PROJECT 9 - Critical Thoughts about Big Data Analysis

Completed on 02-Jul-2016 (57 days)

Introduction >

Motivation

I have been actively involved in numerical modelling and data analysis during the last quite a few years. Nevertheless, most of my work has been focused on models dealing with a very specific subset of problems where typical big-data concerns rarely represent an issue. To know more about my background, visit the next section.

Since some time ago, I have been witnessing (big-)data-related situations whose outputs might be easily improved in a relevant way. Curiously, big data is a very commercially-appealing reality, what is usually associated with high availability of resources.

There are two issues which can be considered the ultimate responsible of this project:

- Online information misusage. Virtually all the relevant websites systematically horde and, what is even worse, don't maximise huge amounts of information. A descriptive example of such behaviours is target advertisement, an activity involving relevant amounts of money and valuable information but rarely delivering high accuracy.
- Data-analysis work opportunities. I (i.e., Alvaro Carballo Garcia, a self-employed programmer using the commercial name Custom Solvers 2.0) am permanently looking for new appealing remote-work opportunities: online challenges, open R&D calls, collaborative environments and, in general, any alternative adequately-rewarding my work. Projects related to (big) data analysis/forecasting definitively belong to the compatible-with-my-skills group. Hence, I also have a personal interest in the way in which big data analysis is being managed.

Technical Background

I am a senior programmer and numerical modeller with relevant experience in data-intensive software developments. In fact, my professional career as a programmer did precisely begin within a data-modelling-intensive environment. To know more about my theoretical and practical background, visit the about page of this site.

Nevertheless, most of my data modelling expertise has been focused on the adequate understanding of small amounts of information, rather than on dealing with as big as required datasets. Modelling the underlying phenomenon by means of analysing small-but-descriptive

sets of data rarely involves dealing with the typical big-data concerns, as described in the corresponding section. Thus and purely speaking, I have a limited big-data forecasting experience.

During the last weeks, I have been participating in various big-data problems and open challenges. I have confirmed the expected differences between small-high-quality and big-random-quality data models and learned quite a lot from these short-but-intense episodes. In fact, I am including my impressions, learned-lessons and recommendations about how to face big-data problems in one of the last sections and the appendix of this project.

Data Modelling >

Overview

Numerical models are basically a way to extend human understanding to situations initially beyond our reach (e.g., too big/complex sources of information or automated decision-making). That's why all the data models, regardless of their configuration, pursue the same goal: reliably understanding certain reality, ideally as well as a person would do. Despite being intrinsically identical, not all the data-understanding problems can be faced in the same way, an idea which underlies this whole project.

Roughly speaking, any modelling process can be divided into the following constituent parts:

- Training data. The past meaningful information used by the model to draw its predictions. The human-understanding equivalence is straightforward: all the information being accounted by a person to understand any situation and decide accordingly.
- Model itself. The set of algorithms in charge of learning (i.e., adequately understanding all the training information) and predicting (i.e., outputting the most likely results for the given inputs). This part emulates all the human learning, understanding and deciding capabilities.
- Resulting predictions. The conclusions delivered by the model for certain set of inputs. Eventually, the default model predictions might be corrected or complemented in order to ensure the highest accuracy; for example, by relying on an auto-learning subsystem. This part emulates the final outputs of a person understanding process (e.g., decision, guess, supposition, etc.), seen as a complex reality which might also involve interactions with other individuals.

Logically, the input conditions and the expectations have a major impact on the development of a model. By continuing with the human-understanding analogy, not everyone can analyse certain situations under certain conditions; for example: abstract impressions against detailed answers of an expert in the given field. The effect of the training information quality seems also quite evident (e.g., the worse the information, the more insightful has to be the person to adequately understand). On the other hand, the size of the training information might seem somehow secondary on this front, but it is certainly not.

As explained in the corresponding section, most of my numerical modelling expertise is focused on accounting for small-and-high-quality training information. One of the goals of this project is to share my impressions about the transition from such a background to big-data conditions. A second goal is to critically analyse the big-data aspects which might better be approached differently.

Big Data Peculiarities

As already explained, I have recently been working on various big-data-related developments (the appendix of this project includes my detailed impressions about one of them). These experiences have helped me gain relevant insights into big data forecasting, as opposed to what is associated with my more-restricted-model background.

