Proceedings of the VI Brazilian Conference on Neural Networks - VI Congresso Brasileiro de Redes Neurais
pp. 73–78, June 3–5, 2003 - Centro Universitario da FEI, Sao Paulo - SP - Brazil

Utilizando Redes Neurais para Estimar o Tamanho de um Software

Gustavo Alexandre de Souza, Aurora Trinidad Ramirez Pozo, Silvia Regina Vergilio
Departamento de Informatica
Universidade Federal do Paraná
UFPR-DInf, CP: 19081,
CEP:81531-970, Curitiba -Brazil
{gustavo, aurora, silvia}@inf.ufpr.br

Abstract

Prediction models are fundamental at the early stages of the software development where decisions must be taken without the required information. A typical information that is not available in these stages is the number of lines of code (LOC), that is the most known and used software size metric. LOC estimation is a hard task that includes historical data and empirical studies. Usually, models for LOC estimation are obtained using statistical regression methods. However, the characteristics of the LOC estimation task make this problem specially interesting for the application of neural network techniques. This work explores Neural Networks techniques in this context. Two different data sets were used to obtain two models to estimate LOC using respectively the metrics function points and number of components, as independent variables. The results show some insights about the use of Neural Network for this task.

1. Introdução

As medidas de tamanho de software tais como as fornecidas pela métrica número de linhas do código (Lines of COde - LOC) são bastante utilizadas em diversas fases da Engenharia de Software. Elas nos fornecem meios de quantificar diversos fatores associados à qualidade de software, assim como auxiliar no cálculo da produtividade e nas atividades de planejamento do projeto de software, manutenção, documentação e teste.

A métrica LOC é frequentemente escolhida por ser facilmente compreendida e fácil de coletar após o término do projeto, pois é uma métrica direta [11], ou seja não está associada a valores subjetivos. Entretanto, nos primeiros estágios do desenvolvimento de um software onde muitas decisões precisam ser tomadas baseadas no tamanho do software sendo desenvolvido, o valor para LOC precisa ser estimado, por não estar ainda disponível.

Para resolver o problema de estimação do tamanho do software, nesse estágio inicial do projeto, existem diferentes abordagens [12], relacionadas à funcionalidade do software sendo desenvolvido. Nesse contexto, dois métodos se destacam. O método de Function Points (FP) [1] que considera o número de pontos de função obtido a partir de diversas características do software, e o método baseado em componente (NOC - number of components) [13] que utiliza os tipos de componentes do software, tais como: subsistemas, módulos, telas de interface, arquivos, relatórios, etc.

As métricas FP e NOC são geralmente utilizadas em modelos de predição do número de LOC. Esses modelos são geralmente obtidos a partir de uma base de dados histórica, formada por informações coletadas de projetos passados. Apesar do esforço da comunidade científica em resolver o problema de estimativa de LOC, os resultados alcançados apresentam diferentes graus de sucesso, porém nenhum método ainda tem se mostrado consistemente efetivo [9].

Os métodos de regressão múltipla têm sido tradicionalmente aplicados para estimar LOC [4]. Entretanto, para a utilização desses métodos é necessário decidir como será a regressão, linear ou quadrática, ou escolher uma família de polinômios ou outros tipos de função. Neste contexto, as técnicas de Aprendizado de Máquina surgem como uma alternativa aos métodos estatísticos. Mais recentemente, estas técnicas também estão sendo aplicadas na área de Engenharia de Software [2, 3, 6, 9]. A maioria desses trabalhos focaliza estimativa de custo e cronograma. Entre as técnicas exploradas estão raciocínio baseado em casos e programação genética, entre outras.

Este trabalho explora o uso de Redes Neurais para estimar o número de LOC de um software. A estimativa de LOC é feita a partir das duas métricas mencionadas acima (FP e NOC) e de dados históricos. A principal motivação para explorar RN nesse contexto é sua capacidade de aprender a partir de dados (inclusive com ruído), utilizar um conjunto de diferentes variáveis de entrada e não existir a necessidade de se pré-determinar o formato da equação. Desta forma, pode-se afirmar que a solução será verdadeiramente descoberta.

