Supplementary Material for Sparse Filtering

Jiquan Ngiam, Pang Wei Koh, Zhenghao Chen, Sonia Bhaskar, Andrew Y. Ng
Computer Science Department, Stanford University
{jngiam,pangwei,zhenghao,sbhaskar,ang}@cs.stanford.edu

1 Matlab Implementation of Sparse Filtering

Algorithm 1 Sparse Filtering Matlab Implementation

```
function [optW] = SparseFiltering(N, X);
   % N = # features to learn, X = input data (examples in column)
   % You should pre-process X by removing the DC component per example,
   % before calling this function.
   %   e.g., X = bsxfun(@minus, X, mean(X));
   addpath minFunc/ % Add path to minFunc optimization package
   optW = randn(N, size(X, 1));
   optW = minFunc(@SparseFilteringObj, optW(:), struct('MaxIter', 100), ...
                  X, N);
   optW = reshape(optW, [N, size(X, 1)]);
end

function [Obj, DeltaW] = SparseFilteringObj (W, X, N)
   % Reshape W into matrix form
   W = reshape(W, [N, size(X,1)]);

   % Feed Forward
   F  = W*X;                      % Linear Activation
   Fs = sqrt(F.^2 + 1e-8);        % Soft-Absolute Activation
   [NFs, L2Fs]  = l2row(Fs);      % Normalize by Rows
   [Fhat, L2Fn] = l2row(NFs');    % Normalize by Columns

   % Compute Objective Function
   Obj = sum(sum(Fhat, 2), 1);

   % Backprop through each feedforward step
   DeltaW = l2grad(NFs', Fhat, L2Fn, ones(size(Fhat)));
   DeltaW = l2grad(Fs,   NFs,  L2Fs, DeltaW');
   DeltaW = (DeltaW .* (F ./ Fs)) * X';

   DeltaW = DeltaW(:);
end

function [Y,N] = l2row(X) % L2 Normalize X by rows
   % We also use this to normalize by column with l2row(X')
   N = sqrt(sum(X.^2,2) + 1e-8);
   Y = bsxfun(@rdivide,X,N);
end

function [G] = l2grad(X,Y,N,D) % Backpropagate through Normalization
   G = bsxfun(@rdivide, D, N) - bsxfun(@times, Y, sum(D.*X, 2) ./ (N.^2));
end
```