The most relevant differences which I observed when facing the aforementioned big-data problems are summarised in the following points:

- Building comprehensive models (i.e., ones adequately accounting for virtually any sub-situation) is very difficult; in most of the cases, such a proceeding isn't even recommendable. The next point helps to understand this issue better; more specifically: the big-data expectations and/or assessing methodologies tend to favour not-so-bad-for-the-most outputs what penalises slightly-mispredicting-more-insightful approaches.
- Generic assessing methodologies. A descriptive example to illustrate this point: by taking an average-based methodology and assuming that the modelled behaviour is defined by (input=>output) 1=>2, 2=>3 and 3=>1, predictions of the form 1=>2, 2=>2 and 3=>2 would be assumed perfect. Such a proceeding would provoke a relevant penalisation for high-accuracy-prone attempts: in the very unlikely scenario of delivering an actually-perfect answer, it would get the same score than the aforementioned simplistic average-value result; in any other case, it would be worse independently upon its real understanding of the underlying behaviour.
- As a consequence of the two previous points, getting adapted to the peculiarities of this format seems an unavoidable requirement. Even a priori easy and intuitive ideas (e.g., keeping it as simple as possible) cannot be immediately applied, mainly in case of coming from a different background. The big-data character (i.e., huge training data sets, together with the conditions and expectations usually associated with these problems) has certainly a big influence on the way in which the given model is being developed.

Expected Evolution

Data analysis/forecasting is already being systematically used in many different contexts. Some typical examples:

- Everyday predictions (e.g., meteorological, financial/economical, polling, etc.).
- Customer-information maximisation, mainly for online businesses (e.g., target advertisement).

- Data-based scientific and R&D activity (e.g., predictive models existing in virtually any scientific field).

We are heading towards a (big) data driven society; adequately understanding data and making reliable decisions on account of it will become one of the basic cornerstones for virtually any technological advancement. Some examples of near-future applications of (big) data modelling:

- Any kind of automated, AI-based system.
- Businesses and organisations of any type systematically maximising the huge amounts of valuable information which they are currently ignoring.

In summary, the already-very-relevant impact of (big) data modelling is expected to keep growing, together with its reliability and accuracy expectations.

Critic >

Data Hording

An increasingly big number of businesses are realising about the numerous benefits of adequately understanding their clients' data. Additionally, internet and the associated huge amount of valuable information have played an important role in the wide adoption of data-analysis techniques. In fact, so much interest and availability of (free and easy-in-appearance) resources have provoked situations where data models are heavily misused. This section deals precisely with one of the main consequences of such a reality.

With data hording, I refer to a widely-spread-among-online-businesses attitude involving the collection of as much information as possible without properly analysing it. Such a (mis)proceeding has been somehow supported by the irruption of numerous big-data tools, commonly misunderstood as easy ways for people from any background to intuitively get worthy conclusions.

The aforementioned misconception has a negative impact on different fronts:

- The underlying assumptions about data analysis (i.e., it is easy, anyone can do it, generally-applicable and absolute answers can be expected, etc.) provoke negligent behaviours and the information to not be properly maximised. The most likely consequence of this point is the allocation of disproportionately-restricted resources; for example: unexperienced analysts, too limited budget/time constraints or unrealistic expectations.
- The more information, the more difficult to create a reliable model. In fact, notable increases in the amount of information being accounted usually provoke (even beyond-acceptable) increases of noise, what makes very difficult to create a model. Thus, more data is only better if the following two conditions are met: the quality of the additional information is high enough (or, at least, the noise increase is kept under control); and the given model is properly adapted (i.e., tuned, extended or even re-built)

to account for all the additional information. Logically, this level of care is incompatible with the described essentially careless behaviours.

Accuracy Unconcerned

An immediate consequence of the aforementioned data hording is that the adequate understanding of the collected information is converted into a somehow secondary concern. Note that an accurate-under-restricted-conditions model has nothing to do with properly defining the given situation. For example: only adding 2 and 3 vs. a fully-functional addition system; the first more restricted alternative might be acceptable in certain cases and only when the associated limitations are fully understood.