Encontrar bases de dados disponíveis para o nosso estudo não foi tarefa fácil. Não foi encontrada nenhuma base de projetos reais que contivesse ao mesmo tempo informações sobre LOC, FP e NOC. Devido a isso, foram utilizadas duas bases de dados diferentes. Uma base de dados foi utilizada para obter o modelo para estimar LOC a partir de FP [8] e uma outra para estimar LOC a partir de NOC [5]. Assim dois modelos foram obtidos. A

diferença entre eles é reflexo das características das bases utilizadas e mostram alguns pontos a serem considerados na obtenção de modelos e no uso de redes neurais nesse contexto.

O trabalho está organizado da seguinte maneira. Na Seção 2, é apresentado um breve resumo de trabalhos similares. A Seção 3 descreve o estudo realizado, bases de dados utilizadas e a configuração da rede. A Seção 4 apresenta e discute os resultados encontrados. A Seção 5 conclui o artigo e mostra os possíveis desdobramentos para o trabalho.

2. Trabalhos Relacionados

A estimativa para o número de LOC nos estágios iniciais do desenvolvimento de um software é fundamental para auxiliar os gerentes de projeto. Nessa etapa eles se deparam com um verdadeiro desafio: tomar decisões sem ter disponibilidade de todas as informações necessárias [11]. Portanto, para auxiliá-los nessa tarefa, as estimativas obtidas precisam ser as melhores e mais confiáveis possíveis.

Para se estimar LOC a partir de dados históricos, tradicionalmente têm sido utilizados os métodos estatísticos de regressão múltipla [4]. Esses métodos, entretanto, necessitam um conhecimento prévio do comportamento da equação que se deseja descobrir. Diversas tentativas e escolhas, ou seja, regressões, são realizadas para que a equação seja obtida.

Técnicas de Aprendizado de Máquina (AM) têm sido utilizadas como uma alternativa aos métodos estatísticos de regressão, com bastante sucesso. Essas técnicas não necessitam um formato inicial, descobrindo a equação ou modelo que melhor representa os dados.

A utilização de técnicas de AM na Engenharia de Software é entretanto bastante recente. A maioria dos trabalhos focaliza estimativa de custo e cronograma e para tanto aplicam diferentes técnicas de AM: redes neurais [7], programação genética [3, 6], regras de indução [10], raciocínio baseado em casos [9], etc.

Nosso trabalho, diferentemente dos trabalhos acima mencionados, focaliza estimativa de tamanho de software, ou seja de LOC a partir de FP e NOC. Um trabalho que motivou o nosso foi o de Dolado [5]. Nesse trabalho Dolado utiliza algumas técnicas de AM para validar a métrica NOC em modelos de predição de LOC. Os modelos obtidos foram comparados com os modelos que utilizam a métrica FP.

O nosso estudo enfoca o uso de RN como alternativa para obter modelos de estimativa de LOC. Nós utilizamos as duas métricas mencionadas acima, obtendo por consequência dois modelos de estimativa de LOC. Entretanto, os nossos experimentos utilizam duas bases de dados com características diferentes. Estas peculiaridades mostram importantes aspectos que devem ser considerados na obtenção destes modelos.

3. Descrição do Experimento

O experimento foi realizado com o objetivo de estudar a aplicação de RN na obtenção de modelos de estimativa de LOC. Esta seção descreve as bases de dados, as medidas de avaliação utilizadas e os principais passos seguidos.

3.1. Bases Utilizadas

Foram utilizadas duas bases de dados para obtenção de dois modelos para estimar LOC.

O primeiro modelo estima LOC a partir de FP e foi obtido utilizando-se uma base de dados fornecida pelo ISBSG (International Software Benchmarking Standards Group) [8]. Esta base é formada por um esforço desse grupo em coletar informações de diferentes organizações. A base inclui uma grande variedade de projetos provenientes principalmente de aplicações comerciais, mas que se utilizam de diversas linguagens de programação e metodologias de desenvolvimento. A base é composta por 1238 projetos, mas somente 51 deles contêm informações sobre FP e por isso apenas esses foram utilizados no nosso estudo.

O segundo modelo, que estima LOC a partir do NOC e das características destes componentes, foi obtido utilizando-se a mesma base de dados utilizada por Dolado em [5]. Os projetos dessa base foram desenvolvidos por alunos de graduação e implementados em uma linguagem de quarta geração (Informix - 4GL). A base de dados é composta de 46 projetos, que incluem o LOC e também informações sobre diferentes tipos de componentes em cada projeto, classificados em: menus, entradas e relatórios. Nessa base, 4 projetos estavam incompletos e foram descartados.

