

REDES BAYESIANAS APLICADAS A LA INGENIERÍA DE SOFTWARE

Carlos Said, Rosana Piergallini, Nelson Di Grazia, Sabrina Pompei.

Escuela de Tecnología, Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA), Buenos Aires, Argentina

carlossaid@gmail.com, mrosanapierg@yahoo.com.ar, nelson.digrazia@telefonica.com,
sabrinapompei@unnoba.edu.ar,

Resumen

En ausencia de una medida acordada de la calidad del software, la densidad de defectos ha sido una medida comúnmente usada. En consecuencia ha habido numerosos intentos para construir modelos para la predicción del número de defectos residuales en el software. Las variables clave en dichos modelos son en general métricas de tamaño y complejidad o medidas derivadas del testing de la información y la codificación. Sin embargo estos enfoques presentan dificultades estadísticas y teóricas. El uso de redes de creencias bayesianas puede superar algunos de estos problemas, teniendo en cuenta los diversos factores implícitos en la prevención de defectos, detección y complejidad.

La línea de investigación propuesta se centra en el uso de redes de creencias bayesianas aplicado al modelo de densidad de defectos proporcionando un nuevo enfoque para los procesos de modelización y de ingeniería de artefactos de software. La naturaleza dinámica de este modelo proporcionará una manera de simular diferentes hechos e identificar cursos óptimos de acción basados en el conocimiento incierto.

Palabras clave: *bayes, ingeniería de software, calidad, métricas, costos, redes bayesianas de conocimiento.*

Contexto

Esta línea de investigación fue aprobada por la Secretaría de Investigación, Desarrollo y Transferencia de la UNNOBA, en el marco de la convocatoria para la acreditación de proyectos y solicitud de subsidios de investigación bianuales (SIB) 2013.

El equipo está constituido por docentes e investigadores del departamento de Informática y Tecnología, quienes dictan clases en las asignaturas Métricas de Software, Ciencias de la Computación, Análisis y Diseño de Sistemas y Programación Orientada a Objetos pertenecientes a las carreras Licenciatura en Sistemas e Ingeniería en Informática, en la Escuela de Tecnología de la UNNOBA.

Introducción

La forma más común de insertar defectos en un producto de software es mediante la aplicación de cambios al mismo y/o mediante la ausencia de aplicación de metodologías de testing de manera exhaustiva. En general, esto último ocurre ya que la llamada ‘crisis de producción de software’ explica que al llegar a las etapas de testing en el proceso de producción de software, los tiempos y presupuestos se han ‘agotado’. Otras

intenciones de explicar esta carencia de productos confiables se sustentan en la ausencia de metodologías aplicadas de control de producción orientadas a la ‘confiabilidad’ del producto. En cada versión de software, decenas o cientos de cambios se aplican al producto de software. Todo cambio tiene su característica única basada en el proceso y los factores del producto. Esta densidad de defectos ha sido una medida comúnmente usada en la calidad del software.

Cada cambio tiene su propia probabilidad de inyectar un defecto. En consecuencia han habido numerosos intentos para construir modelos para la predicción del número de defectos residuales en el software. Por lo general las variables clave en los modelos son o métricas de tamaño y complejidad o medidas derivadas del ‘testing’ de la información y la codificación. Hay sin embargo dificultades estadísticas y teóricas con estos enfoques.

El uso de redes de creencias bayesianas puede superar algunos de los problemas presentes, teniendo en cuenta los diversos factores implícitos en la prevención de defectos, detección y complejidad. A nivel general podemos ver cómo el uso de estas redes aplicadas al modelo de densidad de defectos proporciona un nuevo enfoque importante para los procesos de modelización y de ingeniería de artefactos de software.

La naturaleza dinámica de este modelo proporciona una manera de simular diferentes hechos e identificar cursos óptimos de acción basados en el conocimiento incierto. Estos beneficios se ven reforzados cuando examinamos cómo el modelo explica resultados conocidos, en particular el dilema ‘es más grande mejor?’.

Este nuevo enfoque muestra cómo podemos construir redes complejas de interconexión entre los procesos, productos y recursos de una manera hasta ahora poco viable. También podemos integrar la incertidumbre y los criterios subjetivos en el modelo sin sacrificar el rigor, e ilustrar cómo la toma de decisiones durante el proceso de desarrollo de software tiene influencia en la calidad alcanzada.

Los beneficios esperados de este enfoque son:

- mayor utilidad para la gestión de proyectos que el análisis de valores atípicos y los estadísticos clásicos;
- Incorporación de ideas actuales de la investigación y de la experiencia;
- uso para formación de los gerentes de proyectos de software ;
- permitir la comparación de diferentes decisiones por medio de simulación y los análisis ‘qué sucede si?’;
- Integración de una forma de predicción de costo y calidad.

Se busca desarrollar un modelo de diseño orientado por objetivo, lo que suele llamarse ‘goal oriented’, una descripción detallada del marco de medición del proceso que incluya los conceptos correspondientes de minería de datos. El enfoque se basa en los datos del proceso y los del producto. El modelo resultante es llamado el Modelo de Producción del Software.

Este Modelo de Producción del Software estará construido sobre la base de una versión ampliada del proceso de construcción de software modelo-V que a su vez se deriva del modelo de cascada. Los marcos de procesos y productos de medición combinados con el conocimiento de los expertos describen todas las fases del modelo-V, y de esta manera permite la

representación como un modelo coherente del proceso completo de ingeniería.

El Modelo de Producción del Software estará diseñado principalmente para optimizar la calidad del producto y la distribución del esfuerzo de desarrollo en el tiempo. Permitirá evaluar las medidas del proceso que influyen en la calidad del producto. Debido a las capacidades dinámicas del mismo permitirá además analizar el impacto de los posibles cambios en los procesos y productos, que ocurren en el tiempo.

