Análisis de partidas de ajedrez mediante Data Mining

|  |  |
| --- | --- |
| Rubén Márquez Minería de datos Universidad de Castilla-La Mancha [Ruben.Marquez@alu.uclm.com](mailto:Ruben.Marquez@alu.uclm.com) | Alberto Velasco Minería de datos Universidad de Castilla-La Mancha [Alberto.Velasco1@alu.uclm.com](mailto:Alberto.Velasco1@alu.uclm.com) |
| Diego Pedregal Minería de datos Universidad de Castilla-La Mancha [Diego.Pedregal@alu.uclm.com](mailto:Diego.Pedregal@alu.uclm.com) | Antonio Manjavacas Minería de datos Universidad de Castilla-La Mancha [Antonio.Manjavacas@alu.uclm.com](mailto:Antonio.Manjavacas@alu.uclm.com) |

1 Introducción

La minería de datos se define como un proceso utilizado para extraer, analizar e identificar patrones a partir de conjuntos masivos de datos. Un área donde esta disciplina ha alcanzado una gran relevancia en las últimas décadas ha sido en el contexto del ajedrez, donde, desde los primeros casos como *DeepBlue* [1], se han elaborado múltiples sistemas de inteligencia artificial basados en el análisis de partidas con distintos fines: jugar contra humanos, jugar contra otras máquinas, predecir resultados, analizar partidas, etc.

En base a la utilidad de la minería de datos en el contexto del ajedrez, hemos decidido aplicar técnicas de extracción de conocimiento sobre datos de partidas disponibles en la web.

En este documento se definirán los aspectos iniciales del problema a abordar: fuente de datos, descripción de estos y objetivos que se pretenden conseguir. Posteriormente, se abordará el proceso de recopilación y tratamiento de los datos, para finalmente concluir con los resultados obtenidos en base a los objetivos propuestos.

Tanto los datos como el código empleados son de libre acceso y pueden encontrarse en el siguiente repositorio: <https://github.com/manjavacas/Data-Mining>.

2 Fuente de datos

Para realizar este proyecto se ha elegido como fuente de datos el conjunto de partidas de ajedrez de la web *lichess.org* [2] correspondientes al mes de **septiembre de 2019**. De este conjunto de partidas se han obtenido las partidas correspondientes a los 200 mejores jugadores de la clasificación en modo clásico.

La extracción selectiva de los datos por fecha y mejores jugadores se ha llevado a cabo haciendo uso de la API berserk [3]. Todos los datos recopilados son de dominio público sin derechos reservados.

3 Descripción de los datos

El formato de cada uno de los registros del dataset es el siguiente:

* Event: evento en el cual se enmarca la partida.
* Site: enlace a la partida.
* White: jugador con blancas.
* Black: jugador con negras.
* Result: resultado de la partida.
* UTCDate: fecha UTC.
* UTCTime: hora UTC.
* WhiteElo: puntuación ELO del jugador con blancas.
* BlackElo: puntuación ELO del jugador con negras.
* WhiteRatingDiff: puntos obtenidos por blancas.
* BlackRatingDiff: puntos obtenidos por negras.
* WhiteTitle, BlackTitle: abreviación de la titulación de los jugadores (según estándar FIDE).
* ECO: código estándar de la apertura utilizada.
* Opening: apertura utilizada.
* TimeControl: control de tiempo utilizado.
* Termination: motivo de victoria.
* Lista de jugadas incluyendo evaluación y tiempo de reloj para cada movimiento.

Aunque estos son los campos que se recogen en la mayoría de los datasets en ajedrez, existen campos adicionales permitidos por el formato PGN (Portable Game Notation) utilizado. Información más detallada sobre cada uno de los campos puede consultarse en <http://www.saremba.de/chessgml/standards/pgn/pgn-complete.htm#c9.1.1>.

4 Formulación de hipótesis

Partiendo del conjunto de datos elegido, se pretenden contrastar las siguientes hipótesis:

***H1***) Es posible pronosticar el resultado de un conjunto de partidas de ajedrez en base a la información disponible acerca de los jugadores involucrados.

***H2***) Es posible definir los factores de éxito que influyen en el devenir de una partida de ajedrez entre dos jugadores.

***H3***) Es posible definir el estilo de juego de un jugador en base a su histórico de partidas jugadas.

5 Objetivos

En base a este conjunto de datos, se pretenden abordar los siguientes objetivos aplicando las diferentes fases del **proceso KDD** [4]:

1. Caracterizar a los jugadores en base su histórico de partidas estableciendo un perfil de juego para cada uno de ellos (en base a su agresividad, aperturas, etc.).
2. Tratar de pronosticar el resultado de un conjunto de partidas entre unos jugadores predefinidos.

