Non-Local Means for Audio Denoising

Arthur Szlam

***Abstract*—The application of the NL-Means algorithm (and some cousins) to audio denoising is discussed.**

***Index Terms*—NL-means, PDE on graphs, audio denoising**

The NL-means algorithm of [BCM05] regularizes images by running the heat equation on the space of patches of the image. When well engineered, it is one of the most effective methods currently available for denoising natural images contaminated with white Gaussian noise.

In this note we describe how and why the NL-means algorithm (and some other related algorithms) can be applied to audio denoising. We start by describing the algorithm, and giving heuristics for its successful deployment. We then remark that the locally stationary model of audio signals is a good match for the heuristics we have given. Finally, we conclude with a numerical experiment to demonstrate the above ideas.

I. NL means and PDE on graphs

In its simplest form, NL means goes as follows: given a noisy $m \times n$ image f_0, fix a patch size k, and consider the set of all $k \times k$ patches of f_0. Denote the patch at pixel j by P_j, where $j \in \{1, ..., mn\}$. Fix a variance parameter σ, and form the $mn \times mn$ matrix

$$W(i,j) = e^{-||P_i - P_j||_{\mathrm{F}}^2/\sigma}, \tag{I.1}$$

where 'F' is for Frobenious and

$$||P_i - P_j||_{\mathrm{F}}^2 = \sum_{a=1}^{k}\sum_{b=1}^{k} |P_i(a,b) - P_j(a,b)|^2; \tag{I.2}$$

thus $||P_i - P_j||_{\mathrm{F}}$ is just the norm of the difference of the two patches considered as a k^2 dimensional vectors. Set

$$d(i) = \sum_j W(i,j)\,, \tag{I.3}$$

and let the filter

$$K(i,j) = d^{-1}(i) W(i,j)\,, \tag{I.4}$$

so that $\sum_{y \in V} K(x,y) = 1$, and so multiplication of a vector from the left by K is an averaging operation. To obtain the denoised image f_1 from f_0, write f_0 as a column vector, and set

$$f_1 = K f_0. \tag{I.5}$$

In this way, to find the correct intensity at a given pixel j, we average the intensities of all the pixels i such that i and j have similar neighborhoods, with a Gaussian weight.

A. The heat equation on weighted graphs

We now follows [ZS05]; see also [GO07]. Given a weighted graph with n vertices V and weights W, set the density

$$d(i) = \sum_j W(i,j)\,, \tag{I.6}$$

and let the matrix D be the diagonal matrix with diagonal d. Let g be a function on V. The normalized gradient at a vertex i is defined to be the vector

$$\nabla_w(g)\Big|_i = \begin{pmatrix} \sqrt{\frac{W(i,1)}{d(i)}}g(i) - \sqrt{\frac{W(i,1)}{d(j)}}g(1) \\ \sqrt{\frac{W(i,2)}{d(i)}}g(i) - \sqrt{\frac{W(i,2)}{d(2)}}g(2) \\ \vdots \\ \sqrt{\frac{W(i,n)}{d(i)}}g(i) - \sqrt{\frac{W(i,n)}{d(n)}}g(n) \end{pmatrix}. \tag{I.7}$$

The smoothness functional $\sum_i ||\nabla_w(g)\Big|_i||^2$ is the discrete analog of $\int |\nabla g|^2$ for a smooth image. Gradient descent on this functional starting from g leads to the equations

$$g_0 = g,$$

$$g_{t+1} - g_t = (D^{-\frac{1}{2}} W D^{-\frac{1}{2}} - I) g_t. \tag{I.8}$$

Since $D^{-\frac{1}{2}} W D^{-\frac{1}{2}} - I$ is the normalized graph Laplacian [Chu97], equation I.8 is simply the discrete time (density normalized) heat equation on the graph W with initial condition given by g_0.

