Some Hints on the Teaching of Machine Learning to Industrial Practitioners

Christopher Kermorvant CK@A2IA.COM

Keywords: machine learning for the industry, technology maturity, condition for successful applications, software project

Abstract

Many machine learning algorithms are now leaving academic laboratories to be embedded in industrial applications. This implies that machine leaning students should have a more pragmatic knowledge about machine learning technology. In this article, I present three practical points about machine learning that should be more emphasized in courses: the maturity of machine learning techniques, the way to transform a machine learning algorithm into a successful application and some specificities of conducting a machine learning-based software project.

Machine learning (ML) techniques are more and more used in the industry in a large variety of application domains: information technology (search engines, speech or document recognizers), medicine (image processing, computer-aided diagnostic systems, bioinformatics) or finance (stock market prediction, actuarial science). This implies that machine learning algorithms are being transfered from academic laboratories to private companies. But this transfer from laboratories to applications is not straightforward: it can be slow, it can be more or less advanced depending on the application, some technologies have great chance to be transfered within two years whereas for some other technologies the scope is at least ten years.

The state of maturity of machine learning technologies and applications is a knowledge that should be shared by the community and should be taught to students. Even though, to my knowledge, no publication such as a "state of machine learning technology" exists, the maturity of a technology can be evaluated by the experimental settings in which the technology is tested. In section 1, I describe a four level scale of maturity based the kind of data and problems on which the technology is successfully tested. I also describe a development curve designed in 2005 by an advisory firm to report the level of maturity of any technology and give a summary of the level of maturity of some machine learning technology estimated by this firm over the last four years,

One may think that the good time to transfer a machine learning technology from laboratories to the industry is when the error rate reaches zero or so on the task. But if it was the case, machine learning algorithms would still be in laboratories. The key to a successful machine learning based application is not a zero error rate (though it could help) but is to find a working point. A working point is a setting for which using the machine learning algorithm improves the whole process either in term of productivity or in term of cost. In section 2, I give an example of such a working point for a customer mail processing system. I show that the key to a successful application is not directly the error rate but the confidence measure associated to the prediction. To my mind, this point should be more emphasized both in publications and in machine learning lectures.

Once a software product has been designed using a machine learning algorithm, this product usually has to be delivered to customers through software projects. Projects for softwares based on ML algorithms have some specificities compared to both research projects in laboratories and traditional software projects. First, both the data and the problem on which the ML software will be trained and used can be new to a certain extend and should be considered with a very critical eye. In section 3, I sketch the list of critical points which should be considered. Second, machine learning practitioners have to interact with customers, software engineers, project managers, all kind of people who are not familiar with fundamental machine learning concepts and hypothesis. This means that hypothesis which are usually implicit in the machine learning community must be made explicit in communication outside the community. In the last section, I list some of the concepts and hypothesis that must be explained

in software projects in which a ML algorithm is embedded.

1. Evaluating the maturity of Machine Learning technology

1.1. The different kinds of experimental settings

For a large part, the activity of a machine learning practitioner consists in making experiments on data. Data are the starting point of all machine learning applications and are often seen as what define the problem to tackle. However, not all data are equivalent: some are synthetic, some are natural, some are generated in a very controlled environment, some are collected in the *real field*. Moreover, the nature of the data is not the only parameter in defining a machine learning experiment: on the same data, several problems can be defined. A machine learning task is therefore defined by both the data and the problem. One can categorize the different kinds of experimental settings as follows:

Synthetic data and synthetic problem : data are artificially generated from the definition of a synthetic problem; this experimental setting is used to validate an idea or an algorithm.
Example: points uniformly drawn in a multidimensional space used to test a density estimation or classification algorithm.

Real data but synthetic problem : data are real (collected form the real world) but the problem is simplified; this experimental setting is used to test an algorithm on real data but in a very controlled environment so that the interpretation of the results are made simplier.
Example: in the context of handwriting recognition, separated digit recognition on MNIST database[1].

Real data and realistic problem : data are real, the problem corresponds to a real final application but the data were collected using a simulation of the application.
Example: in the context of handwriting recognition, the handwritten letter recognition task in the RIMES project (Augustin et al., 2006). In this case, letters have been handwritten by volunteers according to a predefined scenario.

