A feasibility study on concept drift in Empirical Software Engineering

Presenter:

Md Alamgir Kabir

Supervisor:
Dr. Jacky Keung
Department of Computer Science, CityU

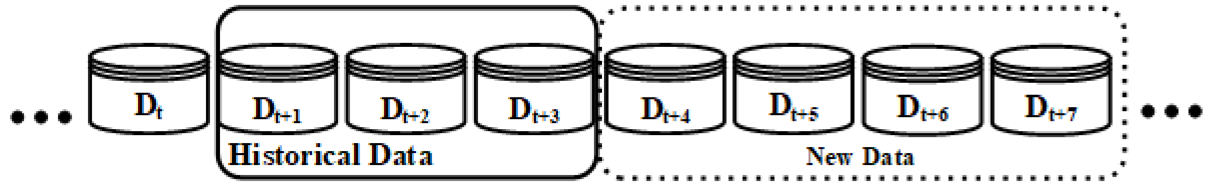
Outline

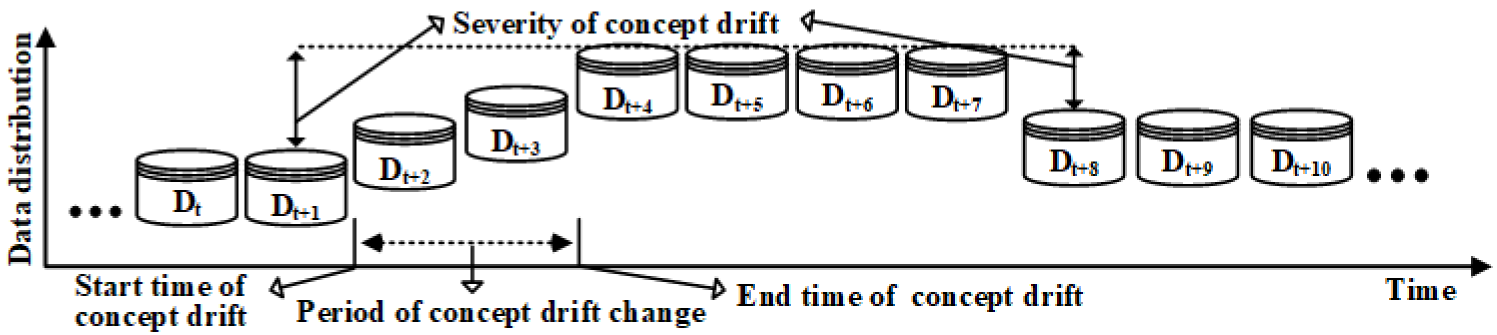
- Concept Drift
- Implications in Software Engineering
- Current Studies
- Motivation
- Concept Drift Approach
  - Drift Detection Method (DDM)
- Experiment Setup
- Discussion on Experiment Results
- Future works
- Conclusions

Concept Drift

- Streaming data: Incoming data changes over time
- Concept drift: The changes in the distributions over time
- Concepts are not often stable but change with time such as weather prediction and customers' preference

Concept Drift





J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering. doi: 10.1109/TKDE.2018.2876857

Implications in Software Engineering/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
- The prediction model may not work if the historical datasets changes overtime (Ekanayake @ MSR'2009)

Implications in Software Engineering/2

- Ekanayake et al. observed that the change in number of author editing a file and a number of defects fixed by them introduce the concept drift.
(Ekanayake @ ESM' 2012)

- They have also observed that the prediction quality significantly varies over time.

Current Studies

1. Bernstein, A., Ekanayake, J., & Pinzger, M. (2007, September**). Improving defect prediction using temporal features and nonlinear 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. [1] 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. [2], a well-trained prediction model could give more inaccurate results if the upcoming data starts to drift in times.