As duas bases foram aleatoriamente divididas em dois conjuntos. Um conjunto de treinamento, contendo 2/3 dos projetos escolhidos aleatoriamente, o qual foi utilizado para treinar a rede e obter o modelo; e um outro conjunto, o conjunto de teste, composto pelo 1/3 restante dos projetos, para avaliar o modelo obtido.

3.2. Medidas de Avaliação

Para avaliar a acurácia do modelo obtido, foram utilizadas duas medidas baseadas no erro, ou seja na diferença entre o valor estimado e o observado no conjunto de teste. Essas medidas são bastante utilizadas em outros trabalhos da literatura [3] e descritas a seguir.

1) MMRE (Mean Magnitude of Relative Error) definida como:

$$MMRE = \sum_{i=1}^{n} \frac{Ea_i - Ep_i}{Ea_i}$$

onde n é o número de projetos, Ea_i é o valor real para a variável LOC em um projeto i, e Ep_i é o valor estimado. Quanto menor o valor para MMRE, melhor o

modelo. Para um modelo ser considerado bom, $MMRE$ deve ser ≤ 0.25 [5].

2) PRED(l) (Prediction at level l): é definida como o quociente entre o número de casos nos quais a estimativa está dentro de um limite **l** absoluto e o número total de casos. O critério padrão para ser aceitar um modelo como bom é $PRED(0.25) \geq 0.75$ [5]. Isto significa que pelo menos 75% das estimativas estão dentro de uma faixa de 25% de erro.

3.3. Configurando a Rede Neural

Para a implementação da rede neural (RN) foi utilizada uma ferramenta de domínio público que pode ser encontrada em [14]. As redes utilizadas podem ser visualizadas nas Figuras 1 e 2. Foi utilizado um modelo com três camadas sendo respectivamente uma camada de entrada, uma intermediária (escondida) e a camada de saída. Os parâmetros utilizados, tais como número de neurônios em cada camada, número de épocas, etc., são apresentados na Tabela 1. Foram treinadas duas redes, uma para estimar LOC a partir de FP com os dados do ISBSG [8], e outra para estimar LOC a partir do NOC e de suas características com os dados de Dolado [5]. Os mesmos parâmetros foram utilizados em ambas as redes, a única diferença, que pode ser observada na Tabela 1 é o número de neurônios da camada de entrada. Na primeira rede, apenas um neurônio foi utilizado para a camada de entrada, representando FP. Na segunda rede, foram utilizados 4 neurônios representando o NOC e características de três diferentes tipos de componentes: menus, entradas, e relatórios.

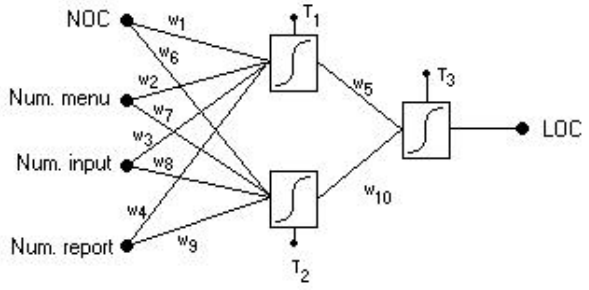

Figura 1: Modelo de RN Utilizando NOC

Tabela 1: Parâmetros utilizados

Parâmetros	**Valor**
n. de camadas	3
n. neur. camada entrada (FP)	1
n. neur. camada entrada (NOC)	4
n. neur. camada escondida	2
n. neur. camada saída	1
função para o neurônio	sigmoide
pesos iniciais	aleatórios
critério de parada	n. de épocas=1000

