ESTIMACIÓN DEL ESFUERZO DE UN PROYECTO SOFTWARE UTILIZANDO EL CRITERIO MDL-EM Y COMPONENTES NORMALES N-DIMENSIONALES. APLICACIÓN A UN CASO PRÁCTICO.

Miguel Garre Rubio, Mario Charro Cubero

Ejército del Aire
Ministerio de Defensa
Departamento de Ciencias Aplicadas, Escuela de Técnicas Aeronáuticas
Base Aérea de Torrejón. 28850-Torrejón de Ardoz. Madrid. España

E-Mail: {mgarre,mchacub}@ea.mde.es

Abstract: Parametric Software Estimation Models that use a single mathematical relation exhibit poor predictive quality in the case of using large and non-homogeneous historical project databases. This is due to the fact that it is difficult to capture in a single model the variety of project characteristics actually present in the database. Segmented Models are an alternative that partitions the input space in segments according to some process of clustering of historical data. In this paper, the use of a tailored version of the EM algorithm is described as an overall approach to derive estimation models. Besides, a practical application of this algorithm over the ISBSG database will be shown.

Resumen: Los Modelos de Estimación de Parámetros Software que proporcionan una sencilla relación matemática, muestran una escasa calidad predictiva en el caso de usar grandes bases de datos históricas de proyectos no homogéneos. Esto se debe a que es difícil capturar en un único modelo la gran variedad de características de los proyectos presentes en estas bases de datos. Los Modelos Segmentados ofrecen una alternativa que consiste en dividir, mediante procedimientos de clustering, los proyectos de una base de datos de proyectos históricos en grupos de elementos afines. En este artículo, se describirá la implementación particular del algoritmo de clustering EM, adaptada a unas necesidades particulares de experimentación, con el que se obtendrán diferentes segmentos sobre los datos de estudio. Se mostrará una aplicación práctica de este algoritmo sobre la base de datos ISBSG.

Palabras Clave: Ingeniería del Software, Estimación del Esfuerzo, Clustering, EM.

1. Introducción

Para realizar la estimación del esfuerzo de realización de un proyecto software se debe de disponer de información histórica que proporcione unas bases de partida, para ello se utilizará la información contenida en una base de datos de proyectos. Cada proyecto se define mediante una serie de atributos, tales como puntos de función, esfuerzo de trabajo, plataforma de desarrollo, tipo de proyecto, tiempo de desarrollo, número de integrantes del equipo de trabajo, si se ha utilizado una metodología o una herramienta case o no, etc. Lo que se persigue es aprender una cierta función de manera que conocidos una serie de atributos se puedan obtener otros desconocidos.

Los métodos de estimación del esfuerzo basados en

Revista de Procesos y Métricas de las Tecnologías de la Información (RPM) ISSN: 1698-2029
VOL. 2, Nº 1, Marzo 2005, 13-24
Asociación Española de Sistemas de Informáticos (AEMES)

la utilización de técnicas estadísticas aplicadas a bases de datos de proyectos históricas, proporcionan ecuaciones matemáticas en las que la variable dependiente es el esfuerzo o el tiempo, y las variables independientes son diferentes aspectos del proyecto o del producto o de ambos. Estas ecuaciones son como por ejemplo las que utilizan la función potencia del tipo $e = as^b$, donde e es el esfuerzo estimado y s alguna medida del tamaño del proyecto.El utilizar una única ecuación de este tipo para toda la base de datos de proyectos, donde los proyectos son heterogéneos, proporciona unos resultados muy pobres. Por ejemplo, utilizando la herramienta Reality de la base de datos de proyectos ISBSG versión 8 (Internacional Software Benchmarking Standard Group[1]), aplicada a 709 proyectos, se obtiene la siguiente ecuación:

$$e = 47.73s^{0.76}$$

donde el esfuerzo se expresa en horas, y el tamaño en puntos de función. Un análisis de la bondad del ajuste nos da como resultado MMRE=1.18 y PRED(.30)=25.6. Ambas medidas son difícilmente aceptables, dado el alto grado de desviación sobre la inmensa mayoría de los datos. Algunos autores [**?**, **?**] han sugerido que la segmentación de los datos contenidos en las bases de datos históricas podría ser un camino adecuado para la obtención de ecuaciones matemáticas que proporcionen una mayor exactitud en las estimaciones. Una forma de obtener mejores ajustes consiste en utilizar algoritmos de agrupamiento *(clustering)* conocidos, para dividir el dominio de proyectos [**?**].