[1] R. Krishna and T. Menzies, ”Bellwethers: A Baseline Method For Transfer Learning,” in IEEE Transactions on Software Engineering. doi: 10.1109/TSE.2018.2821670

[2] 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. doi: 10.1109/ISKE.2017.8258734

Concept drift approaches

- By monitoring data distributions
  (N. Lu et al. @ Artif. Intell.'2014) ; J. Lu et al. @ TKDE'2018)

- By monitoring error-rate of learning algorithms
  (J. Gama @'2004 ; J. Lu et al. @ TKDE'2018)

Drift Detection Method (DDM)

- Monitor the error-rate of learning algorithms
- When the distribution changes, the error-rate will increase
- While the distribution stationary, the error-rate will decrease
- Good performance detecting drift and learning new concept

J. Gama, P. Medas, G. Castillo, and P. Rodrigues, ”Learning with Drift Detection,” In: Bazzan A.L.C., Labidi S. (eds) Advances in Artificial Intelligence SBIA 2004. Lecture Notes in Computer Science, pp. 286 - 295, 2004

Experiment Setup

- Learners: Naïve Bayes and Decision Tree
- Statistical Method: Chi-square test with Yates's continuity correction
  - Significant decrease or increase comparing to recent accuracy suggests that the concept is changing

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}(1-\hat{p})(1/n_o + 1/n_r)}}$$

$$\widehat{p} = (r_o + r_r)/(n_o + n_r)$$

  - If the p-value is less than significant level, the null hypothesis rejected and alternative hypothesis is accepted

K. Nishida and K. Yamauchi, "Detecting Concept Drift Using Statistical Testing," *Discov. Sci. DS 2007. Lect. Notes Comput. Sci.*, vol. 4755, pp. 264–269, 2007.

Experiment Setup

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}(1-\hat{p})(1/n_o + 1/n_r)}}$$

$$\widehat{p} = (r_o + r_r)/(n_o + n_r)$$

- r_o = number of correct classifications
- n_o = overall examples
- r_r = number of correct classification in recent examples
- n_r = recent examples

K. Nishida and K. Yamauchi, "Detecting Concept Drift Using Statistical Testing," *Discov. Sci. DS 2007. Lect. Notes Comput. Sci.*, vol. 4755, pp. 264–269, 2007.

Experiment Setup

- Datasets
  - Tim Menzies datasets
  - Uses the CK OO metrics

Project Name	Release	Module	# Defects
Jm1	1	7782	1672
Jm1	1.2	9593	1759
Jm1	1.3	7782	1672

Experiment Setup

- Datasets
  - Marian Jureckzo datasets – 27 versions of Prop project
  - Uses the CK OO metrics

Project Name	Version	Module
Prop-1	4, 40, 85, 121, 157, 185	18471
Prop-2	9, 44, 92, 128, 164, 192	23014
Prop-3	225, 236, 245, 256, 265	10274
Prop-4	285, 292, 305, 318	8718
Prop-5	347, 355, 362	8516
Prop-6	452, 453, 454	660

Discussion/1

jm-naïve bayes	Training set	precision	pd	fmeasure	gmean	balance	accuracy
	500	0.538	0.226	0.318	0.461	0.451	0.775
	1000	0.279	0.246	0.261	0.45	0.452	0.696
	1500	0.35	0.193	0.249	0.418	0.425	0.758
	2000	0.336	0.305	0.32	0.503	0.494	0.716
	2500	0.345	0.318	0.331	0.515	0.504	0.725
	3000	0.675	0.558	0.611	0.621	0.619	0.62
	3500	0.048	0.013	0.02	0.113	0.302	0.965
	4000	0.172	0.178	0.175	0.41	0.417	0.9
	4500	0.342	0.422	0.378	0.572	0.561	0.699
	2657	0.343	0.313	0.328	0.513	0.501	0.728

