚を呼ぶために（両親からの強い圧力）、親しい友人の何人かを、席数の都合で切り捨てなければならない時の決断

もっと極端な例では、

- 「仕事と私、どっちが大切なの！／なんだ！」と、その場限りのコスト計算しかできず、仕事を辞してしまった後の「『あなた』の価値の下落」を考慮しない、感情的な恋人との折り合い

こうした問題は、それ自体解決方法が難しいのに加えて、もっと大規模なスケール（発表会が同時に複数進行しているとか、結婚式の正体客が1万人のオーダーとか）を想定してみれば、その困難性は明らかだと思います。

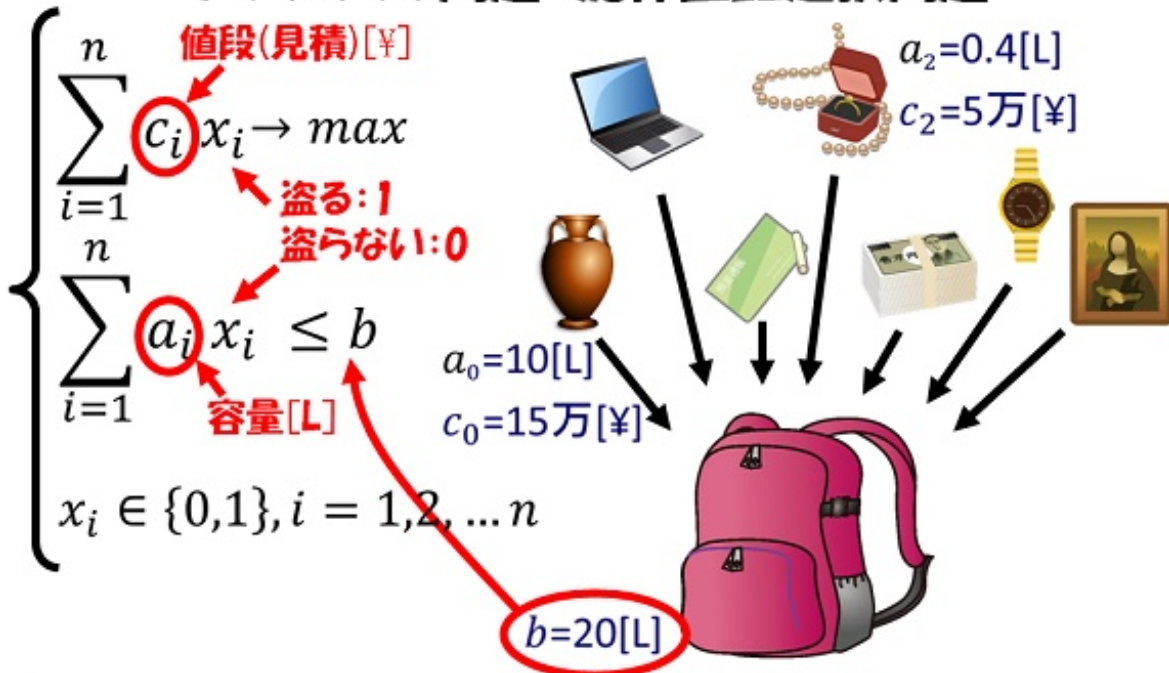
しかし、「その問題の性格」に応じて「何とかしてみせる」のが、最適化アルゴリズムなのです。これは、「問題の性格」を全く教えずに、自動的に学習させる「強化学習」の対極に立つ手法と言えます。

線形計画法の入門中の入門

では、最初に、線形計画法の入門中の入門である「ナップザック問題」を簡単に説明します。

線形計画法(1)

ナップザック問題=泥棒盗品選択問題



ナップザックに入る範囲で、盗品額を最大にする

私は、この問題を「泥棒盗品選択問題」として把握しています。簡単に言えば、持ち運べる容量(このケースでは20リットル)に上限があるという制約下において、合計の盗品金額を最大にする、という問題です。

この問題だけでなく、さまざまな問題には、それぞれの問題に応じた最適化アルゴリズムがありますが、1つ覚えておいていただきたいことに、どの最適化アルゴリズムであっても、「最も良い解を見つける方法はない(だろう)」ということです。

この「最も良い解」のことを「大域的最適解」といいますが、この「大域的最適解」を調べるには、全ての組合せを試してみる必要があります。しかし、その組合せの数が大きすぎて、コンピュータをもってしても計算が終了しないのです*)。

*) 例えば、「宇宙の終わりまで計算し続けても終了しない」という言い方で語られることが多いです。

上記の図の例では、総当たりしてもたいした組み合わせ数ではありませんが、もし盗品候補が1000以上もあれば、「大域的最適解」は諦めなければなりません*)。

*) 組み合わせ数: 1.07×10 の301乗 / (Intelの「Core i7」で計算したら) 2.5×10 の282乗年 (ちなみに宇宙の年齢は、 1.38×10 の10乗年)

最適化アルゴリズムは、組み合わせの総当たり戦をやめて、私たち人間の世界において「最適化」といっても問題のないレベルの解を、実用的な時間内（数秒から、長くても24時間以内）に導き出すものです——つまり「最適化アルゴリズム」は「(大域的)最適解アルゴリズム」ではないのです。