Existen gran cantidad de algoritmos de clustering [**?**]. Establecer una clasificación de ellos no es algo sencillo. Por ejemplo podemos destacar la establecida por Berkhin [**?**], el cual establece la siguiente clasificación:

1. Métodos jerárquicos.
   - *a*) Algoritmos Aglomerativos.
   - *b*) Algoritmos Divisivos.
2. Métodos de particionado.
   - *a*) Algoritmos de recoloación.
   - *b*) Clustering Probabilístico (*Finite Mixture Models*).
   - *c*) Método de los k-vecinos.
   - *d*) Método de las k-medias.
3. Algoritmos basados en densidad.
4. Métodos basados en cuadrículas.
5. Métodos basados en co-ocurrencias de datos categóricos.
6. Clustering basado en restricciones.
7. Algoritmos de clustering usados en Machine Learning.
   - *a*) Gradiente descendente y Redes Neuronales Artificiales.
   - *b*) Algoritmos genéticos.
8. Algoritmos de clustering escalable.
9. Algoritmos para datos de grandes dimensiones.

[1]http://www.isbsg.org/

En el presente trabajo se modelarán los datos mediante clustering probabilístico Finite Mixture Models [**?**] (FM), ya que proporciona muy buenos resultados, además de ser fácilmente interpretables. El resto del artículo se organiza de la siguiente manera. En la sección **??** se describirán los *Finite Mixture Models*. En la sección **??** se mostrará el proceso de obtención de clusters utilizando una vesión del algoritmo EM implementado en C. En la sección **??** se presentarán los resultados de aplicar el algoritmo EM sobre la base de datos de proyectos ISBSG. Finalmente en la sección **??** se verán las conclusiones obtenidas y futuros trabajos.

2. Finite Mixture Models

Estos modelos se usan en problemas de estimación de Funciones de Densidad de Probabilidad (FDP)[**?**, **?**] de forma semi-paramétrica, lo cual se puede extrapolar a tareas de clustering. La FDP desconocida a la que pertenecen el conjunto completo de datos, se puede aproximar mediante una combinación lineal de *NC* componentes, definidas a falta de una serie de parámetros $\{\Theta\} = \cup\{\Theta_j \forall j = 1..NC\}$, que son los que hay que averiguar,

$$P(x) = \sum_{j=1}^{NC} \pi_j p(x; \Theta_j) \text{ con } \sum_{j=1}^{NC} \pi_j = 1 \qquad (1)$$

donde π_j son las probabilidades *a priori* de cada cluster cuya suma debe ser 1, que también forman parte de la solución buscada, $P(x)$ denota la FDP arbitraria y

Revista de Procesos y Métricas de las Tecnologías de la Información (RPM) ISSN: 1698-2029
VOL. 2, Nº 1, Marzo 2005, 13-24
Asociación Española de Sistemas de Informáticos (AEMES)

$p(x;\Theta_j)$ la función de densidad del componente j. Cada cluster se corresponde con las respectivas muestras de datos que pertenecen a cada una de las densidades que se mezclan. FM se puede utilizar en un marco general para estimar FDP de formas arbitrarias, utilizándose FDP normales n-dimensionales, t-Student, Bernoulli, Poisson, y log-normales.

El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el *likelihood* de los datos. Se trataría entonces de estimar los parámetros buscados Θ, maximizando este likelihood (este criterio se conoce como *ML-Maximun Likelihood*). Normalmente, lo que se calcula es el logaritmo de este likelihood, conocido como *log-likelihood* ya que es más fácil de calcular de forma analítica. La solución obtenida es la misma, gracias a la propiedad de monotonicidad del logaritmo. La forma de esta función log-likelihood es:

$$L(\Theta,\pi) = log \prod_{n=1}^{NI} P(x_n) \tag{2}$$

donde NI es el número de instancias, que suponemos independientes entre si.

En este proceso existen dos aspectos a resolver:

1. ¿Cómo se deben calcular los parámetros de manera que se maximice la función L? Más adelante se hablará de ello.