Real data and real problem : data are real and directly extracted form a real world application.
Example: in the context of handwriting recognition, the categorization of complete mails from customers, the mail images being extracted from the customer relation database of the company.

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

Note that this typology defines the different types of experimental settings without any order of superiority: all these experimental conditions are useful at a certain point of the research development but it is important to know the limitation of each setting.

Working on synthetic data should be the first step when testing a new algorithm or when learning how to work with an unfamiliar algorithm. But one can not be satisfied with experiments only on synthetic data.

Working on real data but with a simplified and controlled task is the most common setting in the machine learning academic community. This step is very useful to study the behavior of an algorithm or the effect of the parameters and to evaluate the performance of the algorithm on different tasks. But the performance of the algorithm should not be extrapolated blindly to a realistic problem and the difficulty of the realistic task should not be under-estimated. I develop this point in the next two sections.

In the context of speech recognition, a language modeling task is a task in which the language model is evaluated on its own (without a speech recognizer) on a corpus using a perplexity measure (Rabiner & Biing-Hwang, 1993). This task should be considered as falling in the category "real data but synthetic problem". The real problem is a complete speech recognition system and it is well known that improving the perplexity of the language model does not linearly improves the performance of the speech recognition system (Klakow & Peters, 2002). But testing a language mode in a "Real data and realistic problem" setting would imply to develop a complete and state-of-the art speech recognition system which is impossible for most research groups.

The complexity of designing a complete system is the reason why most machine learning experiments fall in the "Real data but synthetic problem" category. This fact can lead to some misconceptions regarding the status of machine learning applications. For example, the best recognition error rate on the MNIST handwritten digit database is 0.4% (Simard et al., 2003), which may lead to think that handwritten digit recognition is a solved problem. However, one must consider that this result is obtained in a "Real data but synthetic problem" setting, The error rate dramatically increase on a task that belongs to the "real data and real problem", for example when the task is recogniz-

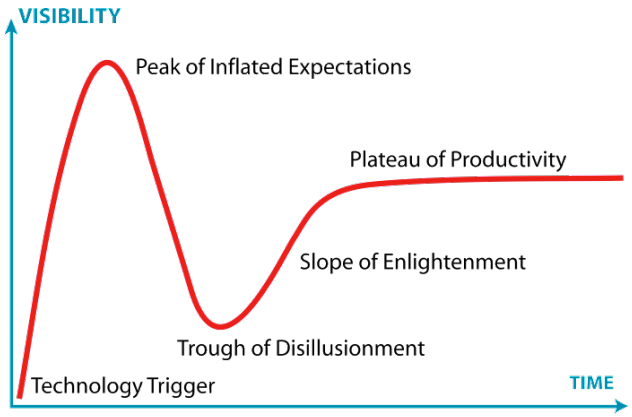

Figure 1. The Gartner's technology hype cycle.

ing french social security numbers (13 digits) on faxed health insurance forms.

The "Real data and realistic problem" setting is the academic setting which is the closest to the real application. However, one limitation of this setting can be its bias in data collection. For example, in the RIMES project (Augustin et al., 2006), a call to volunteers was made to collect handwritten letters based on predefined different scenarii. This call was made on the internet by several mailing list. A large majority of the volunteers were computer literate students between 20 and 30 years old with a university education level. It is easy to see that this population is very different from the category of people who usually write handwritten letters to company customer services (the real problem), who usually don't have a computer, belong to lower social categories or are older than 50. In this case, the socio-professional category bias in data collection makes this database fall into the "Real data and realistic problem" category. Besides, it is worth noting the very active role of the US Defense Advanced Research Projects Agency (DARPA) in promoting the comparison of machine learning techniques on "real data and realistic problem" through annual competitions. DARPA reports are often a good source of information when evaluating the maturity of a technology, if this technology is of any interest for the US defense.

The four categories of experimental settings presented in this section can be easily used to evaluate the maturity of machine learning algorithms for an application, from the infancy to the industrial maturity. Another source of information is described in the next section.

