

EL APRENDIZAJE CON ALGORITMOS GENÉTICOS

Bonifacio de Andrés y Toro

A8.1 INTRODUCCIÓN

A continuación se describe el ejemplo propuesto en el capítulo nueve para ilustrar el proceso de aprendizaje mediante algoritmos genéticos.

El problema elegido es la optimización del proceso de fermentación de la cerveza, por medio de tres algoritmos genéticos diferentes. Para resolverlo, se ha utilizado la toolbox de Matlab EVOCOM¹.

En las siguientes secciones se describe el problema y los programas² de Matlab utilizados para resolverlos.

¹ EVOCOM se encuentra disponible, tras un sencillo proceso de registro, en la página web <http://www.dacya.ucm.es/evocom>. En la misma página web se encuentran disponibles los manuales de usuario y de referencia de la herramienta.

² Al final de este documento se presenta una tabla explicativa de todos los ficheros de Matlab proporcionados. Los ficheros “*Main.m*” recogidos en los tres directorios de código adjunto se corresponden con cada uno de los tres métodos utilizados para resolver el problema mediante algoritmos genéticos.

A6.2 DESCRIPCION DEL PROBLEMA

Vamos a utilizar un ejemplo práctico para explicar la naturaleza y solución de nuestro problema: la fermentación cervecera. Esencialmente el “*saccharomyces cerevisiae*” que se introduce dentro del recipiente de mosto, evoluciona dentro del mismo reproduciéndose y dejando como subproducto el alcohol derivado de la glucosa que contiene dicho mosto. Este proceso debe realizarse a una temperatura determinada y en unas condiciones de calidad precisas. Fundamentalmente existe un perfil de temperaturas que debe seguir el proceso durante el tiempo que dura el mismo. Empezemos por dar una breve descripción del modelo matemático de la fermentación cervecera, que puede encontrarse detalladamente en Andrés Toro, B. y col. (1997, 1998).

En todo problema de aprendizaje lo primero que hay que establecer es el modelo sobre el que se va a implementar. Nuestro ejemplo toma la biomasa activa en suspensión, en cada momento, como base del mismo. Distinguimos en el modelo dos períodos consecutivos en la fermentación: latencia y fermentación activa. Hay que tener en cuenta que en la levadura que se inocula en el mosto existen tres clases de células: latentes, activas y no viables, siendo pequeño el número de activas frente a las otras dos en el momento inicial. La figura A9.1 expresa sintéticamente los principales fenómenos que tienen lugar a lo largo de los dos períodos, y que hemos incluido en el modelo.

Durante el período de latencia existe muy poca actividad en la producción de etanol, las células latentes van pasando a activas y se produce una caída al fondo de biomasa. Describimos estos fenómenos predominantes mediante un conjunto reducido de ecuaciones diferenciales (A9.1 a A9.3).

$$\frac{dx_{activa}}{dt} = \mu_{Lat} (0.48 \cdot x_{inicial} - x_{activa}) \quad (A9.1)$$

$$\frac{dx_{lat}}{dt} = -\dot{x}_{activa} = -\mu_{Lat} \cdot x_{lat} \quad (A9.2)$$

$$\begin{aligned} \frac{dx_{susp}}{dt} &= \frac{dx_{+}}{dt} = -\mu_D \cdot x_{+} = -\mu_D \cdot (x_{susp} - (x_{activa} + x_{lat})) \\ &= -\mu_D \cdot (x_{ussp} - 0.48 \cdot x_{inicial}) \end{aligned} \quad (A9.3)$$