2. Otro aspecto a tener en cuenta es ¿cómo estimar el número de componentes (número de clusters, en nuestra aplicación) de esta combinación? Esta cuestión es más difícil de resolver y para ello se han propuesto varias técnicas [**?**, **?**, **?**, **?**]

En la implementación realizada del algoritmo EM se utilizará el método *Cross-Validation-EM (Expectation-Maximization)* [**?**, **?**, **?**] para estimar los parámetros buscados Θ , y se utilizará el criterio *Minimum Description Length* [**?**] (MDL) para obtener el número óptimo de clusters (NC) y por lo tanto el número más adecuado de parámetros. Se debe tener en cuenta que cuantos más parámetros se tengan mayor *overfitting* aparecerá.

Respecto a $p(x;\Theta)$, se sabe que su elección admite varias alternativas, como la t-Student o la distribución log-normal, aquí se utilizarán distribuciones Normales N-dimensionales (N se corresponde con el número de atributos)

$$p(x;\Theta) \rightarrow f(x;\mu,\Sigma) = \frac{1}{|\Sigma|^{1/2}\,(2\pi)^{N/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{3}$$

por ser una de las más utilizadas, ya que la mayoría de las variables aleatorias obedecen a este tipo de distribuciones.

3. Proceso de obtención de clusters mediante el algoritmo EM

El algoritmo *EM*, se nutre con unos valores iniciales que convergen, después de una serie de iteraciones, a un máximo local de la función *L*. Las medias iniciales de cada atributo para cada cluster, las varianzas y covarianzas iniciales de los atributos para cada cluster y las probabilidades a priori iniciales de cada cluster no se pueden generar a la ligera ya que si no se calculan de una forma apropiada, la convergencia del algoritmo se ve comprometida y habría que recurrir a técnicas de regularización [**?**]. En esta implementación se ha elegido una metodología, que debido a la naturaleza de los datos, da buenos resultados.

Si se utilizan todos los datos disponibles para aplicar el algoritmo *EM* sobre ellos, se obtendría un modelo final que se ajustaría lo mejor posible al valor de dichos datos, lo que se conoce como *model fit*. Este *model fit* tiene la ventaja de que proporciona un modelo que encaja bastante bien los datos tomados, mientras que si se desea probar nuestro modelo con nuevos valores los resultados serán bastante insatisfactorios *(overfitting)*. Una aproximación que soluciona el problema anterior consiste en aplicar el algoritmo *EM* sobre conjuntos disjuntos de entrenamiento y de test de los datos de partida, de manera que podamos obtener un modelo que no solamente encaje bien los datos originales, sino que también sea capaz de realizar buenas predicciones. La técnica que realiza estos pasos de forma adecuada se conoce como *cross-validation*.

Se utilizará, en concreto, *v-fold cross-validation*, para, fijado un número de clusters, encontrar el conjunto de parámetros, mediante el algoritmo *EM*, que es capaz de conseguir *mejores predicciones*. Una vez hecho esto para una serie de números de cluster, se seleccionará aquel que *mejor ajuste* todas las instancias, pero penalizando un elevado número de clusters, mediante un término que monitorice esta situación, lo cual se consigue mediante el criterio *MDL*.

Finalmente se obtendrá un conjunto de clusters que

agrupan el conjunto de proyectos original. Cada uno de estos cluster estará definido por los parámetros de una distribución normal.

Para poder realizar pruebas sobre los clusters obtenidos se pueden hacer dos cosas:

1. Estimar, mediante cualquier herramienta estadística, las rectas de regresión de cada uno de los clusters.
2. Utilizar los parámetros de las diferentes distribuciones normales que definen los clusters.

Ya sea de una forma u otra los resultados son prácticamente idénticos. En este artículo se utilizarán los valores de PRED(.3) obtenidos utilizando los parámetros de las correspondientes distribuciones de probabilidad.

4. Caso estudio

En este estudio se van a utilizar los atributos Summary Work Effort (Esfuerzo, e), Function Points (Puntos de Función, fp) y Project Elapsed Time (Tiempo de desarrollo, t). Los dos primeros han sido utilizados en estudios anteriores [**?**, **?**], sin embargo la introducción del tiempo empleado en el desarrollo de un proyecto no se había planteado hasta ahora en este marco que nos ocupa. Con este estudio se pretende analizar qué ocurre al aumentar el número de atributos, y más aún qué influencia tiene el atributo tiempo en la forma y características de los clusters obtenidos. A continuación se explicará brevemente la preparación previa que es necesario realizar sobre los datos, para a continuación detallar el proceso seguido así como ofrecer los resultados del estudio llevado a cabo.