If we iterate the multiplication in equation (I.5), we get

$$f_{t+1} = K f_t, \tag{I.9}$$

which with a little manipulation becomes the equation

$$D^{\frac{1}{2}}(f_{t+1} - f_t) = (D^{-\frac{1}{2}} W D^{-\frac{1}{2}} D^{\frac{1}{2}} - D^{\frac{1}{2}}) f_t. \tag{I.10}$$

Setting $g_t = D^{\frac{1}{2}} f_t$, we get the equation

$$g_{t+1} - g_t = (D^{-\frac{1}{2}} W D^{-\frac{1}{2}} - I) g_t. \tag{I.11}$$

and so we are actually running the (density normalized) heat equation on the set of patches of f_0 with weights given by W [Szl06], [GO07], and moreover, as above, this is exactly the gradient descent for the energy

$$\sum |\nabla_w g|^2. \tag{I.12}$$

Note that most of the familiar ideas from the Euclidean heat equation carry over to this setting. In particular, the normal relationships between scale, smoothness, and frequency persist in this setting; remembering that frequency now refers to the eigenfunctions and eigenvalues of the graph Laplacian defined above. Also note that if we would like to balance smoothing by K with fidelity to the noisy f, we can choose $\beta > 0$ and, as before, set

$$f_0 = f, \tag{I.13}$$

but now set

$$f_{t+1} = (Kf_t + \beta f)/(1+\beta); \tag{I.14}$$

the noisy function is treated as a heat source. In contrast to the equation evolving without a heat source, which tends to a constant steady state, this version of the equation tends to a non-constant steady state.

One final note: patches are not the only useful features for image denoising. For example, good results can be obtained using Gabor filter responses as features [BSS07], or using grid shifted wavelet or curvelet denoisings as features[Szl06]. With some small amount of engineering, the methods described in this section are among the best known denoising algorithms for natural images.

B. computational costs

In practice the algorithm as described is not feasible, because to even construct the matrices W and D requires time at least $(mn)^2$. One possible solution is to use some sort of fast nearest neighbor searcher in the space of patches, and fixing a number r, for each pixel i set $W(i,j) = 0$ if j is not one of the r nearest neighbors to i, and as before set $W(i,j) = e^{-||P_i - P_j||_F^2/\sigma}$ if j is one of the r nearest neighbors of i. A simpler approach which works even better is to only search for neighbors in an $l \times l$ box around a given pixel. Searching in the $l \times l$ box limits the construction time of W to mnk^2l^2 operations; and D, which is built by summing the rows of W, costs mnk^2 operations. To then run the equation costs $Mmnk^2$, where M is the number of times the matrix K is iterated. The restricted search does not degrade the denoising; in fact, it actually improves the results. The reasons for this are not completely understood, but probably have to do with the importance of certain kinds of image features, especially edges, whose likely patch-space neighbors lie nearby in pixel space.

C. When does NL-means work?

The success of the algorithm rests on three assumptions:

- If two patches are close in Frobenious norm, then the center pixel of the two patches should have nearly the same intensity.
- pixels sit in patches which have many instances in the image; and
- The noise does not grossly change the distance between patches, or at least does not change the nearest neighbors of a patch.

Note if features other than patches are used, the first assumption can be changed to "if the pixels are close in features space, then the pixels should have the same intensity". This assumption is simply that the intensity function is smooth as a function on the space of patches. The second assumption says that the patches (features) of pixels are not isolated; of course any function at an isolated point is smooth. In order for smoothness to be useful, each pixel should lie in a (feature) neighborhood of similar patches. The third assumption says that the geometry of the patches (features) is not badly distorted by noise. In particular, for white Gaussian noise, if $P_j^n = P_j + \eta_1$ and $P_i^n = P_i + \eta_2$ are noisy patches with two different realizations of noise η_1 and η_2,

$$||P_i^n - P_j^n||^2 = ||P_i - P_j||^2 + \langle P_i - P_j, \eta_2 - \eta_1 \rangle + ||\eta_1 - \eta_2||^2.$$

As the patches become large, we expect the middle term to be small, and the last term to be near twice the variance of the noise. Under these circumstances, although all the distances have been changed, the nearest neighbors to any given patch remain roughly the same. Note that if the noise has spatial correlations or is expected to be correlated with the image patches, the distances could be distorted in a much more serious way.

