

Miguel Estevez

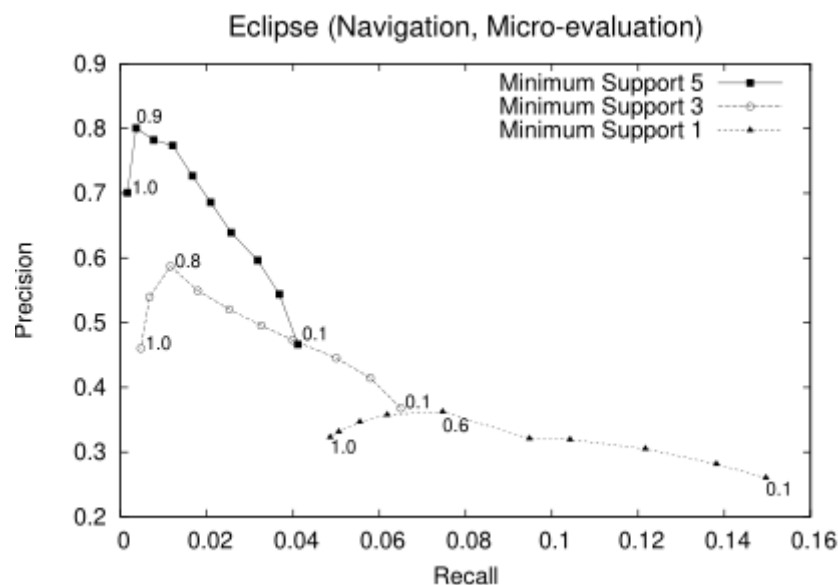
2017-0200

Recomendación y predicción de posibles cambios:

- Usan técnicas de minería de datos para obtener reglas de asociación a partir de las historias de versión.
- Evaluar la habilidad de predecir cambios futuro o que falte.

El servidor rose lee los archivos de version, agrupa los cambios en transacciones, mina las transacciones por reglas la cuales describen las implicaciones entre entidades de software. Cuando el cliente rose consulta el conjunto de reglas aplicables y hace las recomendaciones apropiadas para futuros cambios.

- Una entidad es una triada (f,c,i), donde f es el nombre del archivo afectado, c es la categoría sintáctica del componente afectado y i es el identificador del componente afectado.
- Se adquieren las transacciones de los CVS.
- Para inferir una transacción se usó sliding window donde dos cambios consecuentes del mismo autor es parte de una transacción si solo paso a lo mucho 200 segundos.



- Para la validación de una regla se usó confidence y support Se utilizo a priori pero se usaron algunas optimización como son: restricción de antecedentes(sólo se buscaban las reglas que estaba relacionada con la situación)y solo se utilizaba un consecuente. Con lo que se consiguió un 0.5s tiempo de corrida en promedio.
- Para evaluar se utilizaron 8 grandes proyectos de código abierto, donde se tomaron las últimas 1000 transacciones y se trató de predecirlas desde el historial anterior.

	Navigation			Prevention		Closure	
Support Confidence	1 0.1			3 0.9		3 0.9	
Project	R_{μ}	P_{μ}	L_3	R_{μ}	P_{μ}	R_M	P_M
ECLIPSE	0.15	0.26	0.53	0.02	0.48	1.0	0.979
GCC	0.28	0.39	0.89	0.20	0.81	1.0	0.953
GIMP	0.12	0.25	0.91	0.03	0.71	1.0	0.978
JBOSS	0.16	0.38	0.69	0.01	0.24	1.0	0.981
JEDIT	0.07	0.16	0.52	0.004	0.59	1.0	0.986
KOFFICE	0.08	0.17	0.46	0.003	0.24	1.0	0.990
POSTGRES	0.13	0.23	0.59	0.03	0.66	1.0	0.989
PYTHON	0.14	0.24	0.51	0.01	0.50	1.0	0.986
Average	0.15	0.26	0.64	0.04	0.50	1.0	0.980

Para la evaluación se tomaron en cuentas varios aspectos:

- Navegación por el código.
 - Se obtuvo 15% recall promedio y 26% de precisión promedio.
- Prevención de errores.
 - Se obtuvo una precisión promedio de 50% y un recall de 4%
- Cierre
 - Se obtuvo una precisión promedio de 98% y un recall de 100%
- Granualidad
 - Al minar a los archivos en vez de las variables o funciones se obtuvo un mejora en el modelo.

	Navigation			Prevention		Closure	
Support Confidence	1 0.1			3 0.9		3 0.9	
Project	R_{μ}	P_{μ}	L_3	R_{μ}	P_{μ}	R_M	P_M
ECLIPSE	0.17	0.26	0.54	0.03	0.48	1.0	0.980
GCC	0.44	0.42	0.87	0.29	0.82	1.0	0.946
GIMP	0.27	0.26	0.90	0.08	0.74	1.0	0.963
JBOSS	0.25	0.37	0.64	0.05	0.44	1.0	0.980
JEDIT	0.25	0.22	0.68	0.01	0.44	1.0	0.984
KOFFICE	0.24	0.26	0.67	0.04	0.61	1.0	0.971
POSTGRES	0.23	0.24	0.68	0.05	0.59	1.0	0.978
PYTHON	0.24	0.36	0.60	0.03	0.67	1.0	0.991
Average	0.26	0.30	0.70	0.07	0.66	1.0	0.973

Uno puede o tener recomendaciones precisas o muchas recomendaciones pero no ambas.

En mi opinión, esta investigación es muy interesante ya que me sorprendió en el área de desarrollo, no había pensado en esta posible aplicación de la minería de datos. Aunque los resultados no fueron muy altos con excepción del cierre, fue interesante ver como un poco el cambio del enfoque hizo que subiera considerable la precisión y el recall. Haciendo que en este problema haya una dualidad de la granularidad con la precisión y el recall.