1.2. Gartner Inc.'s Hype Cycle

If machine learning algorithms are used in a large variety of applications, not all applications have reach the same level of maturity. If some of them are commonly used in the industry, for example OCR, some others are still in their infancy, like image search by content. Giving students an overview of machine learning application domains and their level of maturity would be very useful to help them to stand back and better evaluate the development of the field. Technological development evaluation scales like the Technology Readiness Level (Mankins, 1995) used by the US Army and other US agencies can be used. Gartner[2], an information and technology research and advisory firm, publishes annual reports on technology and estimates the level of maturity of many technologies. For each techology, they evaluate its position on what they call a technology hype curve. This curve, presented on Figure 1, defines 5 levels of maturity:

Technology Trigger : the first phase of the cycle is the "technology trigger" or breakthrough;

Peak of Inflated Expectations : publicity around the technology generates over-enthusiasm and unrealistic expectations. Some applications are successful but there are more failures than successes;

Trough of Disillusionment : the technology fails to meet expectations and become unfashionable;

Slope of Enlightenment : some people continue to experiment and understand the benefits and practical applications of the technology;

Plateau of Productivity : the benefits of the technology are widely demonstrated and accepted.

In their four last report, Gartner gave the following evaluation for some machine learning applications:

Technology Trigger: gesture recognition (2004), information extraction (2004), information retrieval and search (2004), networked collective intelligence (2005, 2006), text mining (2005), speech-to-speech translation (2006);

Peak of Inflated Expectations: Speech recognition for mobile devices (2006), social network analysis (2006), collective intelligence (2007), gesture recognition (2007);

Trough of Disillusionment: biometric user-identification (2005), social network analysis (2007);

[2]www.gartner.com

Slope of Enlightenment: automated text categorization (2004), handwriting recognition (2005);

Plateau of Productivity: speech recognition for telephony and call center (2005), text-to-speech (2005).

I think that researcher working on an application involving machine learning should have a relatively clear view of where on such a curve their application is located. Students should also be aware of this level of maturity in order to demonstrate a pragmatic knowledge of machine learning technology, which is required by the industry.

2. How to turn an algorithm into an industrial application ?

Application of machine learning algorithms are often seen as a replacement for humans. In the most known applications of machine learning algorithms like chess playing, web page indexing or spam filtering, the computer is indeed used to replace all of human action. But in most applications, human actions are still present in the process:

- the engineers at Google are permanently monitoring the search engine in order to improve it;
- in the call center applications, a backup to a human operator is planned when the speech recognizer fails to deal with the call;
- in an automatic bank check processing system, checks for which the recognizer is not confident enough are processed by a human agent.

In fact, in many machine learning applications, the machine learning algorithm is used to *partially* replace human agents in the process. The presence of humans in the automated process is due to the fact that machine learning algorithms are not perfect. I illustrate this point with an example in the next section.

Let us consider recognition applications. Despite major progresses over the last 30 years, the error rate of automatic speech recognition systems in a large vocabulary task is still over 10%. Going under the frontier of 10% error rate is considered to be a necessary condition to start using automatic speech recognition in applications (Juang & Rabiner, 2005). However, speech recognition for telephony and call center are now considered on the plateau of productivity by Gartner (Fenn & Linden, 2005) since 2005, which means that real-world benefits of the technology are demonstrated and accepted. How can a technology with an error rate as high as 10% be used in real-world application ?

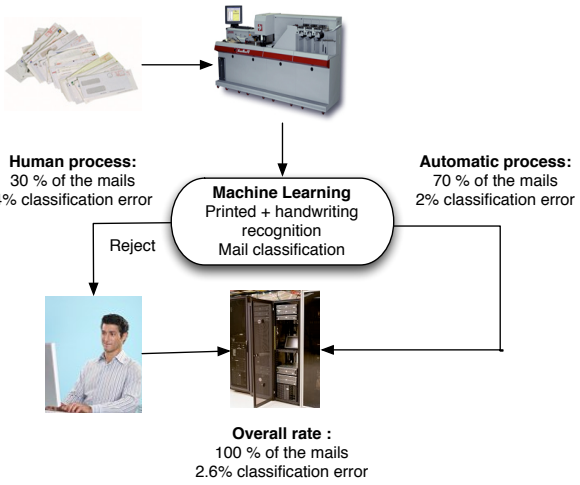

Figure 2. An automated system for customer mail processing.