4.1. Preparación de los datos

Se utilizó la base de datos de proyectos ISBSG-8, la cual contiene información sobre 2028 proyectos. Esta base de datos contiene información sobre tamaño, esfuerzo, y otras características de un proyecto. El primer paso de limpieza consistió en eliminar de la base de datos todos los proyectos con valores numéricos no válidos o nulos en los campos esfuerzo ("Summary Work Effort" en ISBSG-8) y tamaño ("Function Points" en ISBSG - 8). Además todos los proyectos cuyo valor para el atributo "Recording Method" fuese distinto de *Staff Hours* también fueron eliminados. La razón es que se considera que el resto de formas de considerar el esfuerzo son subjetivas. Por ejemplo *Productive Time* es una magnitud difícil de valorar en un contexto organizativo.

Otro aspecto a tener en cuenta para la limpieza de los datos es la forma en la que se obtuvieron los diferentes valores de los puntos de función. En concreto se examinó el valor del atributo "Derived count approach", descartando todos los proyectos que no hubiesen utilizado como forma de estimar los puntos de función métodos como IFPUG, NESMA, Albretch o Dreger. Las diferencias entre los métodos IFPUG y NESMA tienen un impacto despreciable sobre los resultados de los valores de los puntos de función [**?**]. Las mediciones basadas en las técnicas Albretch no se eliminaron ya que, de hecho IFPUG es una revisión de estas técnicas. De la misma forma el método Dreger [**?**] es simplemente una guía sobre las mediciones IFPUG. Por último se procede a la eliminación de los proyectos con valores nulos para el atributo tiempo ("Project Elapsed Time"). Finalmente el estudio se realiza sobre una base de datos de 1569 proyectos.

4.2. Proceso seguido

De los proyectos seleccionados se confeccionaron dos archivos de prueba. En uno de ellos se tomaron en cuenta los atributos e, fp, y t, en otro se consideraron solamente e y fp. La razón de esta selección se debe al hecho de intentar ver si el introducir un nuevo atributo, tal como el tiempo, ofrece unos mejores resultados de PRED con respecto a los correspondientes valores obtenidos si no se tuviera en cuenta ese atributo. Si es así se llegaría a la conclusión de que efectivamente el atributo tiempo tiene un peso significativo en la formación de los clusters, y habría que tenerlo en cuenta en estudios futuros.

Si no fuese este el caso se tendría que suponer que el tiempo no ofrece mayor información al proceso de segmentación. Es por ello por lo que tenemos que obtener valores para los casos en que se tienen en cuenta los atributos e-fp-t y e-fp. Según sean los resultados obtenidos se podrán corroborar o no las hipótesis anteriormente manejadas.

4.3. Resultados obtenidos

Al aplicar el algoritmo EM sobre cada uno de los archivos de prueba se obtuvieron los siguientes resultados:

1. Archivo formado por los atributos e, fp y t. Se ob-

tuvieron 8 clusters, segmentos o grupos de proyectos, como queramos llamarlos. Los datos de cada uno de ellos son los dados a continuación en las tablas 1, 2 y 3.

	CLUS 1	CLUS 2	CLUS 3
Nº Proyectos	215	220	442
Probabilidad	0.1290	0.1326	0.2655
Media e	432.89	995.91	1962.25
Media fp	102.99	86.86	228.63
Media t	2.61	6.41	5.84
Desv Stand e	225.44	495.61	1086.53
Desv Stand fp	55.72	36.89	101.005
Desv Stand t	1.17	3.61	2.54
Coef Cor e-fp	0.3935	0.4260	-0.2645
Coef Cor e-t	0.4192	0.0043	0.1173
Coef Cor fp-t	0.2652	0.0586	-0.0508
PRED(.3)	46 %	58 %	44 %

Tabla 1: Parámetros de los clusters 1, 2 y 3. Archivo e-fp-t.

