A Feasibility Study of CONCEPT DRIFT in Empirical Software Engineering

Md Alamgir Kabir
Supervisor: Dr. Jacky Keung

December 11, 2018

Department of Computer Science

Topic

1. Concept Drift
2. Types of Concept Drift
3. Detecting Concept Drift
4. Practical Problem in Classification
5. Implications in Software Engineering Research

Concept Drift

Concept Drift

- **Streaming data**: Incoming data of heterogeneous sources
- Behavior of these streaming data is changing over time
- **Concept Drift** [1] : The changes in the distribution or concept over time
- A time changing probability distribution in high speed data streams
- Concepts are often not stable but change with time such as Weather Prediction and Customers' Preference

[1] Ditzler, G. (2015). Learning in Nonstationary Environments: A Survey, (November). `https://doi.org/10.1109/MCI.2015.2471196`

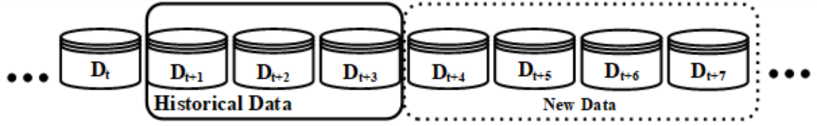

Figure 1: Two windows with different data size. In *concept drift*, the new data will change with different window size while historical data will be fixed. According to J. Lu et al., the new data is determined by the user.

Concept Drift

According to Lu [2], the problem can be framed as follows.

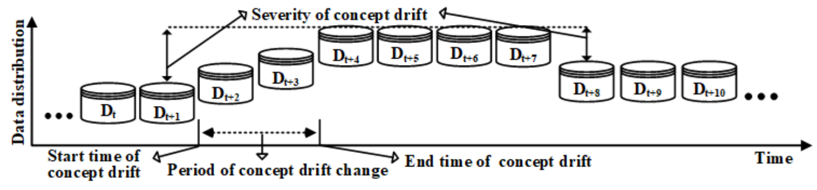

Figure 2: Concept drift occurrence time and its severity. In this example, the starting point of the drift in the time, *t+1* and end time of the drift is *t+3*. It is marked the severity of concept drift.

[2]Lu, N., Zhang, G., & Lu, J. (2014). Concept drift detection via competence models. Artificial Intelligence, 209(1), 11–28. `https://doi.org/10.1016/j.artint.2014.01.001`

Concept Drift

According to Gama et al. [3], the concept drift can be explained with the time t_0 and t_1 as given below:

$$D_{t_0}(A, b) \neq D_{t_1}(A, b) \tag{1}$$

- D_{t_0} explains the distribution at the time t_0 between the input data X in its corresponding label y.
- Again, D_{t_1} prevails the distribution at the time t_0 between the input data X in its corresponding label y.
- These two distributions are not same in the context of pattern or concept.

[3] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM computing surveys (CSUR), 46(4), 44.

Types of Concept Drift

Types of Concept Drift

According to Lui et al., [4]

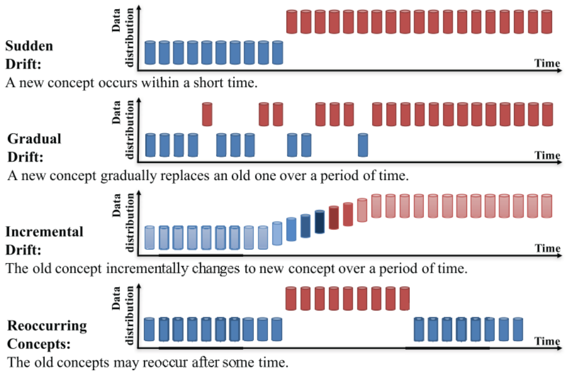

Figure 3: The types of concept drift

[4] Liu, A.,...(2017, August). Regional concept drift detection and density synchronized drift adaptation. Conference on Artificial Intelligence (pp. 2280-2286).

Detecting Concept Drift

Detecting Concept Drift

The first attempt to handle of concept drift with case-based technique was IB3 (Instance-based) learning [5].
Detecting concept drift-

- Monitoring **raw data** (By data distribution)
- Monitoring **parameters of learners** (By parameters)
- Monitoring prediction **errors of learners** (By learners output)

[5] Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. Machine learning, 6(1), 37-66.

