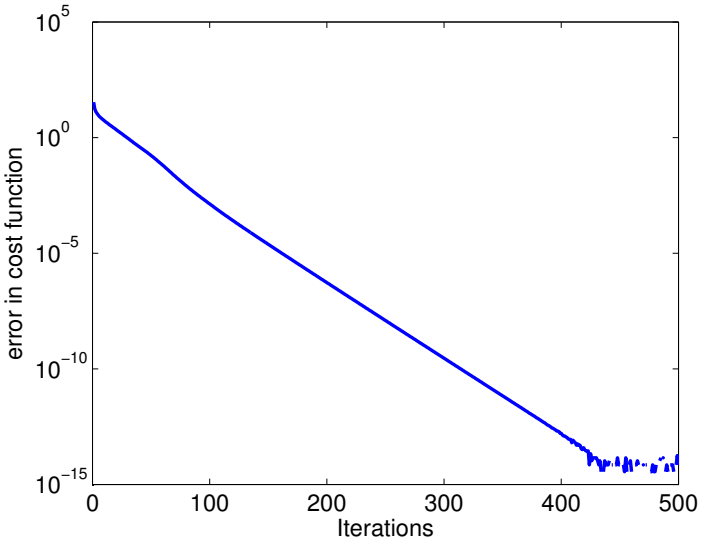
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Prof. D. Kressner
M. Steinlechner

1 ▶ ISTA: A projected proximal gradient method

```
function lasso

    rng(7);
    m = 500;
    n = 2500;
    A = randn( m, n );
    % make the columns have unit l2 norm
    A = A ./ repmat( sqrt(sum(A.^2)), m, 1 );
    % create a sparse exact solution
    x_star = zeros(n, 1);
    idx = randperm( n, 100 );
    x_star(idx) = randn( 100, 1 );
    % create corresponding right hand side
    noise = 10^(-2) * randn( m, 1 );
    b = A*x_star + noise;

    cost = @(x, gamma )0.5*norm(A*x - b)^2 + gamma*norm(x, 1);
    prox = @(x, gamma) max( abs(x) - gamma, 0) .* sign( x );
    grad = @(x) A'*(A*x - b);
    gamma = 0.1 * norm(A'*b, Inf);
    alpha = 0.1;
    cost_star = cost( x_star, gamma );
    maxiter = 500;
    x = zeros( n, 1);
    costs = cost( x, gamma );
    for k = 1:maxiter
        x(:,k+1) = prox( x(:,k) - alpha*grad(x(:,k)), gamma*alpha );
        costs(k+1) = cost( x(:,k+1), gamma );
    end

    % number of nonzero entries
    nnz_x = nnz(x(:,end))

    semilogy(abs(costs(1:end-1)-costs(end)),'linewidth',2)

    set(gca,'fontsize',14)
    xlabel('Iterations')
    ylabel('error in cost function')

end
```



2 ▶ Nonnegative Matrix Factorization (NMF)

The nonnegative rank-k approximation of a nonnegative matrix $A \in \mathbb{R}^{m\times n}$ is a statsistical tool to extract *features* from A. Typical applications are pattern recognition, recommendation systems, or spectral analysis.

The task reads

$$\min_{W\in\mathbb{R}^{m\times k},\ H\in\mathbb{R}^{k\times n}} \frac{1}{2}\|A - WH\|_F^2, \quad W \geq 0,\ H \geq 0,$$

where $\|M\|_F = (\sum_{ij} m_{ij}^2)^{1/2}$ is the Frobenius norm of a matrix. Typically, $k \ll \min(m, n)$.

(a) (i) Let

$$f\colon \mathbb{R}^{m\times k} \times \mathbb{R}^{k\times n} \to \mathbb{R},\ (W, H) \mapsto \frac{1}{2}\|A - WH\|_F^2.$$

At a point (W_0, H_0) calculate the partial gradients $\nabla_W f(W_0, H_0) \in R^{m\times k}$ and $\nabla_H f(W_0, H_0) \in \mathbb{R}^{k\times n}$.

$f(W,H) = \frac{1}{2}\|A - WH\|_F^2 = \frac{1}{2}\|A\|^2 - \langle A, WH\rangle_F + \frac{1}{2}\langle WH, WH\rangle_F$

a)i) Using the bilinearity of the inner product we obtain

$f(W_0+\delta W, H_0) = f(W_0, H_0) - \langle A, \delta W H_0\rangle_F + \langle W_0H_0, \delta W\cdot H_0\rangle_F + \frac{1}{2}\langle \delta W H_0, \delta W H_0\rangle$

Collecting the linear terms:

$f'(W_0, H_0)[\delta W] = \langle W_0H_0 - A, \delta W H_0\rangle_F$

The following rules hold for the Frobenius inner product:

$$\langle A, BC\rangle_F = \langle B^TA, C\rangle_F = \langle AC^T, B\rangle_F$$

Using the second one gives:

$f'(W_0, H_0)[\delta W] = \langle (W_0H_0 - A)H_0^T, \delta W\rangle$

$\rightarrow \underline{\nabla_W f(W_0, H_0) = (W_0H_0 - A)H_0^T}$

Similar calculation, but using the first rule:

$\underline{\nabla_H f(W_0, H_0) = W_0^T(W_0H_0 - A)}$

(ii) Calculate the exact values $\alpha_W \geq 0$ and $\alpha_H \geq 0$ which minimize

$\alpha \mapsto f(W_0 - \alpha \nabla_W f(W_0, H_0), H_0)$, and $\alpha \mapsto f(W_0, H_0 - \alpha \nabla_H f(W_0, H_0))$,

respectively.

a) ii) Let

$$g(\alpha) = f(W_0 - \alpha G, H_0)$$

$$\overset{\text{see above}}{=} f(W_0, H_0) - \alpha \langle \nabla_W f(W_0, H_0), G \rangle_F + \frac{1}{2}\alpha^2 \| G H_0 \|_F^2$$

Then

$$g'(\alpha) = - \langle \nabla_W f(W_0, H_0), G \rangle + \alpha \| G H_0 \|_F^2$$

is zero, if

$$\alpha = \frac{\langle \nabla_W f(W_0, H_0), G \rangle_F}{\| G H_0 \|_F^2}$$

and if the denominator is not zero. In this case, α also represents the unique global minimum.