Despite the ideas in the previous paragraph, the fact of accounting for more or less information shouldn't theoretically affect the model accuracy. On the other hand and in most of the situations, this data-size-indifferent approach would imply a beyond-acceptable increase of complexity and, consequently, its practical impossibility. There is even a third alternative where all the additional not-properly-accounted information would be plainly ignored. This last implementation is much easier, but it still requires some additional work.

Another issue to bear in mind is that the information quality cannot be understood in an absolute way. The basic features which have to be present in any high-quality data source (e.g., correctness, completeness, descriptiveness, etc.) might be seen as mere prerequisites; on top of that, each data set has to meet certain specific quality targets. For example: when analysing the health of a person, the amount and quality of information related to physical features is very relevant, certainly not the case when analysing this person's taxes. Thus, a too careless or generic data-collecting behaviour has also a notable impact on the accuracy of the output conclusions, regardless of its size or the used data-analysing technique.

An adequate understanding of the corresponding context has also a major influence on accuracy. At a first sight, this issue might seem evident although the peculiarities of the automated understanding processes make easier to overlook it. Even under highly restricted conditions, an automated context understanding can be very difficult or impossible; even worse: the model might be unadvertedly ignoring such a relevant aspect.

Other issue to bear in mind is the common misconception that data models can be seen as absolute-answer deliverers working forever with a minimal support. Even the most complex and comprehensive model is very short-sighted and unadaptable in comparison with what human understanding can deliver. For example: a person, after having understood a given phenomenon, can output reliable conclusions on account of such knowledge for a wide variety of equivalent situations (logically, within the limits defined by the phenomenon and the person understanding capabilities). On the other hand, data models are always meant to be used under very restricted conditions and have a limited adaptability.

In summary, building an approach accurately analysing certain (big) data set in an automated way is a quite difficult task. A big number of different aspects have to be taken into account, what isn't a primary concern for many data-analysis unknowledgeable users. Furthermore,

adequately assessing the suitability of certain model to describe a given phenomenon is difficult and misinterpretation-prone.

Shortsighted Goals

Many business opportunities have started to grow around (big) data analysis; a reality which has attracted quite a few not-too-skilled-on-this-front people. Usually, they have a blind trust in easily getting immediate benefits and disproportionate or plainly delusional expectations. Such attitudes tend to be near funding/decision-making spheres, what converts their unreliable opinions into actually influential trends.

Big data challenges are an excellent place to get an accurate idea about this kind of attitudes; actually, the appendix of this project includes a detailed description of my participation in one of these challenges. Roughly speaking, complex (meaning interesting in this context) problems are proposed to a heterogeneous group of skilled, competitive and motivated online data modellers. That's why and regardless any other factor, these contests definitively provide a good reference about what some big-data-concerned(-but-not-necessarily-knowledgeable) companies consider difficult, relevant and even the future.

After having participated in a few of these challenges and analysed a relevant number of additional ones, my ideas about the typical challenge-proposer expectations (at least, the unknowledgeable sub-type) are very clear. I see two main problems here:

- Not-saying-much & easily-manipulatable assessing methodologies. In most of these contests the goals and the way in which solutions are assessed tend to be very simplistic and adapted to specific methodologies (i.e., problem expected to be solved in certain way and defined with this fact in mind). Some people might argue that this is required on account of all what a competition entails. In my opinion, this issue is exclusively provoked by not having properly analysed the problem and the goal; a new representation of the quick-easy-results-and-not-knowing-but-deciding attitudes which underlie this whole critic. Most of these challenges expect very specific answers to highly-restricted problems, but rarely output the ideal good insights into certain set of problems about which the given challenge should only be a mere descriptive sample.
- Plainly useless goals. I have seen quite a few cases where the pursued goal was plainly useless for the proposer. Example: creating a model to recognise to which road, out of 5, certain stretch belongs. This is a clearly overfitting-prone problem whose conclusions will never have a general applicability (i.e., being able to recognise any road from a given stretch).

In summary, wrongly-applied data-analysis/maths can prove virtually anything, what is the same than proving nothing. Additionally to building a proper model, the right questions have to be asked and the delivered outputs have to be adequately understood.

Conclusions >

Big Data Modelling

As previously explained, most of my numerical modelling expertise has been focused on specialised models dealing with well-delimited situations and relatively small datasets. Sharing my impressions about the differences between such an experience and standard big-data conditions is precisely one of the main goals of this project. Below these lines, I am including some generic ideas about (the referred transition to) big data modelling by extending what I wrote in the Big Data Peculiarities section. Additionally, the appendix of this project contains a detailed analysis of certain big-data challenge where I participated.