ところで、ちょっと話は逸れますが、私、日本人の多くに数学が嫌われている理由って、こいつ（右図）だと思っているんですよ。

なんですかね、この不快な呪文のような記述。エラソーで、タカビーな感じもしますよね。『フッ、お前らには、この数式の意味、分かんねーだろうな』って、バカにされているような気すらします。バリバリの現役のエンジニアである私がそう感じるのですから、理系でない人の不快さは、察して然るべきでしょう。

しかし、私からも、皆さん——特に、日常的に、数式を使わず、プログラムとも縁のない方に——文句があります。

「上記のような呪文……でなく、数式をコンピュータに入力すれば、コンピュータが勝手に計算を始める——などと思っているんじゃないか」と感じることもあるからです。

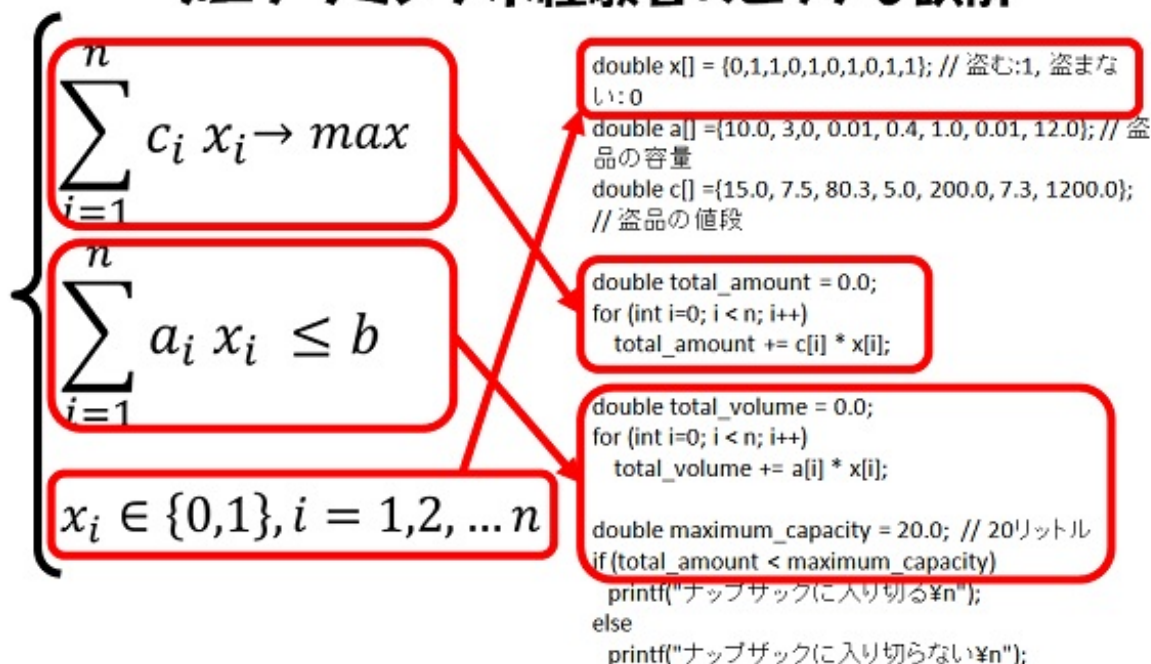
特に、「AI幻想（AIが人類を滅ぼす/AIが職を奪うなど）」を持っている人の多くは、上記のような勘違いをしていると、確信しています。

——冗談ではありません。

$$\begin{aligned} \sum_{i=1}^n c_i x_i &\rightarrow \max \\ \sum_{i=1}^n a_i x_i &\leq b \\ x_i &\in \{0,1\}, i = 1, 2, \dots, n \end{aligned}$$

AI幻想の一つのバックグラウンド

プログラミング未経験者のビッグな誤解



コンピュータに、数式を書き込めば 「自動的に答えが出てくる」という幻想

私たち、コンピュータシステムに関わるエンジニアは、この呪文……ではなく数式を、完全に理解した上で、コンピュータが理解できる言語（プログラム言語）に書き換える作業をしなければならないのです。

それに、数式には具体的な数値が入っていませんが、プログラムには実際の値を入れないと、ウンともスンとも言いませんので、現実世界の問題を「数値に変換」して、「その数値を手で入力」し、そして、「プログラムの間違いを修正」する作業を延々と続けなければならないのです。