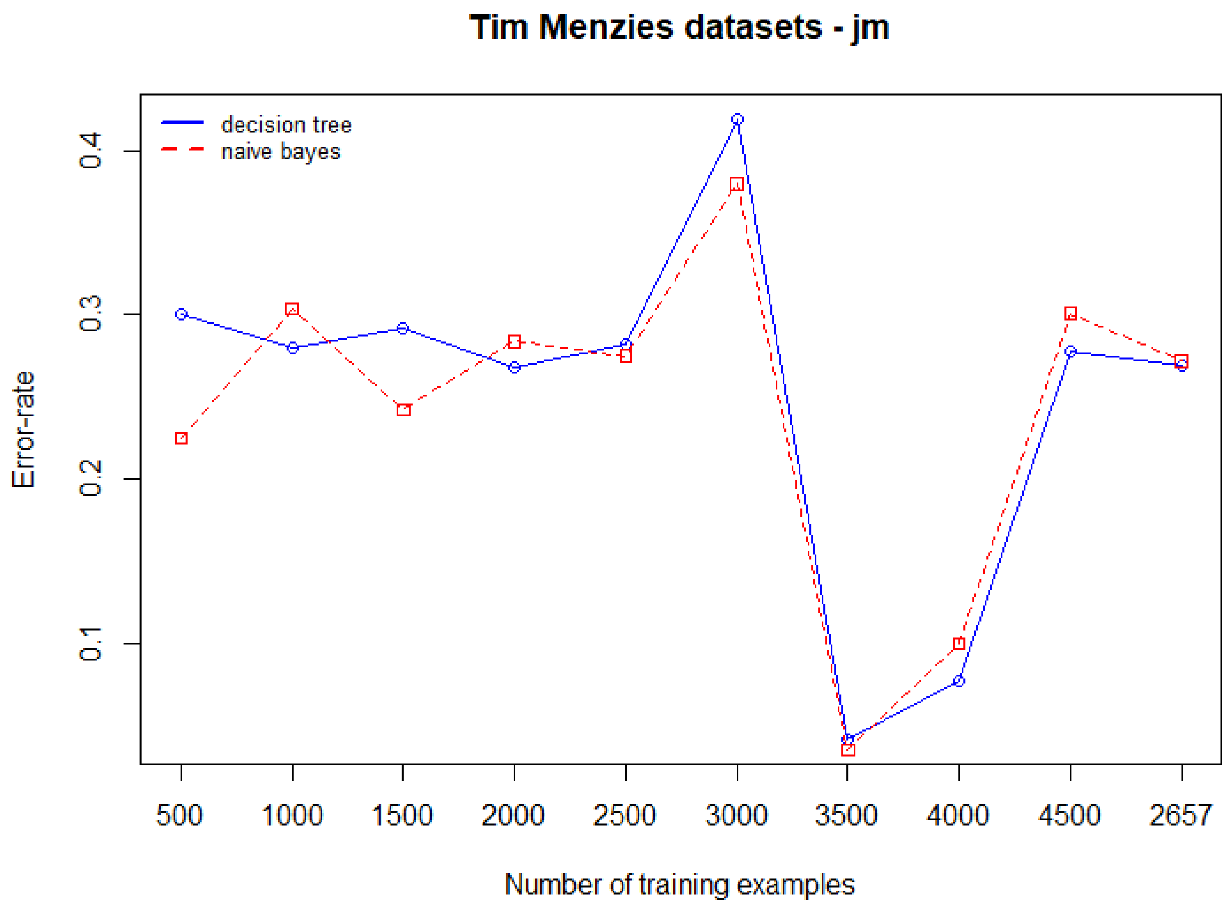
Jm-decision tree	Training set	precision	pd	fmeasure	gmean	balance	accuracy
	500	0.255	0.151	0.189	0.361	0.392	0.7
	1000	0.237	0.126	0.164	0.334	0.377	0.72
	1500	0.166	0.1	0.125	0.295	0.357	0.708
	2000	0.256	0.117	0.16	0.325	0.372	0.732
	2500	0.24	0.147	0.183	0.359	0.39	0.718
	3000	0.599	0.661	0.628	0.568	0.566	0.581
	3500	0.065	0.038	0.048	0.195	0.32	0.958
	4000	0.176	0.079	0.109	0.277	0.348	0.923
	4500	0.225	0.117	0.154	0.322	0.37	0.722
	2657	0.24	0.124	0.164	0.334	0.376	0.731

Discussion/2

	Overall Examples	Recent Examples	p-value
jm-naïve bayes	500	1000	0.263804
	1000	1500	0.248608
	1500	2000	0.295549
	2000	2500	0.717368
	2500	3000	0.10155
	3000	3500	0.027421
	3500	4000	0.121493
	4000	4500	0.041416
	4500	2657	0.312574