Main ideas to bear in mind when facing the development of big-data models:

- Relying on complex and comprehensive approaches since the first moment is certainly a bad proceeding. On the other hand, models based upon the accumulation of simple implementations affecting a relevant number of cases are likely to deliver a good performance.
- There is a huge amount of free resources whose utilisation is almost a must in a big number of cases. Nevertheless, these free resources are usually a two-edged sword: very useful, but easily misunderstandable. It is not just the false certainty which unknowledgeable (or even not-that-experienced in the big-data peculiarities) people might get, but also their usual unfriendly character; what, on the other hand, is quite logical on account of their complexity and poorly-supported-open-source essence.
- The big-data essence should never be forgotten. This point seems intuitively evident and this is precisely what it tries to prevent: intuitively evident but wrong actions. The most distinctive feature of big-data modelling is dealing with huge amounts of information, usually notably beyond our intuitive grasp; this issue can easily be missed when dealing with one of these models. That's why the following to-be-used-no-matter-what rule of thumb: each single intermediate action should always be automated by accounting for the highest number of cases; no intuition, suppositions, my-experience-tells-me-whatever or similar.
- Algorithm optimisation and/or powerful hardware availability have always to be seen as top priorities.
- Overfitting is particularly difficult to be detected and corrected. Nice-looking-but-really-saying-nothing results represent the most common variant of this problem. As discussed in the next section, the notably high number of unknowledgeable attitudes around big data makes this problem even more relevant.
- Coming up with methodologies adequately assessing the performance of these models is also quite difficulty. The aforementioned unknowledgeableness has a notable impact here too.

In a nutshell, there are two main issues to bear in mind when facing most of big-data problems. Firstly, the easily-forgettable unintuitive essence of most of what is related to so huge amounts

of data. Secondly, the cluelessness commonly associated with a relevant proportion of big data situations.

Accuracy Redefinition

Accuracy is a very wide concept with many different implications, much more when dealing with the so complex (big) data reality. That's why the critic of this part was, purely speaking, divided into two different sections: Accuracy Unconcerned and Shortsighted Goals. All the conclusions of this two-part analysis will be included in the current section.

The basic idea is that the (big-data) model accuracy needs to be redefined because of having virtually lost all its meaning. Such a generic statement can be better understood as opposed to the following common misconceptions:

- Big-data modelling is easy and anyone can get worthy results. Numerical modelling is a complex subfield where having a solid heterogeneous background (e.g., mathematics, programming, specific model-development expertise, etc.) represents an almost unreplaceable minimum requirement. Additionally and even under ideal conditions, the generated outputs are just probably-correct guesses. There is no intuitive, easy-for-everyone way to quickly build reliable models accounting for random situations.
- Generally-applicable absolute truths are possible or a properly-built model can work forever. Although mathematics (unlikely science, understood as causal phenomena descriptor) can deliver absolute always-working truths, numerical models define variable realities where such an eventuality isn't possible. Additionally, models rarely possess a truly deep understanding of the given reality.
- The more data, the better the model. Unlikely what happens with all the previous misconceptions, this statement might hold under very specific conditions (which, on the other hand, are rarely present). No good model can be built over bad-quality data; the higher the amount of data, the more difficult it to be high quality. Note that data quality improvements involve a set of imprecisely complex actions; that is: ensuring the (high) quality of certain dataset represents a notable increase of uncertainty.

Budget Redefinition

The immediate consequence of the previous section ideas is that (big-)data modelling isn't cheap; at least, not reliable-enough data modelling.

Main issues to bear in mind on the budget front:

- The more effort is put in the development of the model, the better. Top quality doesn't just imply building an accurate algorithm, but also: making it as scalable and adaptable as possible, collecting (filtering, correcting, grouping, etc.) the best information, developing all the required complementary applications (e.g., model assessment or intermediate actions automation), writing descriptive documentation about each part, etc.

