

Apéndice I

Código Matlab de la función que implementa el algoritmo NMF

```
function [W,H,obj,k] = nmf(V,r,max_iter,lambda)

% NMF - Non-negative matrix factorization
% [W,H,OBJ,NUM_ITER] = NMF(V,R,MAX_ITER,LAMBDA)
% V - Datos de entrada.
% r - Número de bases a obtener.
% MAX_ITER - Número máximo de iteraciones (por defecto son 50).
% LAMBDA - Tamaño del paso de convergencia (por defecto vale 0.000000001).
% W - Matriz de las bases.
% H - Matriz de codificación.
% OBJ - Valor final de la función objetivo.
% NUM_ITER - Número de iteraciones empleadas.

if nargin < 4
    lambda = 0.000000001;
end
if nargin < 3
    max_iter = 50;
end

[m,n] = size(V);
W = rand(m,r); % Se inicializa a valores aleatorios en [0,1].
W = W./(ones(m,1)*sum(W,1));
H = rand(r,n);
eps=0.0000000000000001; % Para evitar singularidades
obj_old = calcula_objetivo(V,W,H);

for k=1:max_iter
    fprintf('k=%d; obj=%f\n',k,obj_old);
    % Reglas de actualización
    W = W.*((V./(W*H+eps))*H');
    W = W./(ones(m,1)*sum(W,1));
    H = H.*(W'*(V./(W*H+eps)));

    obj = calcula_objetivo(V,W,H);
    if abs(obj-obj_old) <= lambda
        return;
    end
    obj_old = obj;
end

function [obj] = calcula_objetivo(V,W,H)
obj = sum(sum(V.*log(W*H+eps) - (W*H)));
```