The key to an industrial application of a technology is to find a *working point*. A working point is a setting in which the machine learning algorithm is applied to the task in the cases where its error rate is lower that the human error rate, the remaining cases being processed by human agents. The overall error rate of the combined system (machine and human) is then lower than the error rate of the purely human process.

An example of such a process is described on Figure 2. In this example, a machine learning algorithm is used to classify incoming mail from customer in a company. The error rate of the automatic classification algorithm is 15%, which is very high compared to the error rate of a human agent, estimated at 4%. However the machine learning algorithm can still be used in the classification process. In order to find a working point, a read/error curve can be drawn as shown on Figure 3. The read rate is the percent of documents automatically processed by the algorithm. When the read rate decreases, the documents for which the algorithm is less confident in its prediction are rejected. The error rate on the remaining documents usually decreases. As presented on Figure 3, if a 2% error rate is requested, 30% of the documents must be rejected. This point (2% error rate for 30% reading rate) is the working point. With this setting, 30% of the documents go to a manual classification and the overall classification error rate is 2.6%.

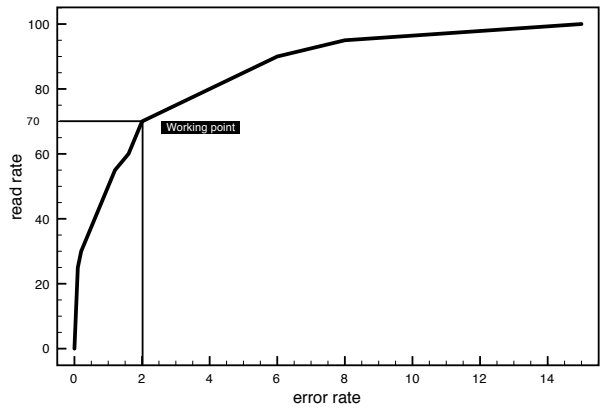

Figure 3. A read/error curve used to find a working point.

In order to find a working point, the key feature of the machine learning algorithm is its ability to provide a confidence measure in its predictions. However, such properties of the machine learning algorithms are seldom mentioned in articles from the the machine learning community. Reject properties and confidence measures are mainly studied in publications from specific domains like medicine (Hanley, 1998) or speech recognition (Jiang, 2005). Since these aspect of machine learning algorithms are crucial to applications, they should be more emphasized both in publications and courses to students.

3. Deployment of a machine learning-based software

In this section, I describe some of the specificities of the deployment of a software including a machine learning algorithm.

3.1. Working with real data

Researchers in the academy have rarely the occasion to work with real data and real problems. Even though working on real data and realistic problem can be very similar to real problems, the two settings differ from several points. I describe some of these differences in this section.

When working on real problems, both the data and the problem should be examined with a critical eye. It is well known by the data mining practitioners that an important step in the data mining process is the pre-processing of the data: cleaning and normalizing the data. When working a new project, no one has explored the data and made experiments: there is no way to compare to previous work. A systematic approach to the data and to the problem must be used:

1. Read the customer specification;
2. Understand and clearly state the problem;
3. Compute descriptive statistics on the data and compare to what the client has specified;
4. Evaluate the ground-truth if it is provided. What is the rate of labeling noise ? Are the labeling rules coherent ?
5. Evaluate a baseline performance on the task using a simple and well understood algorithm ;
6. Select a ML algorithm to take into account user's constraints: cost sensitive classification, unbalanced datasets ;
7. Perform both a quantitative error analysis (error rate, confidence intervals) and qualitative error analysis (going manually though a sample of the errors);

This list is obviously non exhaustive and more detailed procedures can be found in data-mining guides. But I think that machine learning students should be familiar with these kind of procedures.