- Arbitrarily constrained resources have a negative impact on the model. Examples: setting unrealistic targets or expecting to be lied (i.e., looking-nice-rather-than-reliable conclusions).
- Developing a data model rarely represents a one-time event. The most comprehensive and adaptable model needs to be tuned when its associated conditions change. Even under more or less stable conditions, models are formed by complex algorithms based upon multiple assumptions; correcting eventual errors, further extending their applicability or upgrading their functionalities are somehow common requirements. Actually, numerical models usually reach their top performance as a result of an evolution, rather than right after having been created.

The main reason to build a (big-)data model is precisely coming up with a cost-efficient solution for a given problem. Saving money in the development of the tool precisely meant to help save money doesn't make too much sense; much less when arbitrary restrictions might provoke a (perhaps uncorrectable or, even worse, undetectable) reduction of its money-saving effects.

Appendix >

Introduction

This project is the main output of my recent intention of getting insights into the differences between my numerical modelling expertise and big data conditions. I took part in various open challenges, although only spent a relevant amount of time and effort in the one described below these lines.

In this appendix, I am including my impressions about my participation in Kaggle's Expedia Hotel Recommendations. Most of the information associated with this challenge isn't public, that's why I may only share certain bits (e.g., data description).

Note that, since the very first moment, I took this challenge as the ideal benchmark to understand the aforementioned differences; also to eventually build a reliable set of applications, even to come up with a whole proceeding, helping me to quickly and efficiently face future big-data problems. It seemed that focusing on the test dataset (i.e., making many submissions) was the best way to accomplish such a goal. That's why my numerous submissions in this challenge, what shouldn't occur under normal conditions.

Model Evolution

The main algorithm didn't change much since the start. It consisted in sequentially analysing the likelihood of (combinations of) variables predicting the given output (*hotel_cluster*), as defined below these lines.

- All the training cases where the given variable had the same value were grouped together. Most of them were associated with multiple *hotel_cluster*, that's why the total number of occurrences was also stored.
- At the beginning, this algorithm was run (and validated via submissions) with the relevant-in-appearance variables. Gradually, only considering single variables was converted into mostly considering combinations of different variables.
- The coordination of multiple (combinations of) variables was done by taking as many predictions as possible (up to the maximum value of 5) from the current combination before moving to the next one. The order in which the different combinations were analysed was problematic since the start, but mostly relied on a mixture of proven-good-performance, number of variables (the more variables in the combination, the better) and average likelihood of all the given *hotel_cluster* to be right (the lower the number of different *hotel_cluster* associated with the combination, the better).

This basic structure went through the following relevant modifications:

- Gradual increase of the number of combinations and variables per combination until reaching a point where the model stopped being easily-scalable.
- Better way to ease the inclusion of further combinations and variables, by bearing in mind my future-usage expectations (i.e., making it as generic and adaptable as possible).
- After the first hitting-memory-limits problems, new approach able to deal with any number of variables. It was a multi-step/-application methodology, where intermediate information was generated and used at different points.
- Although accounting for as many combinations as required was already possible, the number of potential configurations was too high (15! to 25!) and the process too slow. That's why finding the best combinations became the main concern.
- Searching for the best combinations was proven more tiring and unrewarding than planned; additionally, there wasn't much available time already (with more time, this proceeding might have output good results). One last try at keeping it as simple as possible by ignoring all the stored combinations and restarting the process again.

There wasn't enough time to fully maximise my approach and giving up seemed the best option. Before that, I took a look at the public codes what, as explained in the next section, helped me realise about various issues.

Final Model

After confirming that there wasn't enough time to maximise my approach, I took a quick peek at some of the public (Python) codes. Note that this was one of my first Kaggle challenges and I wasn't too sure about the exact meaning of these public contributions. Apparently, they were created by Kaggle's staff to provide some help to solvers.

This public code (note that all the ones I saw were slight modifications of the same algorithm) was performing notably better than my best attempt so far. Its basic structure was quite similar to the one of my approach; it was even accounting for the best combinations of variables as per

my tests. On the other hand, it also had the following important differences with respect to my model:

- It was accounting for the date/time variables in a quite complex way. During my tests, I did some (much more simplistic) attempts to bring this information into picture, but none of them provoked a relevant improvement.
- It was filtering the cases on account of the *is_booking* variable. On one hand, the problem description clearly stated that this variable was considered in all the test cases; but on the other hand, all my tests and submissions on this front came to the conclusion that accounting for it wasn't beneficial. In fact, my last-moment tests, as described in the final paragraph of this section, seemed to support such conclusions.
- It was giving some relevance to certain variable (*distance*) which my approach was ignoring. As per most of my tests, this variable had a quite low influence.
- There were various looking-quite-arbitrary filters. Not sure about its exact motivation; perhaps a mistake or perhaps a new quite-complex-but-performing-surprisingly-well bit.