6 Target data

De cara a llevar a cabo el contraste de nuestras hipótesis, se procedió con la definición del conjunto de datos objetivo (“tarjeta de datos” o *target data*) compuesto por las siguientes características:

* USER\_ID: identificador de usuario (String).
* GAME\_LINK: enlace a la partida (String).
* ELO: puntuación ELO del jugador (int).
* COLOUR: color de piezas del usuario (String).
* OPENING: apertura (y variación) utilizada en la partida (String).
* RESULTADO: modo de finalización de la partida (int):
  + 0 = gana\_blanco
  + 1 = tablas
  + 2 = gana\_negro
* MOVEMENTS: número de movimientos de la partida (int).
* TOTAL\_TIME: duración de la partida en segundos (double).
* TOTAL\_TIME\_PLAYER: número de segundos invertido por el jugador (double).
* EARLY\_TIMES\_MEAN: tiempo medio entre movimientos en el *early game* (double).
* EARLY\_TIMES\_MEDIAN*:* mediana de los tiempos entre movimientos en el *early game* (double).
* EARLY\_TIMES\_VARIANCE*:* varianza de tiempos entre movimientos en el *early game* (double).
* EARLY\_TIMES\_MAX: máximo tiempo empleado entre movimientos en el *early game* (double).
* EARLY\_TIMES\_MIN: mínimo tiempo empleado entre movimiento en el *early game* (double).
* MID\_TIMES\_MEAN: tiempo medio entre movimientos en el *mid game* (double).
* MID\_TIMES\_MEDIAN*:* mediana de los tiempos entre movimientos en el *mid game* (double).
* MID\_TIMES\_VARIANCE*:* varianza de tiempos entre movimientos en el *mid game* (double).
* MID\_TIMES\_MAX: máximo tiempo empleado entre movimientos en el *mid game* (double).
* MID\_TIMES\_MIN: mínimo tiempo empleado entre movimiento en el *mid game* (double).
* END\_TIMES\_MEAN: tiempo medio entre movimientos en el *end game* (double).
* END\_TIMES\_MEDIAN*:* mediana de los tiempos entre movimientos en el *end game* (double).
* END\_TIMES\_VARIANCE*:* varianza de tiempos entre movimientos en el *end game* (double).
* END\_TIMES\_MAX: máximo tiempo empleado entre movimientos en el *end game* (double).
* END\_TIMES\_MIN: mínimo tiempo empleado entre movimiento en el *end game* (double).
* POINTS\_BALANCE: balance de puntos al final de la partida (int).
* TAKEN\_BALANCE: balance de piezas al final de la partida (int).
* AGGRESSIVENESS: nivel de agresividad del jugador en el rango [0,5]. Se mide a partir de los siguientes factores (double):
  + EARLY­­\_TAKEN*­*: número de piezas comidas en el *early game* (primer tercio de la partida).
    - Alto = +1
    - Medio = +0.5
    - Bajo = +0
  + AGRESSIVE\_OPENING: la apertura utilizada es agresiva para el color de piezas del jugador:
    - True = +2
    - False = +0
  + CASTLING: el jugador se enroca a lo largo de la partida:
    - True = +0
    - False = +2

6 Preproceso y normalización

Una vez extraído nuestro conjunto de datos objetivo, se procedió con su tratamiento preliminar:

- El preproceso de los datos consistió, inicialmente, en la **eliminación de partidas con campos nulos**. La existencia de dichos valores nulos se debía en la totalidad de los casos a juegos demasiado cortos, ocasionados por eventos excepcionales como la rendición de alguno de los jugadores al principio de la partida.

- Una segunda medida llevada a cabo fue la **eliminación de partidas que contenían tiempos negativos**: en *lichess*, un jugador puede ofrecer tiempo a su contrincante de forma voluntaria, provocando que algunos de los registros del dataset que almacenan la información acerca de los tiempos de la partida contengan valores negativos. Así, debido a la imposibilidad de conocer *a priori* el tiempo regalado, nos vimos obligados a prescindir de estas partidas, si bien el número de casos de estas características no resultó ser significante.

- Los campos que reflejaban el color (COLOUR) de los jugadores y la apertura (OPENING) utilizada fueron **transformados en categóricos** y pertinentemente codificados, dando lugar a los campos OPENING\_ENC y COLOUR\_ENC.

- Finalmente, se realizó una **normalización** **de los datos** haciendo uso de la clase MinMaxScaler ofrecida por la librería de Python sklearn.

7 Visualización y *clustering*

Una vez los datos fueron convenientemente preparados para su utilización, el objetivo inicial perseguido fue la caracterización de los diferentes tipos de jugadores presentes en nuestro dataset mediante agrupamiento o *clustering*. Trataríamos de identificar jugadores similares a partir de las características consideradas.

Primero, con el fin de limitar el número de características empleadas, se realizó una **reducción de la dimensionalidad** de los datos mediante PCA (*Principal Component Analysis*). De nuevo, se empleó la clase PCA que ofrece la librería sklearn para llevar a cabo este proceso. El número de componentes considerado fue 3, y los resultados obtenidos los que mostrados en las Imágenes 1 y 2.