こんな馬鹿野郎な機械（コンピュータ）ごときに滅ぼされる人類なら、とっとと滅ぼされてしまえ——

と呟きながら、毎日10時間近くもコーディングし続けているエンジニアがいます—— 私です。

線形計画法を“試験の一夜漬け”で考えてみる

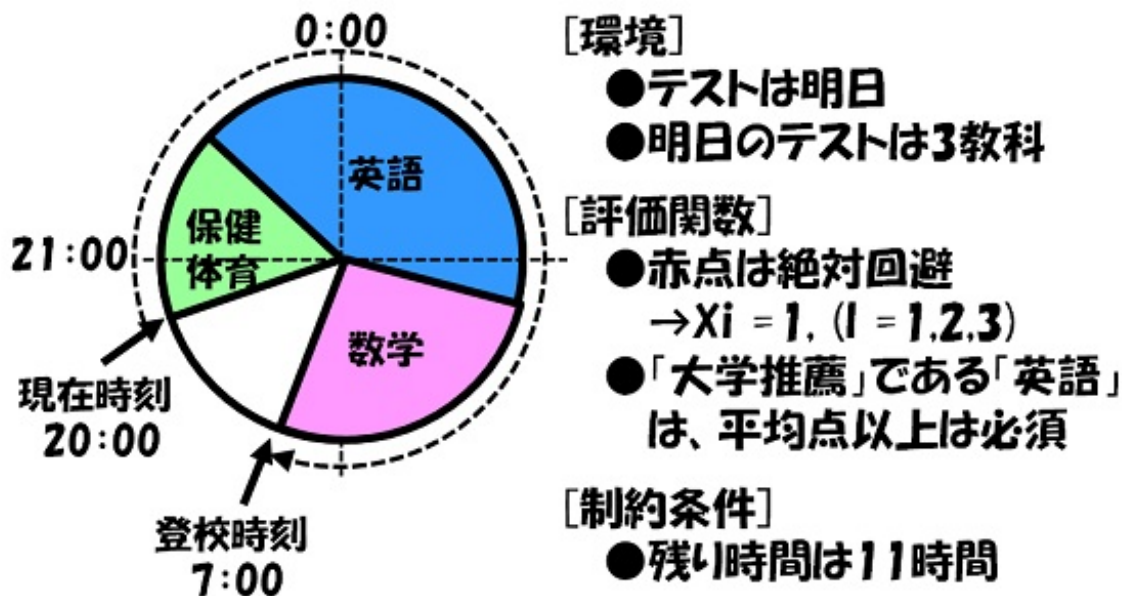
さて、話を戻します。

最適化アルゴリズムには、最低3つの内容を加えなければなりません。「環境」「評価関数」そして「制約条件」です。

この話は面倒くさそうですが、これを、定期試験の「一夜漬けスケジュール問題」に当てはめて考えて記述してみれば、比較的簡単に理解できます

線形計画法(2)

一夜漬けスケジュール問題



人生は、いつだって「線形計画」

一夜漬けをしたからといって、評価関数の内容を達成できる訳ではありませんが、取りあえず、スケジュールの戦略を立てることはできます。

この問題では、いくつかの最適化戦略をとり得ますが、ある教科を優先的に取り扱えば、他の教科が不利になります。これを「パレート最適」といいます。

この問題では、(1) 赤点を回避するためには、特定の教科の勉強を「捨ててかかる」ことはできない、(2) 「英語」だけは、テストの全範囲の勉強が必要であり、(3) 「数学」は、問題にヤマを張っていく、という戦略が立ちます。この戦略を元に、残り時間11時間の時間配分を行うことになります。

これは、最適化アルゴリズムにおける、線形計画といえるものです。

しかし、人間とは不思議なもので、こういう追い詰められた状況になると、なぜか、普段なら絶対にしないようなこと——例えば、「太宰」とか「芥川」とかに手を出して、読みふけてしまい、気が付けば午前2時を指す時計に青冷める、ということがあります——「ああ、本当に、私って『人間失格』…今すぐ『蜘蛛の糸』でも垂れてこないかな……」と呟きながら。

最適化アルゴリズムの非線形計画法を“雪山の遭難”で考えてみる

では、次に、非線形計画法の考え方の一例をご紹介します。

これは、雪山で遭難し、夜間の急激な温度低下で、翌朝までの生存の確率が絶無に近いと判断した時のパラダイムで説明することができます。

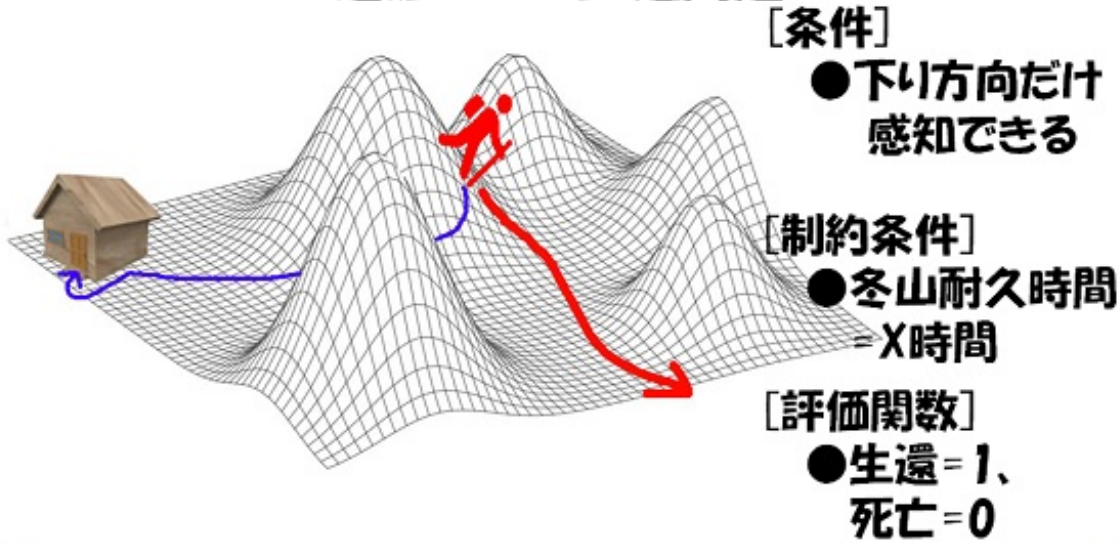
ブリザードを避けるような場所もない登山者は、危険を承知で下山を決意します。



ただし、暗闇と吹雪で視界ゼロ。地図もなく自分の位置も不明。登山者が知り得る情報は、その方向が「上り」か「下り」か、そして、その角度（斜度）だけです。

非線形計画法(1)

