```
   end
end
end
```

The function `buildhardSVM2` builds the constraint matrix and the matrices defining the quadratic functional.

```
function [A,c,X,Xa,q] = buildhardSVM2(u,v)
%   builds the matrix of constraints A for
%   hard SVM h2, and the right hand side c
%   Aso builds X and Xa = X'*X, and the vector q = -1_{p+q}
%   for the linear part of the quadratic function
%   The right-hand side is c = 0 (Ax = 0).
p = size(u,2); q = size(v,2);
A = [ones(1,p) -ones(1,q)];
c = 0;
X = [-u v];
Xa = X'*X;
q = -ones(p+q,1);
end
```

The function `countmlu2` returns a vector consisting of those λ_i such that $\lambda_i > 0$, and the number of such λ_i.

```
function [lambnz, mlu] = countmlu2(lambda,tols)
%  Counts the number of points u_i (in u)
%  such that lambda_i > 0 and returns a vector
%  of these lambda_i

% tols = 10^(-11);
p = size(lambda,1); lambnz = zeros(p,1);
mlu = 0;
for i = 1:p
    if lambda(i) > tols
        mlu = mlu + 1;
        lambnz(i) = lambda(i);
    end
end
end
```

The function `countmlv2` returns a vector consisting of those μ_j such that $\mu_j > 0$, and the number of such μ_j. It is similar to `countmlu2`. Here a judicious choice of `tols` is crucial and one has to experiment with various values.