If all three conditions are satisfied, then the intensity function should be smooth as a function of the noisy patches (features), and thus can be denoised by running the heat equation. For a more rigorous discussion of related ideas see [SSN07]; note also that we do not need any explicit knowledge of the geometry of the patches (i.e. parameterization, etc...), although sometimes these are available (see [Pey07]).

II. Application to audio denoising

It is well known that many common audio signals are locally stationary; in other words, if one takes a windowed Fourier transform with a small sliding window, the absolute value of the coefficients will be roughly constant for short shifts of the window. For audio sampled at the usual rates, with windows of size a few hundred samples, "short shifts" includes shifts several times the size of the window. The upshot is that near any sample we expect to find many (almost exact) copies of the patch centered at that sample; and moreover, the value at a sample should be determined by the phase of that patch. We thus get the first two conditions necessary for an NL means type method to work for audio denoising. If the noise is white Gaussian, and our patch size is large enough, as before, the third condition is satisfied.

We then can run the NL-means algorithm in essentially the same way for the audio signals as for the images (if anything, the algorithm is simplified since the signal and patches are 1-d arrays). Below we will illustrate the ideas in section C in the context of audio signals and then give a denoising example.

A. Experiments

We use an audio sample f (at 11025 Hz) of a recording of Darwin's "On the Origin of Species by Means of Natural Selection", obtained from `http : / / librivox . org / the-origin-of-species-by-charles-darwin/`, in mp3 format. We use the Matlab package mp3read available at `http://labrosa.ee.columbia.edu/matlab/mp3read.html` to read the a 50000 sample from the recording into Matlab. We add .1 times Gaussian white noise to get a degraded signal f^n with SNR $= -7.94$ (correlation $= .373$).

We would first like to illustrate the ideas in section C in figures 4-7. First, in figures 4 and 6, we have 1000 samples from the clean and noisy signals. In figure 7, the set of all patches of length 257 from the 1000 noisy samples (i.e. 1000 points in $\mathbb{R}^{257}$) is shown projected onto its first three principal components. The color signifies the (clean) function value (speaker position) at the sample in the center of the patch; that is, the color is the y axis in figure 4. In figure 5, the set of clean patches is shown, projected onto the same subspace as the patches in figure 7, for ease of comparison. The important thing to notice is that the function value is smooth in the noisy patch space, even though the original audio signal was not smooth in the time domain.

We now build the weights matrix and denoise the signal. For each sample j, set S_j to be the search window consisting of the patches of f^n centered at the 256 samples preceding j, and the 256 samples following j; and let $U_j \subset S_j$ be the 21 nearest patches of f^n to P_j. Then we then build a weights matrix W by setting

$$W(i,j) = \begin{cases} 1 & \text{if } P_i \in U_j \\ 0 & \text{otherwise} \end{cases} ;$$

that is, we set $\sigma = \infty$ in equation (I.1). We average W with its transpose to symmetrize, and then normalize as above to get a matrix K with row sums equal to one. We then set $f_0 = f^n$, and $f_{t+1} = (Kf_t + .01f^n)/1.01$, i.e., set $\beta = .01$ as in (I.14). After three iterations we get a denoised signal with SNR $= 3.29$ (correlation $= .739$).

B. Some notes on the various parameters

The reader will have noticed the large number of seemingly arbitrarily set parameters introduced in the above example. Some descriptions of the tradeoffs entailed in the various choices:

- patch size: the larger the patch size, the more resistant patch similarity is to the influence of noise, as for large patches, the distance between realizations of white noise is roughly constant. However, the larger the patch size, the harder it is to find similar patches, even in clean signals. Our experience suggests that window sizes on the order of .01-.05 seconds seem to work well for speech signals.
- search window size: the larger the search window, the easier to find a similar patch. However, large windows also increase the chance of spurious matches, and the run time of the algorithm is determined by the size of the search window. Search windows one to two times the patch size seem to work well for speech.
- σ: larger σ means more smoothing per iteration. We use $\sigma = \infty$ for simplicity. In image denoising, this parameter is quite important; but here, because we expect almost exact patch matches, setting $\sigma = \infty$ and controlling the smoothing with the number of neighbors seems ok.
- number of neighbors: larger number of neighbors means more smoothing per iteration. Too many and the chosen patches are no longer very similar; too few means bad choices for nearest neighbors are more costly. This parameter seems to be quite important, especially considering in our experiment $\sigma = \infty$, but for many audio signals the smoothing kernel constructed at coherent regions is banded with bands at the fundamental period of the local oscillation. It seems good to choose just enough neighbors so one can see this banded structure.
- number of iterations of the weight matrix: larger number of iterations means less noise, but also less signal. With no forcing weight, iterations of the weight matrix converge to a constant function.
- forcing weight: the larger the forcing weight β, the larger the residual noise, but the smaller the distortion after multiple iterations of K.

