

## Routines for empirical Bayes mixture estimation

Some routines for the S-PLUS language are provided; these are intended as templates rather than as a package to be used blindly. The routines for single sequences allow the methods of the paper to be applied to a single sequence using the standard parts of the S-PLUS language.

Routines, using the S+Wavelets module, are provided for carrying out the smoothing procedure on the wavelet transform of data. If the user does not have this module then the single sequence operations can easily be used to construct routines for smoothing wavelet transforms found using other software.

Finally, the programs that produced the simulations in the paper are provided; since these use repeatable random number generators, the user may generate the same data to compare other methods.

### *Single sequence operations*

#### Marginal maximum likelihood estimation of parameters

Routines are provided for both the double exponential and the Cauchy-tail prior. In the double exponential case, both the scale parameter and the mixing probability can be estimated. In the Cauchy-tail case, there is only a mixing parameter to estimate. The function `ebaysmooth` carries out the calculations for the double exponential case.

The syntax is:

```
ebaysmooth(x, a = 0, sig = 1, returnthresh = F,  
           posteriormean=F, hardthresh=F)
```

On entry

`x` the given data sequence

`a` the scale parameter; if on entry `a=0` then the routine uses marginal maximum likelihood to estimate both `a` and `w`; otherwise only `w` is estimated and `a` is assumed to have the given value

`sig` the noise standard deviation

If `returnthresh=T` then the routine returns a list with four elements:

`a` the scale parameter estimated by the procedure or supplied by the user

`w` the mixing weight estimated by the procedure

`thresh` the threshold value if the posterior median is used

`xsm` the result of applying the posterior median smoothing to the given data

If `returnthresh=F` then just the vector `xsm` is returned

If `posteriormean=T` then the posterior mean is used

If `hardthresh=T` then hard thresholding is applied; the threshold is that yielded by the posterior median function with parameters chosen by marginal maximum likelihood.

The routine `cauchysmooth` carries out the same calculations for the prior with Cauchy tails: in this case the syntax is

```
cauchysmooth(x, sig = 1, approx = T, returnthresh = F)
```

The parameters `x` and `sig` are as above. If `approx=T` then an approximation to the thresholding function is used. If `returnthresh=T` then the routine returns the estimated weight `w`; the threshold `thresh`; and the result `xsm` of applying posterior median smoothing. If `returnthresh=F` then just the vector `xsm` is returned.

## Finding the posterior median and mean with given parameters

To find the posterior median without estimating the parameters, the functions `postmed`, `postmedc` and `postmedcapprox` can be used. In each case it is assumed that the error variance is 1. For general data, divide by the standard deviation, apply the smoothing method, and multiply the result by the standard deviation. The syntax of the functions is

```
postmed( x, w, a=1)
```

```
postmean(x, w, a=1)
```

which finds the posterior median and mean, respectively, for the double exponential prior, and

```
postmedc(x, w)
```

```
postmedcapprox(x, w)
```

which both give the posterior median for the prior with Cauchy tails.

In each case `x` is the data vector, and `w`, and if appropriate `a`, are the parameters of the prior.

The routine `postmedcapprox` is often faster; it uses the asymptotic shrinkage function for large `x`, and a piecewise linear approximation to the thresholding function for `x` nearer to the threshold.

### **Wavelet smoothing routines**

These make use of the module `S+Wavelets`. The routine `ebaywaveshrink` is appropriate for data with independent errors:

```
ebaywaveshrink(x.dwt, vscale = NULL, a = 0, smooth.levels =  
NULL, cauchy = F, ...)
```

On entry:

`x.dwt` Wavelet transform of a sequence `x`, constructed using one of the wavelet transform routines, such as `dwt` for the classical discrete wavelet transform, or `nd.dwt` for the nondecimated transform.

`a` The scale parameter if the double exponential prior is used; if `a=0` then it will be estimated by marginal maximum likelihood together with the weight parameter.

`vscale` Noise standard deviation; if not specified then it will be estimated using the median absolute deviation of the coefficients at the finest scale.

`smooth.levels` If set to a smaller number than the value of `n.levels` in the transform, then only `smooth.levels` levels are processed.

`cauchy` If `T` then Cauchy prior is used, otherwise the double exponential prior is used.

After estimating the overall vertical scale if necessary, the routine applies `ebaysmooth` or `cauchysmooth` to each level of the transform, and then applies the appropriate inverse wavelet transform or reconstruction method to find the function estimate. For level dependent variability, use the routine `ebaywaveshrinklevdep` as a template.

### **Simulation studies**

The function `simulation1()` produces the results of Table 1 for single sequences. The function `simulation2()` provides a four-way array giving the performance on each replication for every combination of method, noise level and model. Either the standard or the nondecimated transform can be used. The function `simulation3()` gives the results of the block thresholding methods. To provide the summaries given in Tables 2 and 4, use the function `table2()`; the function `table3()` gives the comparison presented in Table 3.