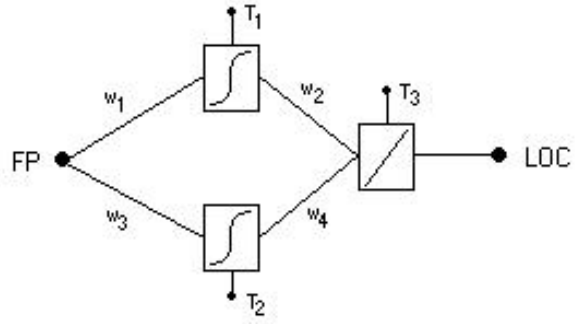

Figura 2: Modelo de RN Utilizando FP

4. Principais Resultados

Abaixo são apresentados os parâmetros que identificam as redes dos modelos obtidos. São quatro: 1) *units*: representa o número de neurônios das camadas de entrada, escondida e de saída; 2) *scales*: especifica as transformações lineares aplicadas para normalizar os valores de entrada; 3) *weights*: os pesos dos neurônios e "thresholds" nas diferentes camadas; e 4) *ranges*: especifica os intervalos dos valores de saída.

Modelo LOC x NOC

$$units = \begin{bmatrix} 4 & 2 & 1 \end{bmatrix}$$

$$scales = \begin{bmatrix} 33.7879 & 0.0560785 \\ 42.3636 & 0.0421942 \\ 197.455 & 0.0065583 \\ 61.7576 & 0.0249372 \end{bmatrix}$$

$$weights = \begin{bmatrix} 0.299 & 0.547 & -0.737 & -1.680 & 2.374 \\ -0.129 & -1.165 & 0.114 & -0.274 & 0.055 \\ -3.13026 & -1.57936 & & 2.54285 & \end{bmatrix}$$

$$ranges = \begin{bmatrix} 0 & 8888 \end{bmatrix}$$

Modelo LOC x FP

$$units = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$scales = \begin{bmatrix} 722.486 & 0.00151064 \end{bmatrix}$$

$$weights = \begin{bmatrix} -2.43673 & 1.95444 \\ -1.26068 & -0.648313 \\ -2.94679 & -1.07931 & -0.573801 \end{bmatrix}$$

$$ranges = \begin{bmatrix} 0 & 1.04169e+06 \end{bmatrix}$$

O parâmetro *weights* é composto pelas seguintes variáveis, representadas nas Figuras 1 e 2:

$$LOC : weights = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \\ w_6 & w_7 & w_8 & w_9 & w_10 \\ T_1 & T_2 & & T_3 & \end{bmatrix}$$

$$FP : weights = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \\ T_1 & T_2 & T_3 \end{bmatrix}$$

A acurácia dos modelos obtidos foram avaliadas utilizando-se os conjuntos de teste e as medidas de avaliação. As Figuras 3 e 4 representam graficamente os pontos de teste utilizados e os valores obtidos. As medidas para ambos os modelos estão na Tabela 2.

Observou-se durante a condução do nosso estudo, que o uso de RN no contexto de estimativa de tamanho de software se mostrou bastante apropriado e válido. A principal vantagem é que o modelo de estimação é realmente descoberto, não é necessário conhecer a curva da equação.

Analisando os dados da Tabela 2, pode-se notar que as acurácias das equações obtidas são bastante diferentes. O valor de MMRE do Modelo 1, que prevê LOC a partir de FP, indica que esse não é um modelo aceitável. O valor de MMRE do Modelo 2, entretanto, é considerado bom, indicando que o segundo modelo pode ser utilizado para prever LOC a partir de NOC.

Isso não quer dizer, entretanto, que NOC é melhor que FP para prever LOC. O conjunto de dados utilizados para obter os modelos são bastante diferentes e não devem ser utilizados para comparar as métricas FP e NOC, pois esse não é o objetivo do experimento aqui descrito. Mas as diferentes características das bases pode explicar as diferenças obtidas com as duas redes. Os dados fornecidos pelo ISBSG incluem informações de diferentes projetos, de diversas organizações, relacionados a diferentes tecnologias, metodologias e linguagens de programação. Os dados não são homogêneos. A segunda base, utilizada por Dolado, envolve projetos desenvolvidos por times com a mesma experiência e na mesma linguagem de programação. Outro fato que pode ter influenciado é que esse modelo possui 4 variáveis de entrada, representando diferentes características dos projetos. Assim, a rede tem mais informações para poder obter uma solução mais exata.