	Overall Examples	Recent Examples	p-value
jm-decision tree	500	1000	0.68734
	1000	1500	0.746536
	1500	2000	0.47504
	2000	2500	0.603108
	2500	3000	0.078384
	3000	3500	0.025408
	3500	4000	0.221955
	4000	4500	0.040859
	4500	2657	0.656246

Discussion/3



Discussion/4

	Training Examples	precision	pd	fmeasure	gmean	balance	accuracy
prop-naïve bayes	500	0.083	0.029	0.043	0.169	0.313	0.89
	1000	0.071	0.018	0.029	0.134	0.306	0.916
	1500	0.378	0.283	0.324	0.518	0.492	0.882
	2000	0.702	0.711	0.707	0.828	0.794	0.939
	2500	0.298	0.199	0.239	0.429	0.431	0.819
	3000	0.579	0.507	0.541	0.65	0.632	0.731
	3500	0.33	0.203	0.251	0.436	0.434	0.84
	4000	0.304	0.275	0.289	0.504	0.484	0.854
	4500	0.303	0.267	0.284	0.509	0.482	0.936
	5000	0.403	0.38	0.391	0.599	0.56	0.892
	5500	0.451	0.617	0.521	0.671	0.669	0.7
	6000	0.277	0.255	0.265	0.491	0.472	0.895
	6500	0.175	0.192	0.183	0.429	0.428	0.925
	7000	0.267	0.421	0.327	0.583	0.569	0.753
	7500	0.254	0.339	0.29	0.551	0.527	0.843
	8000	0.294	0.482	0.365	0.613	0.602	0.732
	2053	0.26	0.187	0.217	0.422	0.424	0.89

Discussion/5

	Overall Examples	Recent Examples	p-value
prop-naïve bayes	500	1000	0.571133
	1000	1500	0.382681
	1500	2000	0.206242
	2000	2500	0.090992
	2500	3000	0.118753
	3000	3500	0.086954
	3500	4000	0.511022
	4000	4500	0.094793
	4500	5000	0.163062
	5000	5500	0.029752
	5500	6000	0.027926
	6000	6500	0.207741
	6500	7000	0.03753
	7000	7500	0.06989
	7500	8000	0.060321
	8000	2053	0.063944