遭難からの生還問題



遭難者には、自分の位置は分からない

この時に、登山者のとり得る戦略は1つしかありません。その場、その時における、下り傾斜が最大の方向に降りていくことだけです。これを、「最急降下法」といいます。

このシンプルな方法では、最短時間で下山に成功する確率が最も高くなりますが、しかし、最も急な斜面を下ったところで、そこが単なる谷底だったという場合があります。こうなると、どの方向に向いても、下ることができなくなり、私の死亡は確定的です。

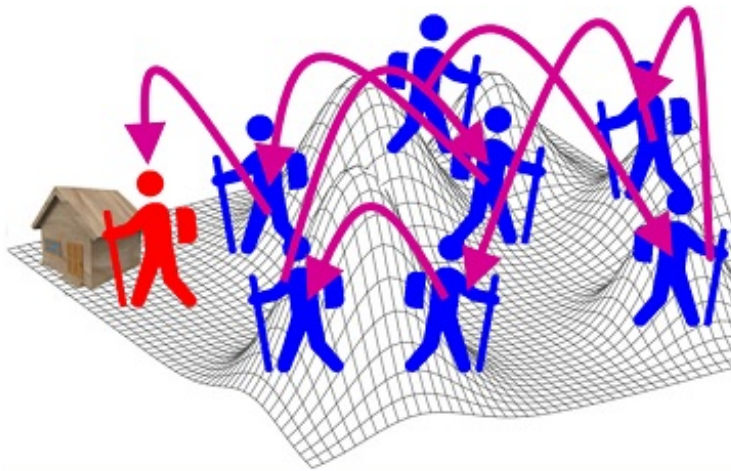
この状態を、局所的最適解（ローカルミニマム）に陥ったといいます。

このように、最急降下法は、初期値（遭難した場所）によって、大域的最適解（山小屋）に到達できるかどうか、確定的に決まってしまうのです。

最適化アルゴリズムの研究の歴史とは、乱暴にまとめれば、この「ローカルミニマム」との闘いであった、とも言えます。

非線形計画法(2)

ワープする / 自分のコピーを作る …とか



●シミュレーテッド
アニーリング(SA)

●遺伝的アルゴリ
ズム(GA)

●滑降シンプレッ
クス法

…その他山程

現実世界の例で説明するのは、無理

そこで、いろいろな方法が考案されてきました。

例えばシミュレーテッドアニーリングという方法は、下山時開始時にはポンポンと空を飛ばせて、遠くの下山場所に変化させ、凍死直前には、飛ばないようにすることで、山小屋への到着率を高める、というものです(『空を飛ぶ体力があるなら、下山は簡単だろう』というツッコミは無しの方で。これは、最適化アルゴリズムの話です)。

また遺伝的アルゴリズム(GA)^{*}は、自分のクローンを大量生産して、全員で下山情報を共有するというものです(『そんな魔法が使えるなら、下山は簡単……』(以下省略))。

^{*} 関連記事: [「抹殺する人工知能 ～ 生存競争と自然淘汰で、最適解にたどりつく」](#)

「最適化アルゴリズム」は、解決する問題にベッタリと寄り沿って、その問題を解くための「最適化」を極めるアルゴリズムですが、いくつかの問題点があります。

第1の問題点は、前述した通り、問題に応じた適当なアルゴリズムを選ばなければならないことが挙げられますが、これは、踏んできた場数(試してみたアルゴリズムの数)に影響を受けます。

第2の問題点は、強化学習とは真逆の発想で「課題や解法を完璧に理解しないと使えない」点も挙げられます(当然ですが)。

さらに、第3の問題点は、計算終了時間が、事前に読み切れないということです。「これまで、1分間で解けていた問題に、未知数をたった1個追加しただけで、計算時間が100年を超えることになってしまった」、なんてケースは、ざらにあります(あまり驚くことではありません)。

滑降シンプレックス法を、井戸堀で考えてみる

ここからは後半になります。この連載の後半は、「私の身の回りの出来事」を使った、「数式ゼロ」のAI解説になります。

本日は、前半に引き続き「最適化アルゴリズム」の1つである「滑降シンプレックス法」についてお話したいと思います。

私は、エージェントを作るときには「ファジィ推論」、パラメータを抽出する時は「遺伝的アルゴリズム(GA)」を使って、ちゃっちゃと実装してしまうのですが、この他にも、「滑降シンプレックス法」も、よく使います。