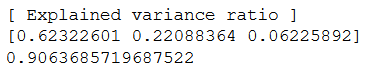


Imagen 1. Ratio de varianzas obtenido mediante PCA

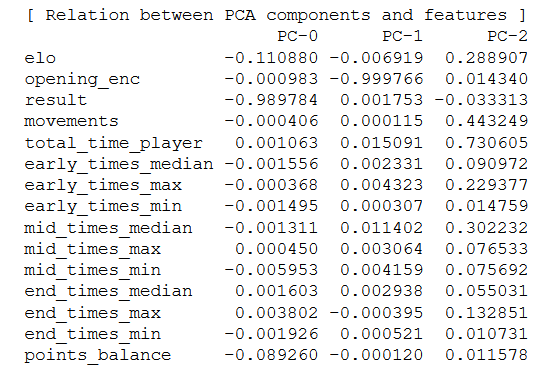


Imagen 2. PCA para la selección de componentes principales

A partir de los resultados obtenidos se consideró el uso de las variables RESULT, OPENING\_ENC y TOTAL\_TIME\_PLAYER para realizar el agrupamiento.

El algoritmo de *clustering* elegido fue *k-means*. De cara a decidir el número de *clusters* óptimo se hizo uso de los coeficientes *Distortion* (Figura 1) y *Silhouette* (Figura 2) para un intervalo de entre 1 y 35 *clusters*, decidiéndonos finalmente por un número de conjuntos K = 9 (Distortion=247.02; Silhouette=0.414).

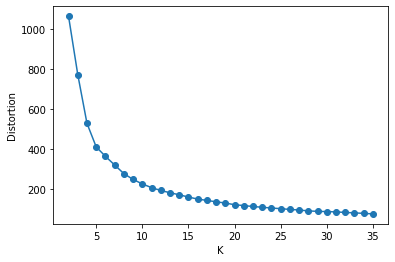


Figura 1. *Distortion* por número de *clusters*

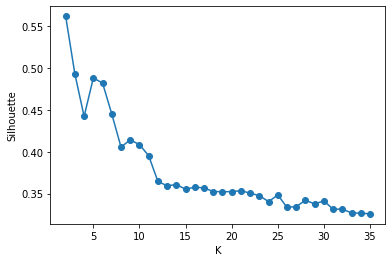


Figura 2. *Silhouette* por número de *clusters*

Los *clusters* obtenidos son los mostrados en las Figuras 3 y 4. Estos pueden interpretarse como las diferentes caracterizaciones (en términos de *apertura utilizada* y *tiempo empleado*) de los conjuntos de jugadores que han *ganado*, *perdido* o *empatado*.

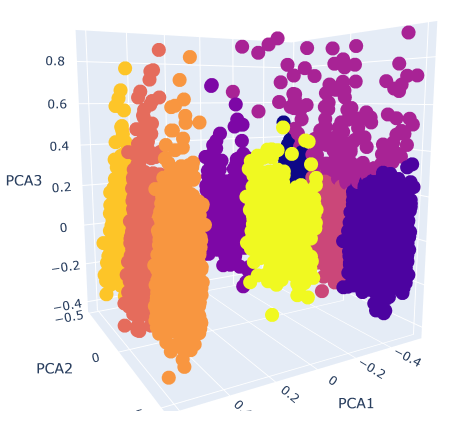


Figura 3. Visualización de los *clusters* (1)

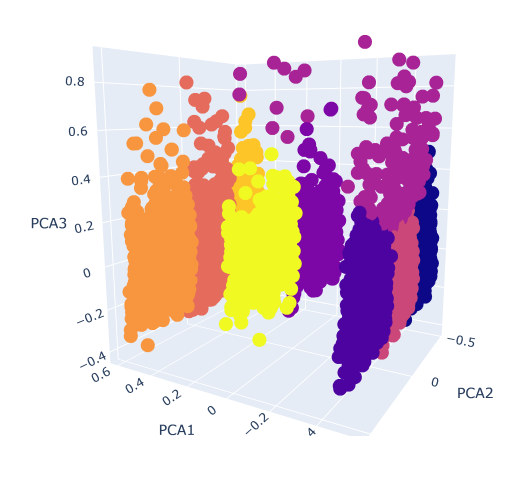


Figura 4. Visualización de los *clusters* (2)

8 Modelo de predicción

Una vez llevada a cabo la caracterización de los jugadores presentes en nuestro dataset, se procedió a la construcción del **modelo de predicción** empleado para tratar de conocer el resultado de las partidas en base a sus características.

REFERENCIAS

[1] Campbell, M., Hoane, A. J., & Hsu, F. (2002). Deep Blue. Artificial Intelligence, 134(1-2), 57–83. doi:10.1016/s0004-3702(01)00129-1

[2] <https://database.lichess.org/>

[3] https://berserk.readthedocs.io/en/master/

[2] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), 27–34. doi:10.1145/240455.240464