	CLUS 4	CLUS 5	CLUS 6
Nº Proyectos	158	246	171
Probabilidad	0.1209	0.1560	0.1119
Media e	2747.08	5386.72	10354.16
Media fp	685.38	324.52	1063.79
Media t	6.83	11.17	13.44
Desv Stand e	1565.10	2937.52	6326.76
Desv Stand fp	325.10	147.31	559.22
Desv Stand t	2.72	4.70	6.23
Coef Cor e-fp	0.0257	0.2100	-0.2228
Coef Cor e-t	0.3166	-0.2608	-0.0306
Coef Cor fp-t	0.2885	-0.1285	-0.3740
PRED(.3)	45 %	46.7 %	41 %

Tabla 2: Parámetros de los clusters 4, 5 y 6. Archivo e-fp-t.

	CLUS 7	CLUS 8
Nº Proyectos	99	20
Probabilidad	0.0699	0.0138
Media e	28790.41	154196.84
Media fp	2151.56	7303.44
Media t	21.98	29.23
Desv Stand e	19739.98	194262.02
Desv Stand fp	1421.64	5650.37
Desv Stand t	12.12	20.76
Coef Cor e-fp	0.1804	-0.4230
Coef Cor e-t	0.0309	0.0229
Coef Cor fp-t	-0.2192	-0.4071
PRED(.3)	38.38 %	15 %

Tabla 3: Parámetros de los clusters 7 y 8. Archivo e-fp-t.

2. Archivo formado por los atributos e, fp. Se obtuvieron 9 clusters. Los datos de cada uno de ellos son los dados a continuación en las tablas 4, 5 y 6.

	CLUS 1	CLUS 2	CLUS 3
Nº Proyectos	74	249	311
Probabilidad	0.046	0.1569	0.2655
Media e	284.68	616.15	0.179
Media fp	48.24	95.95	175.26
Desv Stand e	128.96	309.55	594.94
Desv Stand fp	20.46	39.27	76.67
Coef Cor e-fp	0.5931	-0.2218	-05015
PRED(.3)	63 %	50 %	59.8 %

Tabla 4: Parámetros de los clusters 1, 2 y 3. Archivo e-fp.

	CLUS 4	CLUS 5	CLUS 6
Nº Proyectos	343	161	170
Probabilidad	0.2022	0.1198	0.1173
Media e	2367.9	3072	6679.83
Media fp	278.8	714.9	317.82
Desv Stand e	1063.23	1590.45	2490.04
Desv Stand fp	128.86	302.13	141.02
Coef Cor e-fp	-0.5213	-0.1	0.3625
PRED(.3)	63.5 %	47.2 %	74.7 %

Tabla 5: Parámetros de los clusters 4, 5 y 6. Archivo e-fp.

	CLUS 7	CLUS 8	CLUS 9
Nº Proyectos	155	87	20
Probabilidad	0.1027	0.0621	0.0133
Media e	10984.97	31144.76	161796.42
Media fp	1123.3	2356.22	7509.93
Desv Stand e	6226.94	18749.39	194440.38
Desv Stand fp	540.69	1383.26	5647
Coef Cor e-fp	-0.3855	0.0069	-0.4755
PRED(.3)	42 %	51.7 %	10 %

Tabla 6: Parámetros de los clusters 7, 8 y 9. Archivo e-fp.

A pesar de que en un primer momento se podría haber pensado que introducir un nuevo atributo, en este caso el tiempo, podría mejorar los clusters obtenidos, se observa a partir de los resultados ofrecidos por el algoritmo EM que esto no es así. Con el archivo e-fp-t se obtienen unos valores medios de PRED(.3) de 41.75 %. Con el archivo e-fp se obtienen unos valores medios de

PRED(.3) de 51.2 %. De ello se desprende que el introducir el atributo tiempo no es significativo, y que se puede obviar.

Observando los valores de los coeficientes de correlación entre fp y t, se puede concluir la dependencia existente entre ellos, indicando que si se dispone de valores de fp, se puede encontrar una ecuación de fp con respecto a t que permita calcular t. Es decir, que si utilizamos fp, en cierta manera se está teniendo en cuenta t, y no es necesario utilizarlo. Más aún, de forma indirecta queda manifiesta la relación existente entre e y t. Si tratamos con e, no es necesario tener en cuenta t.