Essas explicações também apontam alguns fatores a serem considerados para a aplicação da técnica de RN para estimar LOC. Similarmente à outras técnicas de estimação, o conjunto de dados de projetos passados influencia no modelo obtido. A qualidade dos dados está relacionada à similaridade dos projetos envolvidos e ao número de métricas coletadas. Se uma base de dados não é homogênea, provavelemente diversas características dos projetos envolvidos deverão ser consideradas para a obtenção dos modelos.

Tabela 2: Acurácia dos Modelos Obtidos

	MMRE	PRED(0.25)
Modelo 1 (FP)	0.95	0.06
Modelo 2 (NOC)	0.23	0.77

5. Conclusões

Esse trabalho apresentou resultados de um estudo que explora o uso de RN para estimar o número de LOC de um software.

Dois modelos de estimativa de LOC foram obtidos a partir de duas bases de dados diferentes, utilizando-se respectivamente as métricas FP e NOC. O modelo obtido para NOC se mostrou bastante bom, segundo as medidas de acurácia utilizadas. O mesmo não aconteceu para o modelo utilizando FP. Como o objetivo desse estudo não foi o de comparar FP e NOC, e diferentes bases de dados foram utilizadas, esse resultado não implica que NOC é melhor que FP para predizer NOC.

Entretanto, analisando-se as características das bases e os modelos obtidos pode-se resumir alguns pontos negativos e positivos observados no uso de RN no contexto de estimativa de tamanho de software, além de outros fatores a serem considerados.

A técnica de RN apresenta a vantagem de poder trabalhar com diferentes variáveis de entrada, como é o caso do modelo para NOC, e de realmente descobrir o melhor modelo para os dados existentes. Diferentemente de outros métodos, tais como os estatísticos, não é necessário pré-definir o formato para a equação. Uma desvantagem observada foi a dificuldade e o esforço inicial para inicializar a rede, o que requer conhecimento básico sobre redes neurais.

A diferença obtida entre os modelos é reflexo das características das bases utilizadas e mostram alguns pontos a serem considerados na obtenção de modelos e que deverão ser avaliados em estudos futuros. Quanto mais homogêneos forem os projetos registrados nas bases, melhor o modelo irá refletir esses dados e uma melhor acurácia será obtida. Entretanto, existe atualmente um esforço da comunidade científica em manter bases de dados heterogêneas, coletadas de diferentes organizações. Para a utilização de RN nessas bases, é importante considerar diferentes características dos projetos, ou seja, diferentes variáveis de entrada. Além disso, é muito importante que as bases sejam completas e consistentes para que os modelos obtidos possam ser efetivamente utilizados.