Estas capacidades ofrecerán visiones diferentes a los administradores de proyectos vinculados a productos de software, para poner de relieve la utilidad práctica del modelo.

El Modelo de Producción del Software estará basado en una Red Bayesiana Dinámica, con capacidad para representar múltiples características de cambio de una versión de software específico. Permitirá a los administradores de proyectos supervisar el impacto de cada cambio realizado en esta versión del software tomando estas características de cambio individuales en cuenta.

Las Redes Bayesianas, elemento de base para cada Red Bayesiana Dinámica, son parte de las técnicas modernas de inteligencia artificial (AI). Aunque un teorema fundamental de la probabilidad condicional fue presentado por Thomas Bayes en el siglo XVIII, el término "Red Bayesiana" y sus conceptos fueron introducidos en la década de 1980 en el trabajo de Judea Pearl. Investigadores han presentado declaraciones referidas a las Redes Bayesianas, por ejemplo Darwiche en 2009: "Las Redes Bayesianas son tan importantes para la Inteligencia Artificial (AI) y el Aprendizaje Automático como los circuitos booleanos son a la informática", "

las Redes Bayesianas han revolucionado la AI".

Una Red Bayesiana Dinámica es un modelo de formación de una secuencia de Redes Bayesianas individuales donde cada Red individual refleja el estado del sistema en un punto de tiempo específico. Una Red Bayesiana puede ser percibida desde dos perspectivas: (1) como una gráfica donde los nodos representan variables aleatorias y los arcos representan las relaciones entre las variables, (2) como una definición de modelo matemático formal donde las variables se expresan como distribuciones de probabilidad condicional.

Las Redes Bayesianas se pueden construir mediante la aplicación de un algoritmo de aprendizaje para el conjunto de datos que permite el aprendizaje de una estructura. El mayor potencial de las mismas reside en la capacidad de combinar el conocimiento experto con los datos empíricos.

Otros beneficios de estas redes incluyen:

- capacidad de reflejar las relaciones causales,
- ejecutar el modelo con datos incompletos y supuestos,
- razonamiento hacia adelante y hacia atrás,
- modelización de la incertidumbre,
- combinación de datos cuantitativos y cualitativos.

Líneas de investigación y desarrollo

- Redes bayesianas en la ingeniería de software: redes bayesianas, toma de decisión frente a la incertidumbre, inferencias aplicando redes bayesianas, limitaciones de las redes bayesianas;

- Calidad del software: modelos de análisis de la calidad del software, modelos de predicción de calidad en el software, métricas de calidad, paradigmas para el diseño confiable.
- Herramientas: herramientas para la construcción de Redes Bayesianas de conocimiento (BBN), comparación y selección.

Resultados y Objetivos

El objetivo general del presente proyecto es construir un modelo de estimación de defectos en la construcción de artefactos de software de mayor precisión que el causal clásico.

Objetivos específicos:

- Mejorar la precisión de estimación de defectos en la construcción de software respecto al modelo causal clásico.
- Definir un Modelo que responda ante características como datos incompletos, variables estocásticas e incertidumbre.

Formación de Recursos Humanos

Integran el equipo de trabajo un docente investigador, quien dirige el proyecto, y tres docentes investigadores en formación.

El proyecto fue aprobado en Diciembre 2012 por lo que dentro de la línea presentada no se cuenta aún con tesis y/o tesinas presentadas.

Bibliografía

[1] Angelis, L., & Stamelos, I. (2000). A Simulation Tool for Efficient Analogy Based Cost Estimation. *Empirical Software Engineering*, 5, 35–68.

doi:10.1023/A:1009897800559

[2] Baresi, L., Morasca, S., & Paolini, P. (2003). Estimating the design effort of Web applications, In *Proceedings Ninth International Software Measures Symposium*, September 3-5, (pp. 62-72).

[3] Bibi, S., Stamelos, L., & Angelis, L. (2003) Bayesian Belief Networks as a Software Productivity Estimation Tool. In *Proceedings 1st Balkan Conference in Informatics*, Thessaloniki.

[4] Briand, L. C., El-Emam, K., Surmann, D., Wiczorek, I., & Maxwell, K. D. (1999). An Assessment and Comparison of Common Cost Estimation Modeling Techniques. In *Proceedings of ICSE 1999*, Los Angeles, CA, (pp. 313-322).

[5] Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., & Vitiello, G. (2006). Effort estimation modeling techniques: a case study for web applications, In *Proceedings of the Intl. Conference on Web Engineering (ICWE'06)*, (pp. 9-16).

[6] Druzdzal, M. J., & van der Gaag, L. C. (2000). Building Probabilistic Networks: Where Do the Numbers Come From? *IEEE Transactions on Knowledge and Data Engineering*, 12(4), 481–486. doi:10.1109/TKDE.2000.868901

[7] Kuo, W. (2007) Compatibility and simplicity: the fundamentals of reliability, *IEEE Trans. Reliab.*, vol. 56.

[8] Raheja, D. G., and Allocco M. (2006) Assurance Technologies Principles and Practices: A Product, Process, and System Safety Perspective, 2nd ed., Wiley, Hoboken, NJ.

[9] Kan, S. H., Metrics and Models in Software Quality Engineering, Addison-

Wesley, Reading, MA, 1995, p. 192
(Rayleigh model discussion).

[10] IEEE Guide for Selecting and Using
Reliability Predictions, IEEE 1413.1–2002
(based on IEEE 1413), IEEE, Piscataway, NJ.

[11] Analysis Techniques for System
Reliability: Procedure for Failure Mode and
Effects Analysis (FMEA), IEC 60812,
International Electrotechnical Commission,
Geneva, 2006.