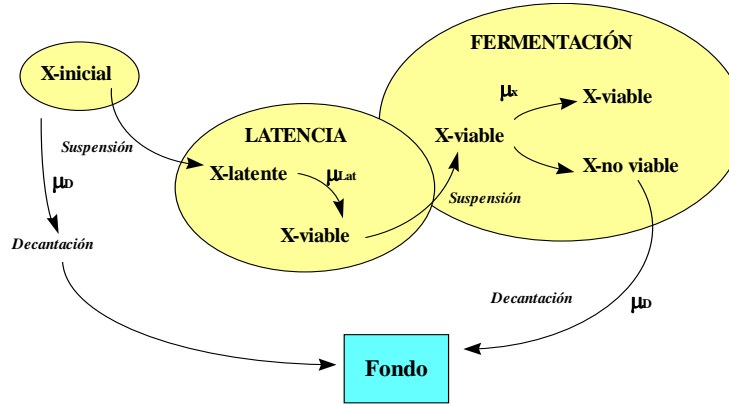


Figura A9.1 Descripción del modelo

En la fase de fermentación activa debemos considerar otros factores. La concentración de levadura activa en suspensión, crece y decrece según una trayectoria en forma de campana. Si aumentamos la temperatura, el máximo de la campana se hace más pronunciado. Sigue existiendo sedimentación, ecuaciones (A9.4) y (A9.5) si bien se ve contrarrestada por la actividad carbónica.

$$\frac{dx_{activa}}{dt} = \mu_x \cdot x_{activa} - k_+ \cdot x_{activa} + \mu_L \cdot x_{lat} \quad (A9.4)$$

$$\mu_x = \frac{\mu_{xo} \cdot s}{0.5 \cdot s_{inicial} + e} \quad (A9.5)$$

La levadura total en el fermentador es una función creciente. La evolución de la concentración de azúcares viene determinada por las ecuaciones (A9.6) y (A9.7).

$$\frac{ds_{cons}}{dt} = \mu_s \cdot x_{activa} \quad (A9.6)$$

$$\mu_s = \frac{\mu_{so} \cdot s}{ks + s} \quad (A9.7)$$

La concentración de alcohol es creciente y su evolución depende de la temperatura: incrementando la temperatura se obtiene un crecimiento más rápido,

ecuaciones (A9.8), (A9.9) y (A9.10). El ritmo de producción de etanol viene limitado por la concentración de azúcares totales iniciales.

$$\frac{de}{dt} = \mu_a \cdot f \cdot x_{activa} \quad (A9.8)$$

$$f = 1 - \frac{e}{0.5 \cdot s_{inicial}} \quad (A9.9)$$

$$\mu_a = \frac{\mu_{ao} \cdot s}{ka + s} \quad (A9.10)$$

Tanto la concentración de levadura activa, ecuación (A9.4), como la de etanol tienen inhibición en su crecimiento, ecuación (A9.9), debida precisamente al propio aumento de la concentración de etanol.

También se ha modelado la biomasa que muere y se deposita en el fondo del fermentador, ecuaciones (A9.11), (A9.12) y (A9.13),

$$\frac{dx_+}{dt} = k_+ \cdot x_{activa} - \mu_d \cdot x_+ \quad (A9.11)$$

$$\mu_D = \frac{0.5 \cdot s_{inicial} \cdot \mu_{D0}}{0.5 \cdot s_{inicial} + e} \quad (A9.12)$$

$$\frac{dx_{fondo}}{dt} = \mu_D \cdot x_+ \quad (A9.13)$$

Nuestro modelo incluye también la producción de acetato de etilo (A9.14) y de diacetil, ecuación (A9.15), que son dos importantes subproductos, cuya concentración por encima de ciertos límites es indeseable. Nos interesa conocer su evolución durante la fermentación, con el fin de poder optimizar el proceso sin degradar la calidad del resultado.

$$\frac{d(ea)}{dt} = \mu_{eas} \cdot \frac{ds}{dt} = \mu_{eas} \cdot \mu_x \cdot x_{activa} \quad (A9.14)$$

$$\frac{d(vdk)}{dt} = k_{DC} \cdot s \cdot x_{activa} - k_{DM} \cdot (vdk) \cdot e \quad (A9.15)$$

Para determinar las características y parámetros del modelo, en Andrés-Toro y col. (2001) se pueden encontrar datos relativos a lo expuesto, donde se incluyen más de 250 fermentaciones según las siguientes directrices:

a) Han sido fermentaciones isotermas a las temperaturas de 8°C, 12°C, 16°C, 20°C y 24°C para abarcar el rango de posible actuación del proceso.

b) Mediante análisis de laboratorio se han evaluado las concentraciones de levadura, maltotriosa, maltosa, glucosa, sacarosa, fructosa y etanol. En cada fermentación se ha tomado una media de 30 puntos de muestreo, siendo éstos más frecuentes en los momentos considerados críticos. Como resultado de la modelación proponemos el modelo ya mostrado para la fermentación cervecera industrial.