そこで、今回は、最適な井戸を掘る場所を発見する手法をイメージして、この「滑降シンプレックス法」の使い方をご紹介します。

実は私、最近町内会で発行している会報の記事を担当することになり、現在、「地震発生時のシミュレーション」のシリーズを執筆中です。

それで今回、市役所や関係者にインタビューを行った上で、机上シミュレーションをやってみたのですが、正直、青冷めました。まあ、ここでは詳しく述べませんが、「苦痛の中で死んでいきたくなかったら、ちゃんと被災時の準備をしておけ」とだけ申し上げます。

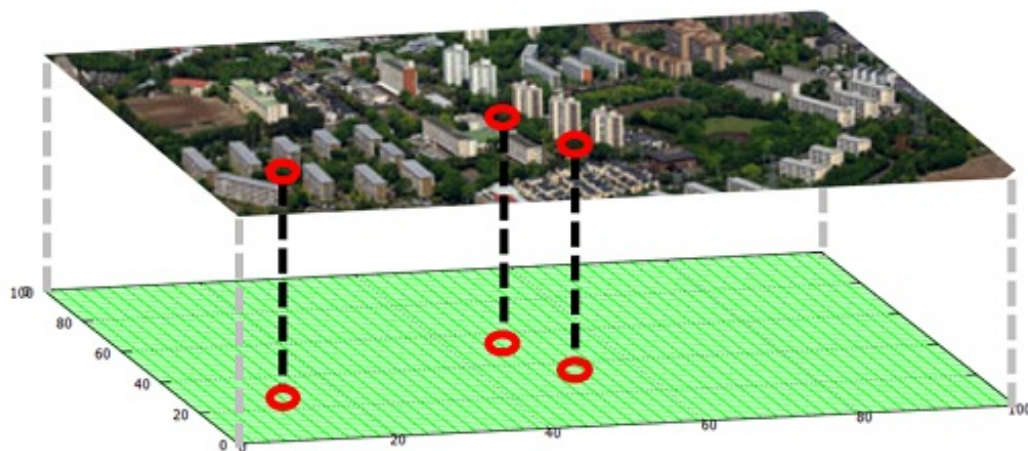


特に、飲料水の確保が相当に厳しいことになることが分かり、今後は新しい井戸の開発が必要になると考えました。しかし井戸を掘ることは、結構難しいのです。良い水源を見つけなければならないからです。

そこで、現存している井戸と、試掘をベースに、水源を見つける戦略アルゴリズムとして、「滑降シンプレックス法」を適用するケースを、想定してみました。

どこに井戸を掘ればいいのか(1)?