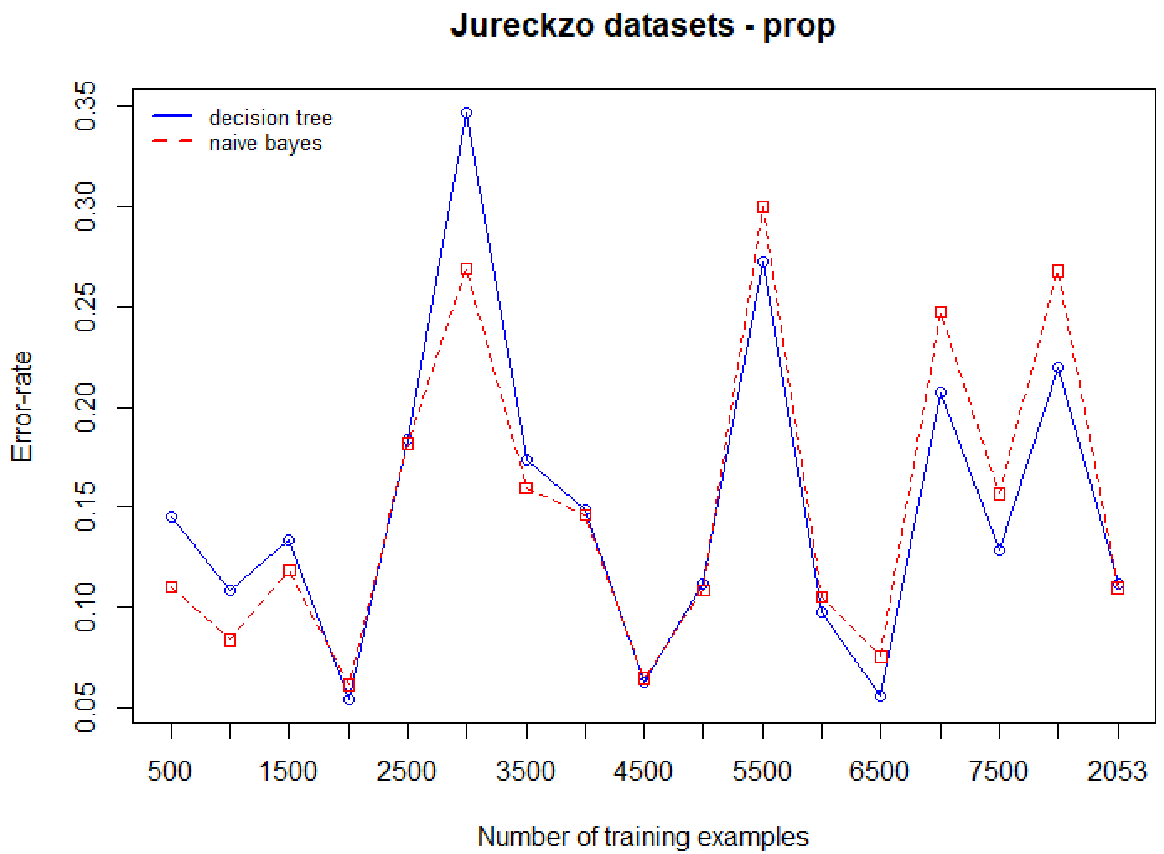
Discussion/6

	Training Examples	precision	pd	fmeasure	gmean	balance	accuracy
prop-naïve bayes	500	0.083	0.029	0.043	0.169	0.313	0.89
	1000	0.071	0.018	0.029	0.134	0.306	0.916
	1500	0.378	0.283	0.324	0.518	0.492	0.882
	2000	0.702	0.711	0.707	0.828	0.794	0.939
	2500	0.298	0.199	0.239	0.429	0.431	0.819
	3000	0.579	0.507	0.541	0.65	0.632	0.731
	3500	0.33	0.203	0.251	0.436	0.434	0.84
	4000	0.304	0.275	0.289	0.504	0.484	0.854
	4500	0.303	0.267	0.284	0.509	0.482	0.936
	5000	0.403	0.38	0.391	0.599	0.56	0.892
	5500	0.451	0.617	0.521	0.671	0.669	0.7
	6000	0.277	0.255	0.265	0.491	0.472	0.895
	6500	0.175	0.192	0.183	0.429	0.428	0.925
	7000	0.267	0.421	0.327	0.583	0.569	0.753
	7500	0.254	0.339	0.29	0.551	0.527	0.843
	8000	0.294	0.482	0.365	0.613	0.602	0.732
	2053	0.26	0.187	0.217	0.422	0.424	0.89

Discussion/7

	Overall Examples	Recent Examples	p-value
prop-decision tree	500	1000	0.519514
	1000	1500	0.497514
	1500	2000	0.151435
	2000	2500	0.084665
	2500	3000	0.065097
	3000	3500	0.0562
	3500	4000	0.315242
	4000	4500	0.089983
	4500	5000	0.145608
	5000	5500	0.033338
	5500	6000	0.029807
	6000	6500	0.147845
	6500	7000	0.04183
	7000	7500	0.079372
	7500	8000	0.074134
	8000	2053	0.095129

Discussion/8



Future Work

1. Explore the concept drift detection methods by monitoring the parameters of learners and data distributions

2. Being the environments non-stationary, it exhibits class imbalance. Our plan is also to explore the techniques that handle class imbalance in streaming data.

3. Compare with the methods of bellwether moving window with concept drift to produce a better prediction models

Conclusions

- The target of our experiment is to investigate to check the concept drift existence in software defect datasets.

- Based on our experiment, we observe that the concept drift exits in the defect datasets.

- The prediction models would not give accurate results if the training examples are taken from these datasets.

Thanks for listening!