De-noising Caruso’s recording

Shota Gugushvili

June 8, 2019

Caruso’s recording

Enrico Caruso (1873–1921) was a famous Italian tenor, who sang to great acclaim in major opera houses throughout the world. A good deal of his popularity was due to numerous commercial audio recordings he made. Despite primitive equipment of the time, Caruso’s voice recorded extremely well, and therein lay his advantage over many of his competitors. Plus, of course, he was a consummate artist, and his was a truly exceptional voice.

In this demonstration, I’ll use an excerpt from Caruso’s recording of “E lucean le stelle” (a tenor aria from Giacomo Puccini’s opera “Tosca”) to illustrate an amusing fact that residuals in the (non-parametric) regression analysis can not only be *displayed*, but also *heard*.

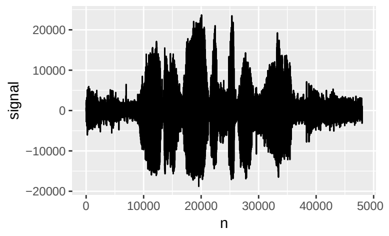
Data

First some preliminaries: I’ll use the following **R** packages in this document.

```r
library(ggplot2)
library(tidyverse)
library(waveslim)
library(tuneR)
library(EbayesThresh)
```

A digital signal of the aria (at the sampling rate 8192 Hz) is included in WaveLab, a free library of **Matlab** routines for wavelet analysis. No further information on the recording is provided, but it can be surmised that it was made sometime during the first decade of the 20th century. Throughout his career, Caruso recorded “E lucean le stelle” several times. The piano accompaniment that we hear would fit the following discs: Pathé from 1901 (matrix number 84004), Zonophone from 1902 (matrix number X 1553), or Gramophone Typewriter from 1903 (matrix number 52349). You can download the dataset here. After loading the dataset, it can be visualised as in the plot below.

```r
n <- 48000
caruso <- read.table("caruso.txt")[1:n,1]
ggplot()+
  geom_line(mapping = aes(x = 1:n, y = caruso)) +
  xlab("n") +
  ylab("signal")
```



It's more enlightening to listen to the recording. In **R**, one can either export the signal as a wave file (using the **tuneR** package) and play it externally, or call a command line audio player from within **R**. The code for the latter option is platform-dependent, so I'll use the former, and let you do the manual work.

```
caruso_wave <- Wave(caruso, samp.rate = 8192, bit = 16)
writeWave(caruso_wave, filename = "caruso.wav")
```

You can now play the caruso.wav file on your computer. I've also linked it here.

De-noising

Upon hearing the recording, presence of a substantial noise on top of the signal becomes apparent. There're many cutting-edge audio de-noising techniques, but since my primary goals are illustrative, I'll stick to using the Discrete Wavelet Transform (DWT), see (Percival and Walden 2000). DWT is a special kind of a linear (orthogonal) transformation. Loosely speaking, it has a property of concentrating most of the signal in a few large components (coefficients), while spreading the original noise "uniformly" in the transformed version. Wavelet approach to de-noising consists in squashing "small" wavelet coefficients to zero, since they hardly contain any signal whatsoever. I'll use the empirical Bayes approach to wavelet de-noising, see (Johnstone and Silverman 2005b) and (Johnstone and Silverman 2005a). The latter method is implemented in the **ebayesthresh** package.

The first step is to evaluate DWT; there are several options available, but I'll use the **waveslim** package. I'll compute a (partial) DWT with 2 levels of the transform. De-noising more than 2 levels appears to be too invasive in the present setting.

```
caruso_dwt <- dwt(caruso, wf = "la8", n.levels = 2, boundary = "reflection")
```

The next step is to de-noise the DWT.

```
EBayes <- ebayesthresh.wavelet(caruso_dwt, threshrule = "mean", prior = "laplace", a = NA, vscale = "indep
```

Now I'll invert the transform, and next round off the values to be able to play the audio file.

```
caruso_ebayes <- waveslim::idwt(EBayes)[1:n] %>%
  round()
```

Finally, I'll create the wave file of the de-noised recording. It's also available for download here.

```r
caruso_ebayes %>%
  Wave(samp.rate = 8192, bit = 16) %>%
  writeWave(filename = "caruso_ebayes.wav")
```