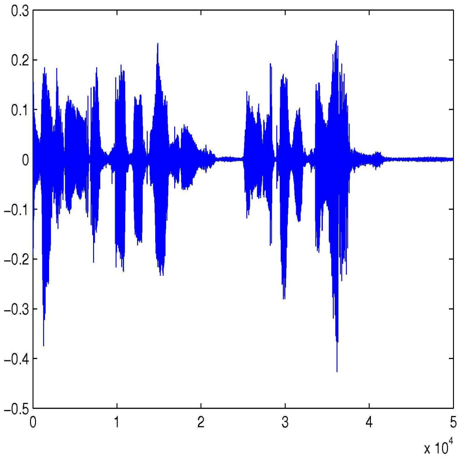

Fig. 1. 50000 samples from a recording of a reading of "On the Origin of Species", sampled at 11025 Hz

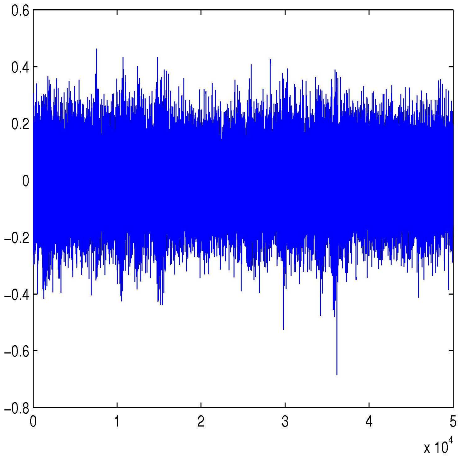

Fig. 2. The audio sample from 1 with Gaussian white noise added, SNR $= -7.94$ (correlation $= .373$)

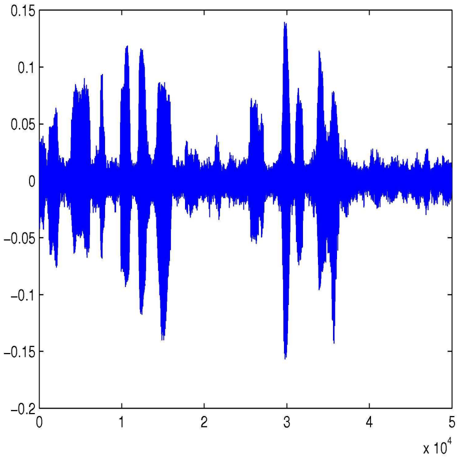

Fig. 3. The noisy audio sample in figure 2 denoised using the heat equation on the weighted graph of patches. Recovered SNR = 3.29 (recovered correlation = .739)

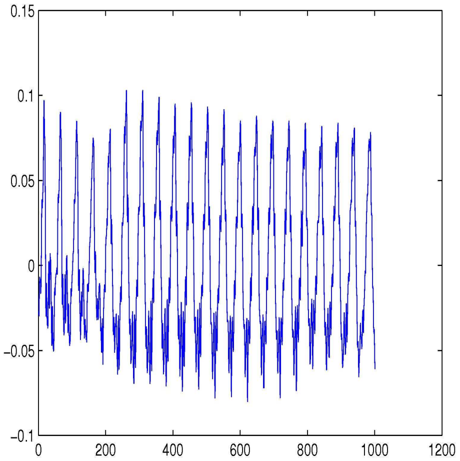

Fig. 4. 1000 samples from an audio recording of "The Origin of Species"

III. Conclusions and future work

The NL-means algorithm is useful for audio denoising because many audio signals are locally stationary. It is possible that other features are also useful, e.g. Gabor filter responses, or perhaps local (in time) PCA vectors.