Practical Problem in Classification

Practical problem in classification

Being the environments non stationary it exhibits class imbalance that causes the practical problem in classification i.e., one of the most major problem in data mining and machine learning [6].

- It requires a lot of effort to observe the **new minority class instances in new incoming data**.
- The techniques of **incremental learning in imbalance datasets** (i.e., limited label data and plenty unlabeled data) with semi-supervised learning are more complex **compare to the techniques of fixed distributions**.

[6] Hoens, T. R., & Chawla, N. V. (2012). Learning in non-stationary environments with class imbalance. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12. https://doi.org/10.1145/2339530.2339558

Implications in Software Engineering Research

Implications in SE Research/1

- Prediction model are developed by using historical datasets (i.e., fixed distributions)
- Proposed models to predict the number and the location of future bugs in software source codes
- Such predictions can help the project manager to lead the project by proper utilizing the testing resources
- Unfortunately, **the prediction model may not work if the historical datasets changes overtime** [7]

[7] Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A. (2009). Tracking concept drift of software projects using defect prediction quality. Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009, 51–60.

Implications in SE Research/2

- Due to the **change of some influencing features**, bug generation process becomes more unsuitable. [8]
- They observed that **the change in number of author editing a file and a number of defects fixed by them introduce the concept drift**.
- They have also seen that the **prediction quality significantly varies over time**.

[8] Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A. (2012). Time variance and defect prediction in software projects. Empirical Software Engineering, 17(4–5), 348–389.

Implications in SE Research/3

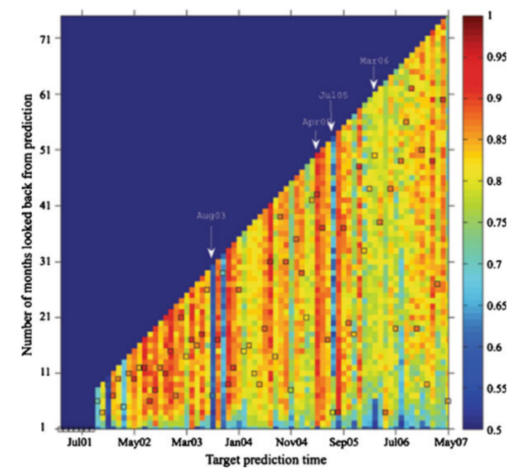

Figure 4: Eclipse heat-map: Prediction quality using different training periods with the points of highest AUC highlighted

Implications in SE Research/4

- The maximum **AUC values** for each target period typically lie on neither ends.
- **The models should not be trained on data collected from a very long or very short history.** [9]

[9] Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A. (2012). Time variance and defect prediction in software projects. Empirical Software Engineering, 17(4–5), 348–389.

Current studies

1. Bernstein, A., Ekanayake, J., & Pinzger, M. (2007, September). **Improving defect prediction using temporal features and non linear models**. In Ninth international workshop on Principles of software evolution: in conjunction with the 6th ESEC/FSE joint meeting (pp. 11-18). ACM.
2. Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A. (2009, May). **Tracking concept drift of software projects using defect prediction quality**. In Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on (pp. 51-60). IEEE.
3. Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A. (2012). **Time variance and defect prediction in software projects**. Empirical Software Engineering, 17(4-5), 348-389.
4. Finlay, J., Pears, R., & Connor, A. M. (2014). **Data stream mining for predicting software build outcomes using source code metrics**. Information and Software Technology, 56(2), 183-198.

Motivation

- Rahul et al. [10] cautions that if the prediction model changes because of the changing of historical data, managers will unable to faith on the predictions models causing to fail the proper use of testing resources.
- According to Dong et al. [11], a well-trained prediction model could give more inaccurate results if the upcoming data starts to drift in times.

---

[10] R. Krishna and T. Menzies, "Bellwethers: A Baseline Method For Transfer Learning," in IEEE Transactions on Software Engineering.

[11] Dong, J. Lu, K. Li and G. Zhang, "Concept drift region identification via competence-based discrepancy distribution estimation," 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, 2017, pp. 1-7.

Future Work

- Explore the concept drift detection methods by monitoring the parameters of learners and data distributions
- Being the environments non-stationary, it exhibits class imbalance. Our plan is also to explore the techniques that handle class imbalance in streaming data.
- Compare with the methods of bellwether with concept drift to produce a better prediction models

Thank you so much for listening!