```
end
end
```

The function `countumf2` finds those λ_i such that $\lambda_i = K$.

```
function [lamK,mf] = countumf2(lambda,K,tolh)
%  Counts the number of margin failures, that is,
%  points u_i (in u) such that lambda_i = K

p = size(lambda,1);
mf = 0; lamK = zeros(p,1);
for i = 1:p
    if lambda(i) >= K - tolh
       mf = mf + 1;
       lamK(i) = lambda(i);
    end
end
end
```

Similarly, the function `countvmf2` finds those μ_j such that $\mu_j = K$.

The function `countmlu2` finds those λ_i such that $\lambda_i > 0$.

```
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
```

Similarly, the function `countmlv2` finds those μ_j such that $\mu_j > 0$. The function `findsvl2` finds the λ_i corresponding to blue support vectors of type 1 and 2 and the error vector ϵ. The number of blue errors is *psf* (the u_i for which $\epsilon_i > 0$). Similarly the function `findsvm2` finds the μ_j corresponding to red support vectors of type 1 and 2 and the error vector ξ. The number of red errors is *qsf* (the v_j for which $\xi_j > 0$).