現状、町内には3つの井戸がある



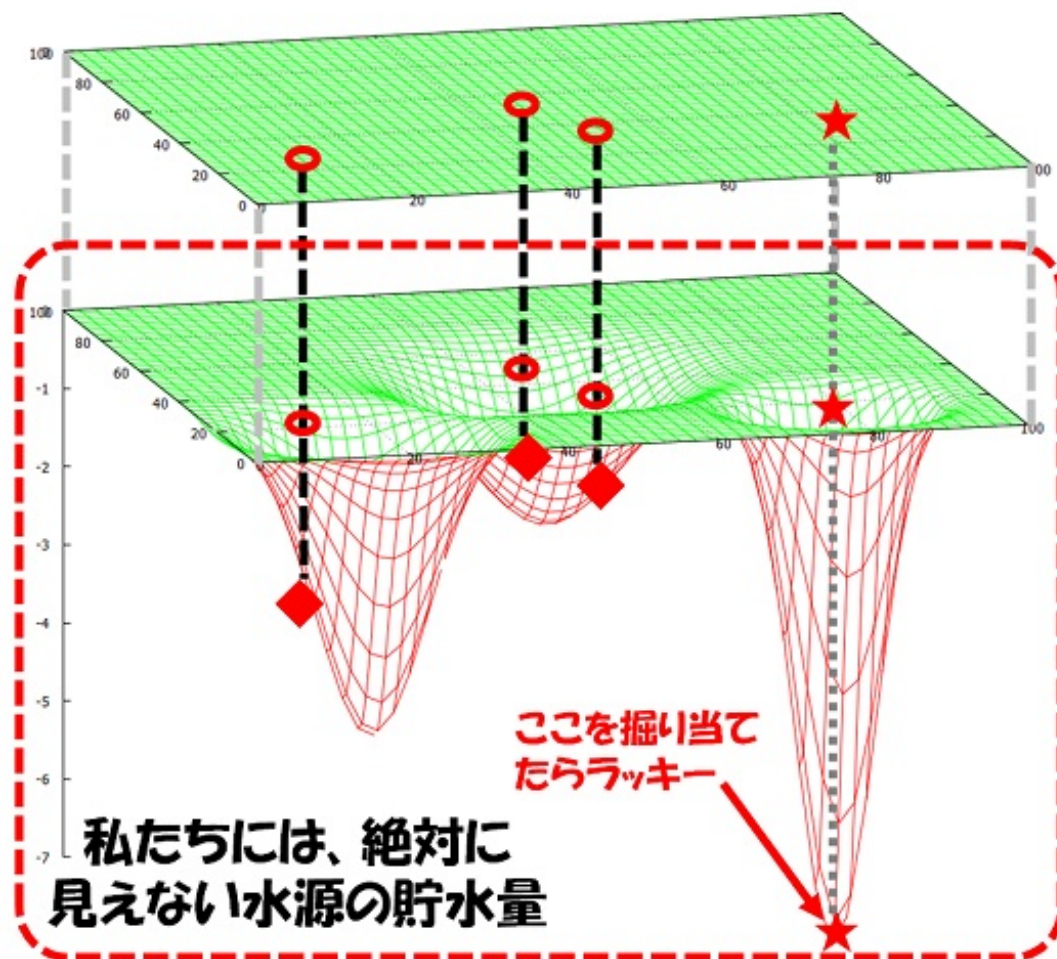
現存の井戸の取水量では全然足りない

上図のように、現状、町内には3つの井戸があるものと仮定します。そして、町内には、いくつかの貯水量を確保できる場所があるのですが、当然、これは私たちには分かりません（神サマだけが知っている）。

私たちは実際に井戸を掘る前に、簡単な試掘を行えることとします。この問題の目的は、試掘の回数を少なくして、貯水量の大きな水源のある場所を発見することにあります。

どこに井戸を掘ればいいのか(2)？

神サマだけに見えている水源の貯水量



見えないう水源を、少ない試掘数で辿りつく

滑降シンプレックス法は、現在ある情報を使って、次の4方向で試掘するものです。

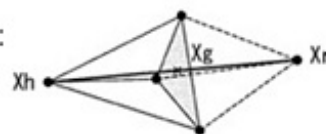
どこに井戸を掘ればいいか(3)?

テトラメに試掘するのではなく、4方向に定める

操作1 反射 x_h を以下の x_r で置き換える:

$$x_r = (1 + \alpha)x_g - \alpha x_h \quad \alpha=1 \text{ がいい}$$

ただし $\alpha > 0$ は反射係数



操作2 拡大 $x_g - x_r$ 方向に沿って x_r を以下の x_e に置き換える:

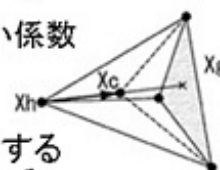
$$x_e = \gamma x_r + (1 - \gamma)x_g \quad \gamma=2 \text{ がいい}$$



操作3 縮小 x_h を以下の x_c で置き換える:

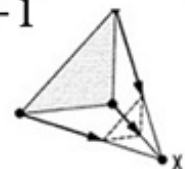
$$x_c = \beta x_h + (1 - \beta)x_g, \quad \text{ただし } 0 < \beta < 1 \text{ は縮小係数}$$