En estas ecuaciones, los parámetros dependen de la temperatura según funciones de Arrhenius. Este modo de expresión relativa a la fuerte influencia de la temperatura en el proceso, hace que el modelo sea no lineal.

El modelo ha sido validado experimentalmente y la correspondencia del modelo con los datos reales resulta ser bastante satisfactoria.

A6.3 SOLUCION CON ALGORITMOS GENÉTICOS

Para resolver este problema multiobjetivo mediante algoritmos genéticos se pueden utilizar diferentes técnicas de evaluación de la bondad de cada una de las soluciones, que tengan en cuenta los objetivos que hay que optimizar en este problema, y diferentes formas de codificar el problema. Se adjuntan tres propuestas diferentes, junto con su código correspondiente, que se caracterizan por:

- 1) Codificación del perfil de temperaturas a intervalos de tiempo constante, junto con una evaluación basada en la combinación aditiva de los diferentes objetivos. Esta forma de proceder se presenta en el directorio ***MonoObj***.
- 2) Codificación del perfil de temperaturas a intervalos de tiempo constante, junto con una evaluación de la bondad en la que se disponen los objetivos en diferentes niveles, y se comprueba si sus valores se encuentran dentro de los límites considerados correctos. Esta forma de proceder se presenta en el directorio ***MultiNivel***.

- 3) Codificación del perfil de temperaturas a intervalos de tiempo variable, junto con una evaluación de la bondad en la que se disponen los objetivos en diferentes niveles, y se comprueba si sus valores se encuentran dentro de los límites considerados correctos. Esta forma de proceder se presenta en el directorio *TiempoVariable*.

Todas las propuestas han sido implementadas con EVOCOM, una herramienta para el desarrollo de algoritmos evolutivos para Matlab, que se fundamenta en una clara separación entre la definición de la codificación elegida, la definición de los objetivos que hay que optimizar, y la definición de los operadores del algoritmo genético. Estas definiciones se realizan por medio de la elección de las funciones que implementan el comportamiento deseado, motivo que facilita una resolución progresiva del problema. Inicialmente suele ser suficiente con definir las funciones objetivo y utilizar las funciones de definición de la codificación y de los operadores proporcionados dentro de la herramienta. Finalmente, se pueden definir codificaciones u operadores propios, que incorporen características específicas del problema. La forma final de proceder se utiliza en las tres propuestas: sus codificaciones son específicas del problema ya que en los dos primeros casos se fija el valor de la temperatura inicial, y en el tercero se utiliza una codificación especial para representar los instantes de tiempo en los que cambia la temperatura y su valor final en cada instante. Además, en el último caso también se definen operadores de cruce y mutación específicos del problema. El primero trata el problema desde un punto de vista mono-objetivo tras combinar los valores de los diferentes objetivos, mientras que en las dos restantes se utiliza un método multiobjetivo pareto, con múltiples niveles y limitación de los valores apropiados.

En las siguientes tablas se presentan los ficheros utilizados para resolver cada uno de los problemas. En todos ellos, se resalta el nombre del programa principal en negrita.