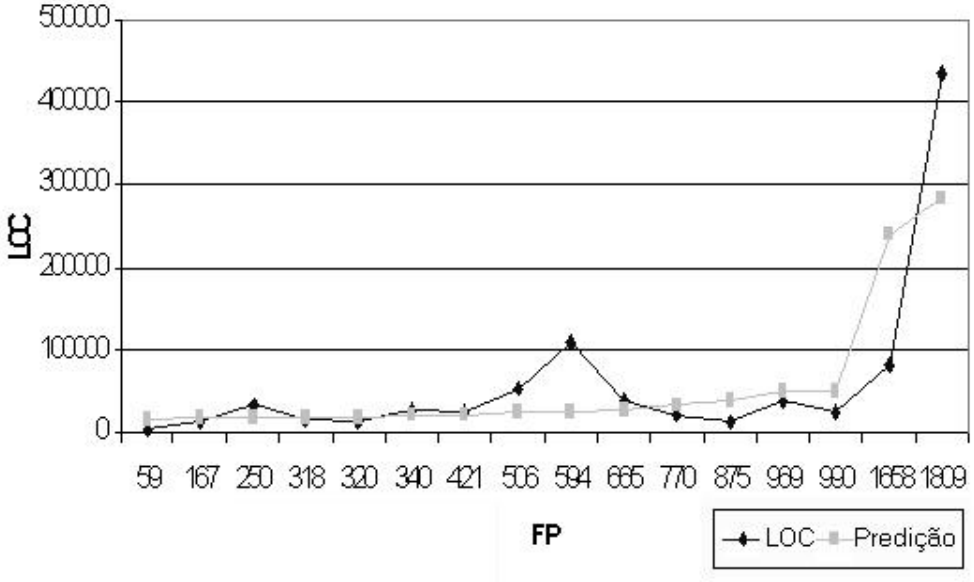

Figura 3: Modelo LOCxFP

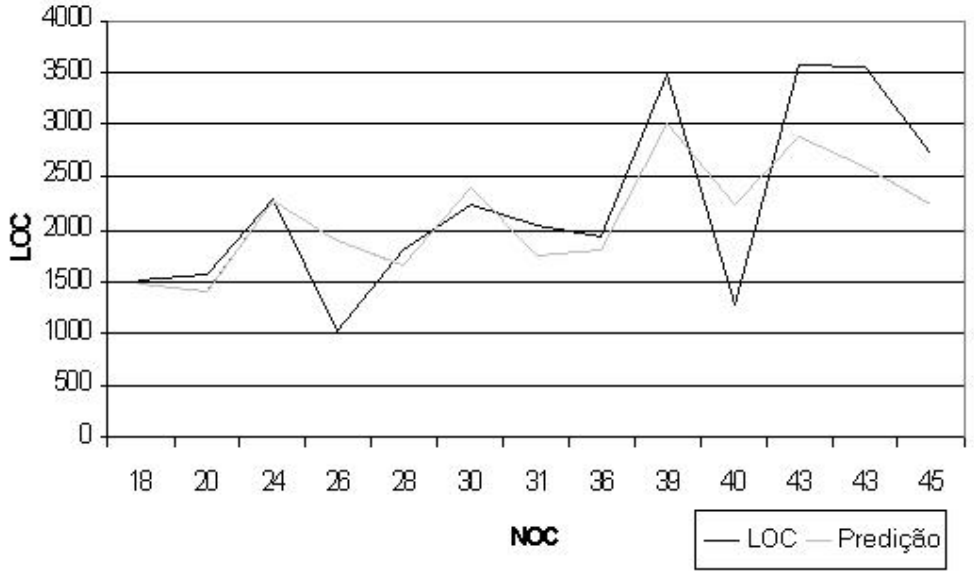

Figura 4: Modelo LOCxNOC

Referências

[1] Albrecht, A.J. *Measuring Application Development Productivity*. Proc. IBM Application Development Symposyum. Montery, CA, October, (1979) 83-92.

[2] Aguilar-Ruiz, J.S., Ramos, I., Riquelme, J.C., Toro, M. An Evolutionary Approach to Estimating Software Development Projects. *Information and Software Technology*, 43, (2001), 875-882.

[3] Burgess, C.J., Lefley, M. Can genetic programming improve software effort estimation? A comparative evaluation. *Information and Software Technology*, 43 (2001), 863-873.

[4] Cockcroft, S. Estimating CASE Development Size from Outline Specifications. *Information and Software Technology*, 38, (1996) 391-399.

[5] Dolado, J.J. A Validation of the Component-Based Method for Software Size Estimation. *IEEE Trans. on Soft. Engineering*. vol 25(10), October, (2000) 1006-1021

[6] Dolado, J.J. On the problem of the software cost function. *Information and Software Technology*, 43, (2001), 61-72.

[7] Finnie, G.R., Witting, G.E., Desharnais, J-M. A comparison of software effort estimation technique4s using function points with neura l networks, case based reasoning and regression models. *Journal of Systems Software*, 39, (1997) 281-289.

[8] International Software Benchmarking Standards Group Web Site. http://www.isbsg.org.au, December (2002)

[9] Kirsopp, C., Shepperd, M., Hart, J. Search Heuristics, Case-Based Reasoning and Software Project Effort Prediction. *Proc. GECCO*, (2002) 1367-1374.

[10] Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., Webster, S. An investigation of machine learning based prediction systems. *Journal of Systems Software*, 53(1), (2000) 23-29.

[11] Pressman, R.B *Software Engineering: A Practitioner's Approach*. McGraw-Hill, Fifth Edition, (2000).

[12] Putnam, L. and Myers, W. *Measures for Excellence*. Yourdon Press, 1982.

[13] Verner, J. , Tate, G. A Software Size Model. *IEEE Trans. on Soft. Engineering*. 18(4), April, (1992) 265-278.

[14] Working Group Neural Networks and Fuzzy Systems. Neural Network Training Tool. http://fuzzy.cs.uni-magdeburg.de/ borgelt/software.html