$\beta=0.5$ がいい



操作4 収縮 シンプレックス全体を x_i の方向へ半分に縮小する

$$x_i = 0.5(x_i + x_i), \quad \text{ただし } i = 1, \dots, n+1$$



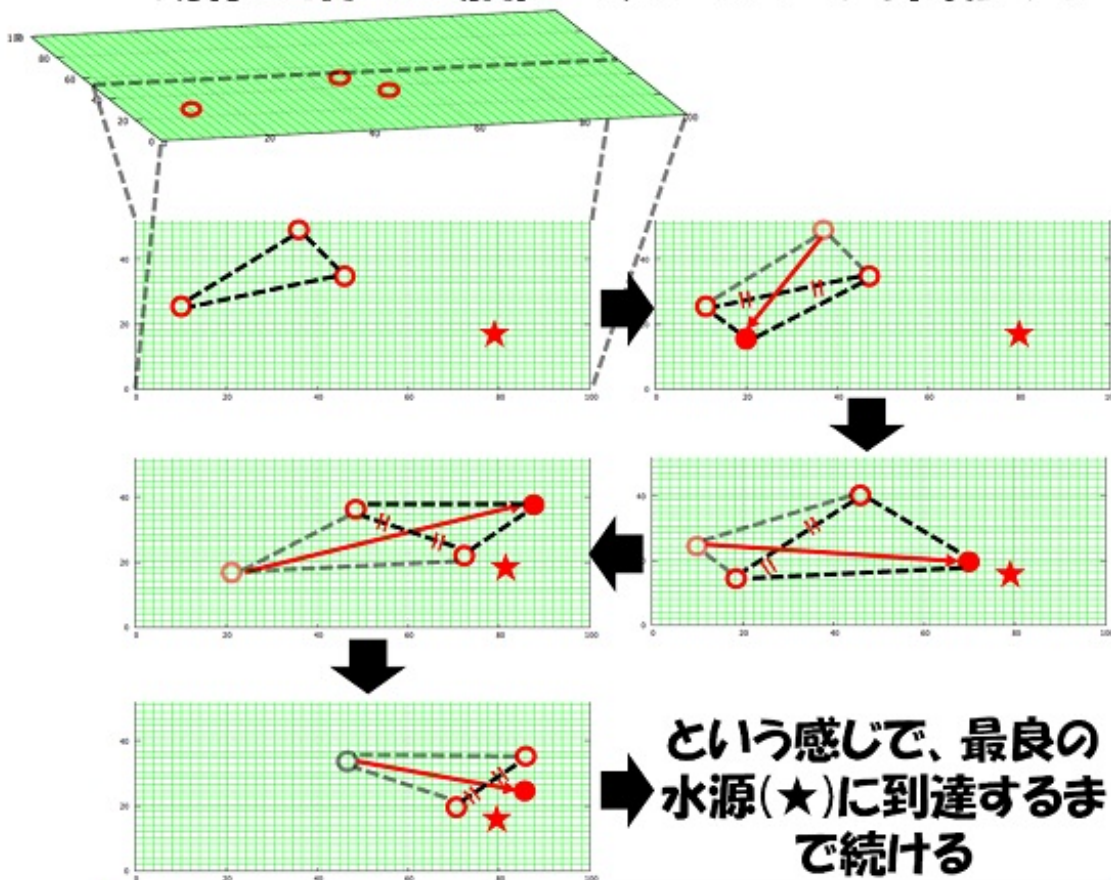
逆方向に/遠くに/近くに/縮めて
の4パターンで試掘する

初期の3つの井戸の情報をベースに、このシンプレックスを繰り返していきます。この4パターンによって、ローカルでの探索(局所探索)と、ローカルミニマムの回避を同時に実施します。

今回のケースでは、試掘を繰り返しながら、成績の良い3点で三角形を何度も作り直して、最も良い解の水源を探し続けていくことになります。

どこに井戸を掘ればいいか(4)?

既存の井戸の場所から、アプローチを開始する



それでも、大域的最適解に至る保証はない

私が、この滑降シンプレックス法を気に入っているのは、ベクトルとか解曲面の微分とかの計算をする必要が一切全くなく、単に良い値が発見できたら、そこに三角形の頂点を動かすだけ、という「単純さ」にあります。

もちろん、この手法でも、ローカルミニマムの問題からは逃げられませんが、それでも、時間をかけずに、簡単に、そこそこの最適解が欲しい時には、この方法は本当においしいのです。

前述した通り、最新のAI技術は、大量のメモリやディスク、そして超高速のCPUやGPUなどが必要です。

しかし、そういうものが存在しなかった時代(パソコンすら存在しなかったころ)の「最適化アルゴリズム」は、シンプルの上に、美しく、短いコードでバグが入り込む可能性も小さく、何より、少ない繰り返し数で、解答に至ることができて——私は、大好きなのです。

はしごは“2本”、かけておこう

それでは、今回のコラムの内容をまとめてみたいと思います。

【1】今回は、「最適化アルゴリズム」について説明しました。私個人としては、先人たちが築き上げてきた「至宝の最適化アルゴリズム」を、軽々しく「AI技術」など呼びたくはないのですが、まあ、この連載の都合上、「AI技術」と見なすことにしました。

【2】今回の第3次AIブームにおきましては、強化学習や深層学習が、そのブームをけん引していますが、その理由として、「何も教えなくても、放っておけば、勝手に学習していく」という「神秘性」が人々を魅了している、という江端仮説を紹介しました。

【3】しかし、強化学習や深層学習は、その「神秘性」と引き換えに、膨大な「リソースコスト(メモリ、CPUなど)」と「非効率(学習時間)」を引き受けなければならないことと、これに対して「最適化アルゴリズム」は、そのような「神秘性」などカケラもなく、むしろ解決する問題にベッタリと寄り沿って、その問題を解くための「最適化」を極めるアルゴリズムであることを説明しました。

【4】この最適化アルゴリズムの事例として「ナップサック問題」「定期試験前の一夜漬けスケジュール問題」の他、雪山遭難シーンを想定した「最急降下法」の考え方を紹介しました。

【5】最適化アルゴリズムが、「ローカルミニマム」という問題から運命的に逃げられないこと、そしてその問題を軽減するために、各種の手法(シミュレーテッドアニーリングなど)が考案され続けてきたというお話をしました。

【6】最後に、水源発見問題(最適な井戸掘り場所を特定する問題)を例として、滑降シンプレックス法という、最適解探索手法について説明しました。



私、入社直後のころ、遺伝的アルゴリズム(GA)の適用事例として、図形パターンマッチングの研究をしていました。

私は、最新のAI技術は、過去の技術を軽く凌駕(りょうが)する効果を持っていると、盲信的に信じていました ―― だって、最新技術なんだから、最高の効果を発揮するはずだ、と。

その研究を学会発表するために、他方式との比較が必要になって、ちょっと試しに、私が生まれる前に考案された、超古典の「最適化アルゴリズム」のプログラムを試作して、その問題に適用してみました。

そして、そのプログラムは ―― 今となっては、はっきりと覚えていませんが ―― 私の考案した方式の1000分の1以下の時間で、計算を完了しました。

私のやっていた方式は、超古典的な最適化アルゴリズムの前に、言い訳の余地がないほど完全に負けたのです ―― 恥ずかしい。死にたい。どのツラ下げて、学会発表なんぞをしに行けば良いのか ―― と、本当に途方に暮れました。

私は、その時から、その研究への興味を完全に失いました。1000倍以上の差をつけられて完敗した私に、立ち上がる気力など1mmも残っていませんでした。

もちろん、この話は「GAが悪いアルゴリズム」という話ではありません。

私は、『課題を解決するためには、まず、その課題に適した手法(アルゴリズム)を検討しなければならない』という、研究員としての初歩の初歩を、字面(じずら)として知ってはいても、魂のレベルでは理解していなかったのです。

