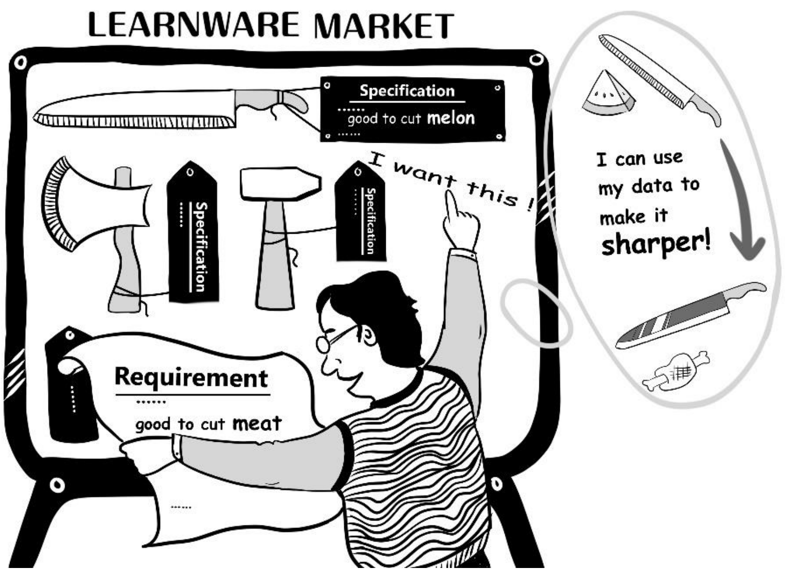
Learnware: On the Future of Machine Learning

Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China

Current machine learning techniques have achieved great success; however, there are many deficiencies. First, to train a strong model, a large amount of training examples are required, whereas collecting the data, particularly data with labels, is expensive or even difficult in many real tasks. Second, once a model has been trained, if environment changes, which often happens in real tasks, the model can hardly perform well or even become useless. Third, the trained models are usually black-boxes, whereas people usually want to know what have been learned by the models, particularly in real tasks where decision reliability is crucial and rigorous judgment by human beings are critical.

In addition to the above deficiencies, there are several relevant issues require attention. First, some data have to be shared in most current machine learning studies if one hopes to pass helpful information from one task to another. The data privacy or data proprietary, however, usually disable public data sharing. Thus, it is hard for people to build their learning tasks based on the results of other people. Second, machine learning is still a kind of magic: Even with sufficient training data, most end users, except machine learning experts, can hardly produce strong models.

Considering the above issues, here we propose ***learnware***. A learnware is a well-performed pre-trained machine learning model with a ***specification*** which explains the purpose and/or specialty of the model. The specification can be logic-based descriptions, and/or statistics that reveal the target to which the model aimed, and/or even a few simplified training samples that disclose the scenario for which the model was trained. The owner of a learnware can put it into a market, with little risk of data privacy leakage. As the comic illustrates, when a person is going to tackle a machine learning task, rather than building his model from scratch, he can do it in this way: Figure out his own **requirement**, and then browse/search the market, identify and take a good learnware whose specification matches his requirement. In some cases he can use the learnware directly, whereas in more cases he may need to use his own data to adapt/polish the learnware. Nevertheless, the whole process can be much less expensive and more efficient than building a model from scratch by himself.



For this purpose, a learnware should have at least three important properties: **Reusable**, **Evolvable**, and **Comprehensible**.

A learnware should be reusable, and otherwise it can hardly be useful for other users. In particular, the pre-trained model should be able to be enhanced or adapted, by its new user through a slight modification or refinement using information, such as a small amount of training data, from the task of the new user. This process may be subtle: On one hand, one needs to avoid important learned knowledge be washed out by the refining; on the other hand, the model should have sufficient flexibility to incorporate necessary modification desired by its new user. There are some machine learning studies that can be regarded as preliminary attempt for this purpose, e.g., model adaptation [1], transfer learning [2].

Evolvable means that the learnware should be able to get accustomed to environment change. If reusable is viewed as the learnware's ability of passive adaptation driven by the user, evolvable can be viewed as its ability of active adaptation: The learnware should be able to perceive the environment change and do the adaptation by itself. There are at least three reasons for this need. First, the learning task of the new user is usually somewhat different from the original task for which the learnware was constructed, because one can hardly expect a learning task exactly appear again. Second, the learnware specification and/or the user requirement can hardly be very accurate descriptions, and there may exist some gap that the learnware must be able to get through. Third, many real environments are non-stationary and changing in nature, e.g., data distribution may change [3], new classes may occur [4,5], features may change [6], etc. These issues have also been emphasized to be tackled on the way toward robust artificial intelligence [7].

A learnware should be comprehensible; that means, the learning models need to be transparent to some extent, at least enabling the writing of specification for the learnware. For example, one needs to know what kind of target the learnware was trained for, how good the learnware performed, what specific problem structure the learnware can be applied, etc. On one hand, black-box models need to be made white; for this there are many efforts, e.g., trying to improve the comprehensibility of black-box models [8], or even produce accurate and comprehensible models [9]. On the other hand, one needs to be able to write the specification that well describes a pre-trained machine learning model; unfortunately there is little study about this. Inspiration may be learned from the field of software engineering, where specification has a long history of study and application.

Note that in most cases, the end user may be unable to identify a single learnware which exactly matches his requirement; instead, he may find multiple learnwares each meets a part. In such cases, ensemble methods [10] that combine multiple models to use may offer some solutions, just like the description of *reusable ensemble* [10, pp.184] where reusable components are searched and put together, and only functional components that could not be found need to be constructed.

If learnwares come true, strong machine learning models can be achieved even for tasks with small data, because the models are built upon well-performed learnwares, and only a small amount of data are needed for adaptation or refinement. Data privacy will become a less serious issue because publicizing learnwares does not need to share the data. More importantly, it will enable common end users to attain tricky learning results that can only be attained by machine learning experts nowadays. Needless to say, a promising learnware industry will be open.

Acknowledgements:
The author wish to thank Wang-Zhou Dai for converting a hand drawing sketch to comic, and thank LAMDA group members for helpful discussion.

References:

[1] N. Li, I. W. Tsang, and Z.-H. Zhou. Efficient optimization of performance measures by classifier adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(6): 1370-1382.

[2] S. J. Pan and Q. Yang. A survey of transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[3] M. Sugiyama and M. Kawanabe. Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation. Cambridge, MA: MIT Press, 2012.

[4] Q. Da, Y. Yu, and Z.-H. Zhou. Learning with augmented class by exploiting unlabeled data. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14), Quebec City, Canada, 2014, pp.1760-1766.

[5] X. Mu, K. M. Ting, and Z.-H. Zhou. Classification under streaming emerging new classes: A solution using completely random trees. CORR abs/1605.09131, 2016.

[6] C. Hou and Z.-H. Zhou. One-pass learning with incremental and decremental features. CORR abs/1605.09082, 2016.

[7] T. G. Dietterich. Towards robust artificial intelligence. AAAI Presidential Address at the 30th AAAI Conference on Artificial Intelligence (AAAI’16), Phoenix, AZ.

[8] Z.-H. Zhou, Y. Jiang, and S.-F. Chen. Extracting symbolic rules from trained neural network ensembles. AI Communications, 2003, 16(1): 3-15.

[9] Z.-H. Zhou and Y. Jiang. NeC4.5: Neural ensemble based C4.5. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(6): 770-773.

[10] Z.-H. Zhou. Ensemble Methods: Foundations and Algorithms. Boca Raton, FL: CRC Press, 2012.