There wasn't much time remaining, this was my first contact ever with Python (and its spacing peculiarities) and that specific code was quite memory inefficient (at least, inefficient enough to not run on my computer). Despite all these problems, I was able to put together a reasonably-good benchmark which seemed to work perfectly; at least, until right the last moment (2 hours before the deadline), when the memory problems came back. My tests indicated that removing the reliance on *is_booking* would have allowed this model to score notably higher; but unfortunately, the last-moment memory problem didn't let me confirm such an assumption. It was definitively a curious end for a curious challenge.

Learned Lessons

Undoubtedly, this has been a very valuable experience from which I have learned various worthy lessons. In fact, the original expectations of adapting my specific modelling expertise to big-data conditions have been widely overpassed.

Kaggle is quite important within the (big-)data-challenge community and also getting relevant insights into the best proceeding to deal with its peculiarities. Without entering into too much detail about the exact essence of their public codes (the ones created by its staff and, eventually, improved by some contestants), I am certain about something: they help to quickly and efficiently get into the problem. Note that there is a very important difference between creating (mainly from scratch) and improving; even completely redoing a reasonably-good approach is notably easier than working without any guidance.

There is also a worthy conclusion from my unplanned first contact with Python: (reconfirming that) getting adapted to a different programming language isn't too difficult, mainly for an experienced programmer like me. Note that, despite my relevant programming expertise and having worked with many different frameworks, I never used Python and its defining spacing-matters feature (a non-issue in all the languages I have ever used). I started running/analysing the given script almost immediately; but after realising about its memory problems, my first

thought was moving the algorithm to C#. I changed my mind back after realising that the code was more intrincate than what it seemed at first sight. That is: the day after having used the language for the first time (and just hours before the final deadline), I started rewriting a not-that-simple algorithm such that it could be run in my (just!) 12 GB RAM computer. Unfortunately, this story didn't end well and my improved version wasn't able to deal with a notably bigger amount of data in one of my last tests. Nevertheless, this problem wasn't exactly provoked by my limited Python knowledge because of referring to an unexpected-and-difficult-to-be-fixed issue.

Regarding facing big-data problems, I learned the following:

- Keeping things as simple as possible on all the fronts. Note that I applied as-simple-as-possible ideas since the very first moment. In any case, my intentions were inadvertently changing, mostly via wanting to build an as comprehensive as possible approach able to deal with as many variables as possible.
  The Kaggle's public code helped me see that a low number of (combinations of) variables might also output a good performance. Logically, this fact doesn't say anything about what an approach accounting for many more variables (the target of my model) can do. On the other hand, the hardware/time constraints can certainly convert such a theoretical no-problem into a practical no-way.
  Hence, the keep-it-simple ideas have to be applied as systematically as possible when dealing with big data problems. What doesn't just refer to the complexity of the different methodologies forming the model, but also to the number of variables and to the best way to face the whole situation (i.e., properly analysing a few rather than accounting for many).
- At least in open challenges or similar situations where the proposer's background is unclear, the quality of the input information shouldn't be assumed perfect. The quality of the inputs (understood in its widest sense as training data, description of the problem, expected problems, etc.) has a tremendous impact on the reliability of the resulting model; even the slightest issue here should be immediately fixed.
  In a challenge where the aforementioned fixing isn't possible, such a reality should be accepted and the model developed accordingly.
- The assessing methodologies used when dealing with objective-correctness-prone scenarios and contest-like conditions are also very different. The model should definitively be assessed such that its performance is properly understood (i.e., likelihood of its predictions to be right, by paying special attention to avoiding correct-in-appearance misinterpretations). Further issues, like scalability and adaptability, should ideally be also brought into picture while determining the adequacy of a given numerical model.
  Under contest-like conditions, the previous paragraph has no real applicability. All what matters is scoring as high as possible by applying the corresponding assessing methodology. In these cases, any other consideration (including developing an approach performing objectively better) would provoke unnecessary wastes of time.