そして、この事件は、私に1つの教訓を与えました。「はやりモノは、正しいアプローチを見えにくくする」ということです。「はやりモノ」は、研究していて楽しいし、他人に自慢しやすいし、何より提案が通りやすく、予算が付きやすい。

本当に、今や「AI」と言うだけで、どの会社(の社長や幹部)も簡単に開発予算を付けています——毎日、新聞とかニュースとか見ながら「本当に、みんな、ちょろいなあ」と思っています(私、サラリーマンエンジニアですから、これ以上の発言は差し控えます)。

まあ、会社の社長や幹部なんぞはどうでもいいです。私が心配しているのは、現在のAIブームに載せられて、上の人間から、うまいことおだてられて、働かされている(かもしれない)若い研究員の皆さんです。

若手研究員の皆さん。本当に、世の中のブームの動きを良く見ながら仕事してくださいね。

はしごは簡単に外されます。

ですから、はしごというのは、常に自分で、2本以上掛けておくものなのです。

⇒「Over the AI ——AIの向こう側に」[⇒連載バックナンバー](#)



Profile

江端智一(えばたともち)

日本の大手総合電機メーカーの主任研究員。1991年に入社。「サンマとサバ」を2種類のセ

センサーだけで判別するという電子レンジの食品自動判別アルゴリズムの発明を皮切りに、エンジン制御からネットワーク監視、無線ネットワーク、屋内GPS、鉄道システムまで幅広い分野の研究開発に携わる。

意外な視点から繰り出される特許発明には定評が高く、特許権に関して強いこだわりを持つ。特に熾烈(しれつ)を極めた海外特許庁との戦いにおいて、審査官を交代させるまで戦い抜いて特許査定を奪取した話は、今なお伝説として「本人」が語り継いでいる。共同研究のために赴任した米国での2年間の生活では、会話の1割の単語だけを拾って残りの9割を推測し、相手の言っている内容を理解しないで会話を強行するという希少な能力を獲得し、凱旋帰国。

私生活においては、辛辣(しんらつ)な切り口で語られるエッセイをWebサイト「[こぼれネット](#)」で発表し続け、カルト的なファンから圧倒的な支持を得ている。また週末には、LANを敷設するために自宅の庭に穴を掘り、侵入検知センサーを設置し、24時間体制のホームセキュリティシステムを構築することを趣味としている。このシステムは現在も拡張を続けており、その完成形態は「本人」も知らない。

本連載の内容は、個人の意見および見解であり、所属する組織を代表したものではありません。

関連記事



[外交する人工知能 ～ 理想的な国境を、超空間の中に作る](#)

今回取り上げる人工知能技術は、「サポートベクターマシン(SVM)」です。サポートベクターマシンがどんな技術なのかは、国境問題を使って考えると実に分かりやすくなります。そこで、「江端がお隣の半島に亡命した場合、“北”と“南”のどちらの国民になるのか」という想定の下、サポートベクターマシンを解説してみます。



[陰湿な人工知能 ～「ハズレ」の中から「マシな奴」を選ぶ](#)

「せっかく参加したけど、この合コンはハズレだ」——。いえいえ、結論を急がないでください。「イケてない奴」の中から「マシな奴」を選ぶという、大変興味深い人工知能技術があるのです。今回はその技術を、「グルメな彼氏を姉妹で奪い合う」という泥沼(?)シチュエーションを設定して解説しましょう。



[上司の帰宅は最強の「残業低減策」だ ～「働き方改革」に悩む現場から](#)

あなた(あなたの会社)は、「働き方改革」に本気で取り組んでいますか? 読者の皆さんの中には、「誰かの上司」という立場である方も極めて多いと思われます。そんな皆さんに伝えたい——。シミュレーションで分かった「残業を減らす最善策」、それは皆さんが今すぐ、とっとと帰宅することなのです。



[誰も知らない「生産性向上」の正体 ～“人間抜き”でも経済は成長?](#)

「働き方改革」に関連する言葉で、最もよく聞かれる、もしくは最も声高に叫ばれているものが「生産性の向上」ではないでしょうか。他国と比較し、「生産性」の低さを嘆かれる日本——。ですが、本当のところ、「生産性」とは一体何なのでしょう。



[ニッポンのお家芸“カメラ”にも押し寄せるスマホ用チップセットの波](#)

リコーの360度全天カメラ「THETA」を取り上げる。2017年9月15日に発売されたばかりの最新モデル「RICOH THETA V」と従来モデルを比較していくと、外観にはさほど違いがないにもかかわらず、内部には大きな変化が生じていたのだった——。



[微細化の先導役がPCからモバイルに交代——先端プロセスを使いこなすスマホメーカー](#)

10nm世代の微細プロセス採用プロセッサを搭載したスマートフォンが出そろってきた。これら最新スマートフォンの内部を観察すると、各スマートフォンメーカーが10nmプロセッサを使いこなすための工夫が垣間見えてきた。

Copyright© 2017 ITmedia, Inc. All Rights Reserved.