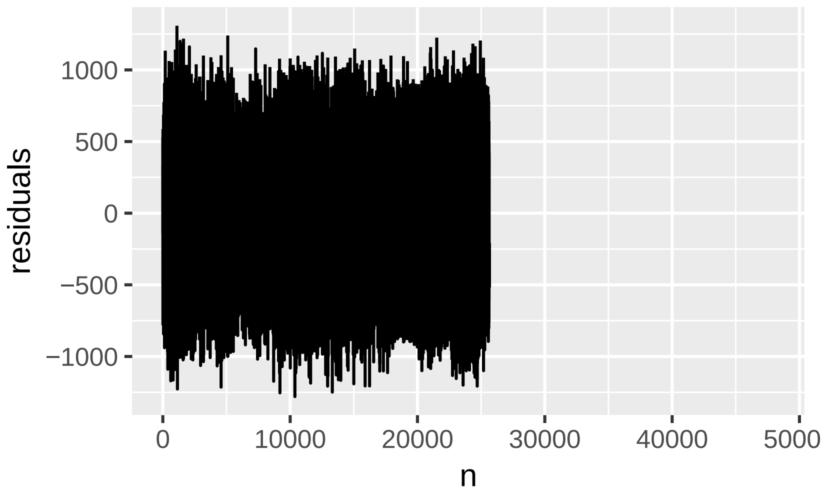
Listen to the file and judge it for yourself. To my ear, the results aren't that brilliant: the original "noisy" version sounds more natural! (play it again)

The truth is that DWT is not the most appropriate transform for audio de-noising purposes. Perhaps the same should be said of the empirical Bayes de-noising in (Johnstone and Silverman 2005b), as it wasn't designed in view of audio de-noising applications, and hence doesn't account for features important in that field; cf. (Wolfe, Godsill, and Ng 2004) and (Godsill et al. 2007). This despite all kinds of nice mathematical theorems established in (Johnstone and Silverman 2005b), that deal with optimality properties of the empirical Bayes de-noising.

Residuals

One can also judge the de-noising quality by examining the residuals that resulted from the empirical Bayes method. These residuals can be plotted, as in the figure below.

```r
caruso_residuals <- caruso - caruso_ebayes
ggplot()+
  geom_line(mapping = aes(x = 1:n, y = caruso_residuals)) +
  xlab("n") +
  ylab("residuals")
```



It looks like the method has removed quite some noise there, but let's *hear* the residuals. I have also placed the wave file here.

```
caruso_residuals %>%
  Wave(samp.rate = 8192, bit=16) %>%
  writeWave(filename = "caruso_residuals.wav")
```

It turns out that when performing de-noising, the empirical Bayes has chopped off a decent amount of the signal too: amidts the noise, one can clearly distinguish Caruso's voice, though at a reduced volume! Now subconsciously, a human brain is unlikely to accept such an outcome. One possibility here is to use a multi-step method. That is, one can repeat the whole procedure, but now applied on the residuals from the first step. One may hope that thereby one will pile off another layer of the signal and add it to the "de-noised" reconstruction from the first step. This doesn't sound like a technique commonly considered in the statistics literature when dealing with de-noising applications, though it's been suggested elsewhere; see (Berger, Coifman, and Goldberg 1994).

And this brings the present illustrative example to completion. I hope you enjoyed reading it!

References

Berger, Jonathan, Ronald R. Coifman, and Maxim J. Goldberg. 1994. "Removing Noise from Music Using Local Trigonometric Bases and Wavelet Packets." *J. Audio Eng. Soc* 42 (10): 808–18.

Godsill, S. J., A. T. Cemgil, C. Févotte, and P. J. Wolfe. 2007. "Bayesian Computational Methods for Sparse Audio and Music Processing." In *15th European Signal Processing Conference (EURASIP)*, 345–49.

Johnstone, Iain M., and Bernard W. Silverman. 2005a. "EbayesThresh: R Programs for Empirical Bayes Thresholding." *J. Stat. Softw.* 12 (8): 1–38. doi:10.18637/jss.v012.i08.

———. 2005b. "Empirical Bayes Selection of Wavelet Thresholds." *Ann. Statist.* 33 (4): 1700–1752. doi:10.1214/009053605000000345.

Percival, Donald B., and Andrew T. Walden. 2000. *Wavelet Methods for Time Series Analysis*. Vol. 4. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511841040.

Wolfe, Patrick J., Simon J. Godsill, and Wee-Jing Ng. 2004. "Bayesian Variable Selection and Regularization for Time-Frequency Surface Estimation." *J. R. Stat. Soc. Ser. B Stat. Methodol.* 66 (3): 575–89. doi:10.1111/j.1467-9868.2004.02052.x.