IV. Acknowledgements

Thanks to the NSF for generous support by DMS-0811203.

References

[BCM05] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4(2):490–530 (electronic), 2005.

[BSS07] J. Bremer, Y. Shkolnisky, and A. Szlam. Image denoising using diffusion on curvelet-scaled gabor filter responses. Technical Report CAM Rep. 07-46, Computational and Applied mathematics, UCLA, August 2007.

[Chu97] F. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

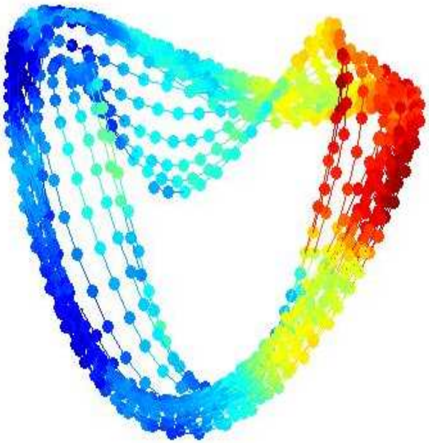

Fig. 5. 1000 patches (the same ones as from figure 4) from an audio recording of "The Origin of Species", as points in patch space; each point is a patch projected onto the principal components of the 1000 noisy patches in figures 6 and 7. The color is the clean function value; and the segments indicate adjacency in time. Note that the function value is smooth in patch space.

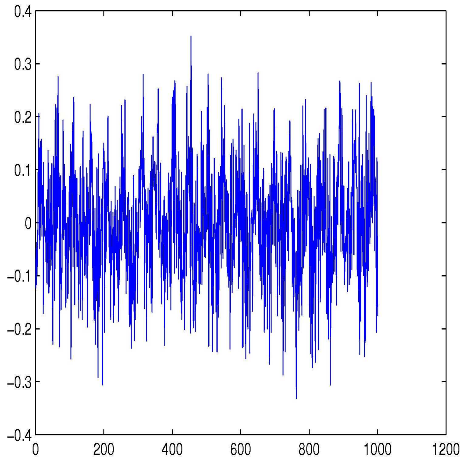

Fig. 6. 1000 samples from an audio recording of "The Origin of Species" after the addition of noise.

[GO07] Guy Gilboa and Stanley Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[Pey07] G. Peyr. Manifold models for signals and images. *Preprint Ceremade*, 2007.

[SSN07] A. Singer, Y. Shkolnisky, and B. Nadler. Diffusion interpretation of non-local neighborhood filters for signal denoising. In *preprint*, 2007.

[Szl06] A.D. Szlam. *Non stationary analysis on data sets and applications.* PhD thesis, Department of Mathematics,Yale University, June 2006.

[ZS05] D. Zhou and B. Schlkopf. Regularization on discrete spaces. pages 361–368, Berlin, Germany, 08 2005. Springer.

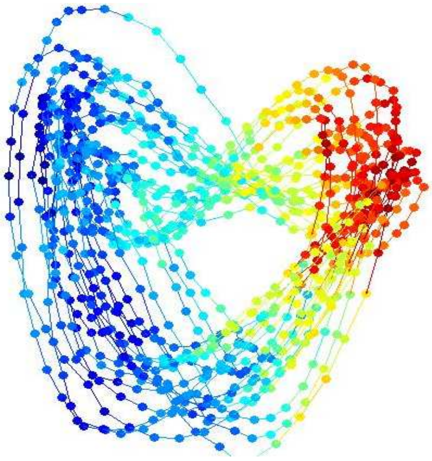

Fig. 7. 1000 patches (the same ones as from figure 6) from an audio recording of "The Origin of Species" after the addition of noise, as points in patch space; each point is a patch projected onto the first three principal components of this set of noisy patches. The color is the clean function value; and the segments indicate adjacency in time. Note that the much of the geometry has been preserved; in particular, the function value is still smooth even on the noisy patches.