5. Conclusiones

Usar el atributo t no introduce mejora alguna, quizás debido a su dependencia con respecto al otro atributo fp y con e, como se ha comentado al final del apartado anterior.

Esto no quiere decir que no haya que seguir buscando nuevos atributos significativos, sí que habrá que hacerlo. Lo único que indica es que habrá atributos, como en este caso t, que no sean significativos por depender de otros que se usan (e y fp). Futuros trabajos se dedicarán a buscar atributos de verdadero peso en la formación de grupos de proyectos lo más adecuados posibles.

Algo que se propone como futuro trabajo es aplicar el algoritmo EM sobre los archivos con los atributos fp-t y e-t. En base a los resultados obtenidos, es posible que mediante la transitividad existente, a partir de las relaciones anteriores, entre fp-e se puedan ofrecer otro tipo de resultados. Queda abierto por tanto el seguir en esta línea de investigación.

Referencias

[Abran 2003] Abran, A. (2003). Software Estimation: Black Box or White Box. Presentado en el *Workshop ADIS 2003*.

[Reifer *et al.* 1999] Reifer, D., Boehm, B., Chulani, S. (1999). *The Rosetta Stone. Making COCOMO 81 Estimates Work with COCOMO II*. CrossTalk, 12(2), 11-15.

[Garre *et al.* 2004] Garre, M., Cuadrado, J.J. and Sicilia, M.A. (2004). Recursive segmentation of software projects or the estimation of development effort. In *Proceedings of the ADIS 2004 Workshop on Decision Support in Software Engineering*, CEUR Workshop proceedings Vol. 120, disponible en http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-120/

[Ian y Eibe 1999] Ian H. Witten y Eibe Frank. *Data Mining, Practical Machine Learning Tools and techniques with Java Implementations*. Morgan Kaufmann Publishers, San Francisco, California, 1999.

[Berkhin 2002] Pavel Berkhin. *Survey of Clustering Data Mining Techniques*. Accrue Software, Inc. (2002).

[McLachlan y Peel 2000] G. J. McLachlan y D. Peel. *Finite Mixture Models*. Wiley, New York, NY, (2000).

[Archambeau *et al.* 2003] C. Archambeau, J. A. Lee y M. Verleysen. *On the Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures* . European Symposium on Artificial Neural Networks, Brujas (Bélgica), 99-106, (2003).

[Archambeau *et al.* 2004] C. Archambeau, F. Vrins y M. Verleysen. *Flexible and Robust Bayesian Classification by Finite Mixture Models*. European Symposium on Artificial Neural Networks, Brujas (Bélgica), 75-80, (2004).

[Jain *et al.* 2000] A. K. Jain, R. P. W. Duin y J. Mao. *Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37, (2000).

[Redner y Walker 1984] E. Redner y H. Walker. *Mixture Densities, Maximum Likelihood and the EM Algorithm*. SIAM Review, 26(2), (1984).

[Mitchell 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill. (1997).

[Hartley 1958] H. Hartley. *Maximum Likelihood Estimation from Incomplete Data Sets*. Biometrics, 14:174-194, (1958).

[Dempster *et al.* 1977] A. Dempster, N. Laird y D. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, Series B, 39(1):1-38, (1977).

[McLachlan y Krishnan 1997] G. McLachlan y T. Krishnan. *The EM Algorithm and Extensions*. Wiley series in probability and statistics, John Wiley & Sons, (1997).

[Rissanen 1983] J. Rissanen. *A Universal Prior for Integers and Estimation by Minimum Description Length*. Annals of Statistics, 11(2):417-431, (1983).

[Cuadrado *et al.* 2004] J. Cuadrado Gallego, Daniel Rodríguez, Miguel Ángel Sicilia. *Modelos Segmentados de estimación del esfuerzo de desarrollo del software: un caso de estudio con la base de datos ISBSG*. Revista de Procesos y Métricas de las Tecnologías de la Información (RPM). VOL. 1, Nº 2, Agosto 2004, 25-30 ISSN: 1698-2029.

[NESMA 1996] NESMA (1996). *NESMA FPA Counting Practices Manual* (CPM 2.0).

[Dreger y Brian 1989] Dreger, J. Brian. *Function Point Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1989.