In the problem, $G = \nabla_W f(W_0, H_0)$

$$\rightarrow \quad \underline{\underline{\alpha_W = \frac{\| \nabla_W f(W_0, H_0) \|_F^2}{\| \nabla_W f(W_0, H_0)\, H_0 \|_F^2}}}$$

Similarly:

$$\underline{\underline{\alpha_H = \frac{\| \nabla_H f(W_0, H_0) \|_F^2}{\| W_0 \cdot \nabla_W f(W_0, H_0) \|_F^2}}}$$

(iii) Given arbitrary W and H, how do the orthogonal projections $P_1(W)$ and $P_2(H)$ onto the feasible set read?

a) iii) By formula (5.32) in the lecture notes,

$$P_1(W)\big|_{ij} = \max(w_{ij}, 0)$$

$$P_2(H)\big|_{ij} = \max(h_{ij}, 0)$$

In Matlab it is convenient to write

$$W = \max(W, 0)$$

$$H = \max(H, 0)$$

to replace W and H by their projections.

*(b) Prove that the NMF problem always has at least one global solution.

b) $f(W,H) = \frac{1}{2}\|A - WH\|_F^2 \to \min, \quad W \geq 0, \ H \geq 0.$

Let $(W_n), (H_n)$ be a minimizing sequence,

$$f(W_n, H_n) \longrightarrow \inf_{W\geq 0, H\geq 0} f(W,H) \geq 0.$$

Obviously, $(W_n H_n)$ has to be bounded in $\mathbb{R}^{m\times n}$.

By rescaling the columns of W_n, we can assume that the rows of H are either zero or have infinity norm 1.

Let $w_i^{(n)}, h_i^{(n)}$ denote the i-th rows of W_n and H_n^T, respectively. Then since the sum

$$W_n H_n = \sum_{i=1}^{k} w_i^{(n)} (h_i^{(n)})^T$$

is bounded and every summand is nonnegative, every term $w_i^{(n)} (h_i^{(n)})^T$ has to be bounded, say $\max_{k,\ell} (w_i^{(n)})_k (h_i^{(n)})_\ell \leq C$

Case 1: $h_i^{(n)} = 0$: Then we assume without loss of generality, that $w_i^{(n)} = 0$.

Case 2: $\|h_i^{(n)}\| = 1$: Then there exist an entry $(h_i^{(n)})_\ell = 1$.

Thus: $\|w_i^{(n)}\|_\infty = \max_k (w_i^{(n)})_k = \max_{k,\ell} (w_i^{(n)})_k (h_i^{(n)})_\ell \leq C.$

It follows that we can assume both sequences $(W_n), (H_n)$ to be bounded. Thus there exist common subsequences $(W_{n_k}), (H_{n_k})$ converging to $W_* \geq 0$, $H_* \geq 0$, respectively.

By continuity of f, $f(W_*, H_*) = \inf_{W\geq 0, H\geq 0} f(W,H)$. ∎

3 ▶ Solving NMF with alternating projected gradients

(a) Implement in MATLAB the following function for NMF:

Require: matrix $A \geq 0$, $k \in [1, \min(m,n)]$, starting guesses $W_0, H_0 \geq 0$, maxiter
$i \leftarrow 1$, $W \leftarrow W_0$, $H \leftarrow H_0$
while $\|P(\nabla f(W,H))\|_F > 10^{-5}\|\nabla f(W_0,H_0)\|_F$ and $i <$ maxiter **do**
$W \leftarrow P_1(W - \alpha_W \nabla_W f(W,H))$
$H \leftarrow P_2(H - \alpha_H \nabla_H f(W,H))$
$i \leftarrow i + 1$
end while

```
function [W,H] = nmf_projgrad(A,W0,H0,maxiter)

s = size(A);
f = @(B,C) norm(A - B*C)^2/(2*s(1)*s(2));

% Project W0 and H0 if necessary
W0 = max(W0,0); H0 = max(H0,0);

% Gradients of starting value
g_W0 = (W0*H0 - A)*H0';
g_H0 = W0'*(W0*H0 - A);
g0 = [g_W0; g_H0'];

W = inf; H = inf; g = g0;

c1 = 1e-2; beta = .1; tol = 1e-4; i=1;

while norm(max(g,0),'fro') > tol*norm(g0,'fro') && i < maxiter
    fprintf(1,[num2str(i) ' ␣'])
    if mod(i,10) == 0
        fprintf(1,'\n')
    end
    %% Projected gradient step for W
    alphaW = norm(g_W0,'fro')^2/norm(g_W0*H0,'fro')^2; %optimal for unconstrained
    W = max(W0 - alphaW*g_W0,0);

    W0 = W;
    g_H0 = W0'*(W0*H0 - A);

    %% Projected gradient step for H
    alphaH = norm(g_H0,'fro')^2/norm(W0*g_H0,'fro')^2; %optimal for unconstrained
    H = max(H0 - alphaH*g_H0,0);

    %% new gradients
    H0 = H;
    g_W0 = (W0*H0 - A)*H0';
    g_H0 = W0'*(W0*H0 - A);
    g = [g_W0; g_H0'];

    i = i+1;
end
end
```