<i>DrawFinal.m</i>	<i>Función que representa las soluciones obtenidas en diferentes iteraciones que el algoritmo genético ha ido almacenando durante su ejecución</i>
<i>GACerveza.m</i>	<i>Función en la que 1) se especifican que funciones deben ser utilizadas por EVOCOM para definir la codificación, los objetivos, y operadores, y 2) se solicita a EVOCOM que resuelva el problema con las funciones elegidas. La solución se almacena, periódicamente, en múltiples ficheros</i>
Main.m	<i>Programa principal: resuelve el problema y visualiza los resultados almacenados en los ficheros</i>
<i>gen_cerveza.m</i>	<i>Función en la que se define la codificación del problema</i>
<i>proce_basic.m</i>	<i>Función que calcula los objetivos del problema</i>

<i>proce_cerveza.m</i>	<i>Función que combina los múltiples objetivos para convertir el problema en uno mono-objetivo.</i>
<i>repair_cerveza.m</i>	<i>Función que repara aquellas soluciones que se salgan del rango correcto debido a los operadores genéticos elegidos</i>
<i>show_cerveza.m</i>	<i>Función que muestra las soluciones durante el proceso de optimización</i>

Tabla A9.1. Ficheros suministrados en el directorio **MonoObj**

<i>DrawFinal.m</i>	<i>Función que representa las soluciones obtenidas en diferentes iteraciones que el algoritmo genético ha ido almacenando durante su ejecución</i>
<i>GACerveza.m</i>	<i>Función en la que 1) se especifican que funciones deben ser utilizadas por EVOCOM para definir la codificación, los objetivos, y operadores, y 2) se solicita a EVOCOM que resuelva el problema con las funciones elegidas. La solución se almacena, periódicamente, en múltiples ficheros</i>
Main.m	<i>Programa principal: resuelve el problema y visualiza los resultados almacenados en los ficheros</i>
<i>gen_cerveza.m</i>	<i>Función en la que se define la codificación del problema</i>
<i>proce_basic.m</i>	<i>Función que calcula los objetivos del problema</i>
<i>proce_cerveza.m</i>	<i>Función que combina los múltiples objetivos para convertir el problema en uno mono-objetivo.</i>
<i>repair_cerveza.m</i>	<i>Función que repara aquellas soluciones que se salgan del rango correcto debido a los operadores genéticos elegidos</i>
<i>show_cerveza.m</i>	<i>Función que muestra las soluciones durante el proceso de optimización</i>

Tabla A9.2. Ficheros suministrados en el directorio **MultiNivel**

<i>DrawFinal.m</i>	<i>Función que representa las soluciones obtenidas en diferentes iteraciones que el algoritmo genético ha ido almacenando durante su ejecución</i>
<i>GACerveza.m</i>	<i>Función en la que 1) se especifican que funciones deben ser utilizadas por EVOCOM para definir la codificación, los objetivos, y operadores, y 2) se solicita a EVOCOM que resuelva el problema con las funciones elegidas. La solución se almacena, periódicamente, en múltiples ficheros</i>
Main.m	<i>Programa principal: resuelve el problema y visualiza los resultados almacenados en los ficheros</i>
<i>crossing_cerveza.m</i>	<i>Función que define como se realiza el cruce en el algoritmo genético</i>
<i>crosspoint_cervezafix.m</i>	<i>Función que define como se seleccionan los puntos de</i>

	<i>cruce en el algoritmo genético</i>
<i>gen_cerveza.m</i>	<i>Función en la que se define la codificación del problema</i>
<i>mutnew_cerveza.m</i>	<i>Función que define como se mutan los valores de las soluciones</i>
<i>mutsel_cerveza.m</i>	<i>Función que define que genes mutan</i>
<i>proce_basic.m</i>	<i>Función que calcula los objetivos del problema</i>
<i>proce_cerveza.m</i>	<i>Función que combina los múltiples objetivos para convertir el problema en uno mono-objetivo.</i>
<i>repair_cerveza.m</i>	<i>Función que repara aquellas soluciones que se salgan del rango correcto debido a los operadores genéticos elegidos</i>
<i>show_cerveza.m</i>	<i>Función que muestra las soluciones durante el proceso de optimización</i>
<i>super_cerveza.m</i>	<i>Función que modifica las mejores soluciones periódicamente</i>

Tabla A9.3. Ficheros suministrados en el directorio **TiempoVariable**