3.2. Machine learning-based software project

Software projects which encompass a machine learning component have several specificities compared to classical software projects. These specificities are due to the probabilistic nature of machine learning algorithms. When facing customers who are not familiar with machine learning algorithms or with statistics, the ML practitioner must make explicit a number of concepts which are usually implicit or even unconscious.

One specificity of a ML-based software is its test setup. A classical software test plan includes a list of functional tests which are passed or not passed by the software. In the case of ML-based software, the test plan is not so simple. Let us consider a classification application. The customer gave in the specifications a target to reach: an error rate on a certain percent of documents. But it may be no trivial to know if the software meets the specifications, that is whether the error rate is below the target value and the read read rate above the target value or not.

First the error rate measure on the task is not a definitive value. Most of the time it is (and should be) a

value with a confidence interval, at a certain risk value. What is the risk value that the customer is ready to accept ? What is the confidence interval that the customer is ready to accept ? These questions must be made explicit, even if most of the time the customer is not able to give an answer.

Second, the test plan is generally conducted on data made provided by the customer. Several questions must be raised:

- How the data was collected by the customer ? Is the sampling random ?
- Is the sampling uniform ? Is the sampling representative of the task ?
- Is the sampling representative of the evolution of the data with time ?
- Is the test set different from the train set ?
- On which data the hyper-parameters of the algorithm have been tuned ? How will tune the hyper-parameters ?
- Are ground-truth values available ?
- What is the noise level on ground-truth values ?

Even if they seem trivial, all these points should be explicitly on the check list of the software test plan since the ML practitioner can rely neither on the software engineer nor on the customer to check them.

Finally, once the product is delivered, the company must provide support for its product. For machine learning algorithms, it means analyzing the behavior of the algorithm in new testing conditions: what happens if the running conditions change, if the distribution of the data changes, if the problem slightly changes ? Up to now, these problems have received little interest in the machine learning community but they are crucial for the success of machine learning applications in the duration.

4. Conclusion

In this article, I have developed three points related to machine learning applications that should be more emphasized in teaching machine learning. First an overview of the level of maturity of machine learning technologies should be shared by the community and the students. The level of maturity of a technology can be simply measure by the type of data and problem on which it has been shown to be successful. This level of maturity can be presented on a development curve as proposed by an advisory firm. Second, the conditions necessary to the transfer of a machine learning technology from laboratories to the industry should be more studied. The knowledge of these conditions will foster the spread of machine learning techniques in the industry. Finally, I sketched the specificities of conducting a machine learning based software project by comparison to a traditional software project. Even if these specificities are well know in the machine learning community, they should be made explicit in projects involving people from other background.

References

Augustin, E., Carre, M., Grosicki, E., Brodin, J.-M., Geoffrois, E., & Preteux, F. (2006). Rimes evaluation campaign for handwritten mail processing. *Proceedings 10th International Workshop on Frontiers in Handwriting Recognition* (pp. 231–235).

Fenn, J., & Linden, A. (2005). *Gartner's hype cycle special report for 2005* (Technical Report). Gartner.

Hanley, J. (1998). *Encyclopedia of biostatistics*, vol. 5, chapter Receiver Operating Characteristic (ROC) Curves, 3738–3745. John Wiley & Sons, Ltd.

Jiang, H. (2005). Confidence measures for speech recognition: A survey. *Speech Communication*, *45*, 455–470.

Juang, B. H., & Rabiner, L. R. (2005). *Elsevier encyclopedia of language and linguistics*, chapter Automatic Speech Recognition–A Brief History of the Technology. Elsevier. Second edition.

Klakow, D., & Peters, J. (2002). Testing the correlation of word error rate and perplexity. *Speech Communication*, *38*, 19–28.

Mankins, J. (1995). *Technology Readyness Levels* (Technical Report). Advanced Concepts Office, Office of Space Access and Technology, NASA.

Rabiner, L., & Biing-Hwang, J. (1993). *Fundamentals of speech recognition*. Prentice Hall.

Simard, P., Steinkraus, D., & Platt, J. (2003). Best practice for convolutional neural networks applied to visual document analysis. *International Conference on Document Analysis and Recognition* (pp. 958–962).