ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Auto-adaptive Laplacian Pyramids

Ángela Fernández[1], Neta Rabin[2], Dalia Fishelov[2] and José R. Dorronsoro[1,3] *

[1] Instituto de Ingeniería del Conocimiento
[2] Department Exact Sciences, Afeka
[3] Universidad Autónoma de Madrid

angela.fernandez@iic.uam.es,netar@afeka.ac.il,daliaf@post.tau.ac.il,jose.dorronsoro@uam.es

Abstract. An important challenge in Data Mining and Machine Learning is the proper analysis of a given dataset, especially for understanding and working with functions defined over it. In this paper we propose Auto-adaptive Laplacian Pyramids (ALP) for target function smoothing when the target function is defined on a high-dimensional dataset. The proposed algorithm automatically selects the optimal function resolution (stopping time) adapted to the data defined and its noise. We illustrate its application on a radiation forecasting example.

1 Introduction

Manifold learning algorithms have become a common way for processing and analyzing high-dimensional data. The so called "diffusion analysis" allows us to find the most appropriate geometry to study such data and the functions defined on them [1]. These methods are based on the construction of a diffusion operator that depends on the local geometry of the data, which is then used to embed the high-dimensional points into a lower-dimensional space while maintaining their geometric properties.

However, these low dimensional embeddings are not given by an input-output function and extending them or, more generally, interpolating a function defined on such an embedding to new points may be challenging. Traditionally, Nyström based methods such as Geometric Harmonics (GH) [2] have been used for this purpose. Nonetheless these approaches also require a careful setting of parameters and, in addition, there does not exist a robust method for selecting the correct neighborhood. A first attempt to simplify these approaches was via Laplacian Pyramids (LP) [3], a multi-scale model that generates a smoothed version of a function in an iterative manner by using Gaussian kernels of decreasing widths [4]. These kind of models lose the GH property of being maximally concentrated on the dataset, but they have the gain of its simplicity.

Nevertheless, when we apply an LP model, there is a risk of overfitting if we try to refine too much the prediction during the training phase, as was shown in [5]. A usual approach to avoid overfitting is to apply Cross Validation (CV) to measure a validation error during training and then stop it when this error starts

*With partial support from Spain's grants TIN2013-42351-P (MINECO) and S2013/ICE-2845 CASI-CAM-CM (Comunidad de Madrid), and the UAM–ADIC Chair for Data Science and Machine Learning. The authors gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at UAM. They wish to thank Prof. Ronald R. Coifman for helpful remarks.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

to increase. The extreme form of CV is Leave One Out CV (LOOCV), which is theoretically supported and often yields good results, but with the drawback of a heavy computational cost.

In this paper we propose Auto-adaptive LP (ALP), which is a modification in the LP process. It merges model training and an approximate LOOCV estimation in one single phase. To do so we will simply modify the kernel matrix to have zeros on its diagonal. This change, first, reduces significantly training complexity but, also, provides an automatic estimate of the CV error that allows to stop training when overfitting risk may appear. Moreover, even if approximate LOOCV is achieved, it adds no extra cost compared to other classical neighbor-based interpolation methods.

This paper is organized as follows. In Section 2 we review the LP model and in Section 3 we describe the new ALP approach. Then, we will illustrate it in a radiation forecasting problem in Section 4. The paper ends with conclusions in Section 5.

2 Laplacian Pyramids

Laplacian Pyramids (LP) is an iterative model that was introduced by Burt and Adelson [4] for its application to image processing. In [3] a multi-scale algorithm was introduced in the spirit of LP to be applied in the setting of high-dimensional data analysis. We will follow this reference to review the LP procedure.

Let $S = \{x_i\}_{i=1}^N \in \mathbb{R}^M$ be the sample dataset; the algorithm approximates a function f defined over S by constructing a series of functions $\{\tilde{f}_i\}$ obtained by several refinements d_i over the approximation errors. In a slight abuse of language we will use the same notation f for both the general function $f(x)$ and also for the vector of its sample values $f = (f_1 = f(x_1), \dots, f_N = f(x_N))$. This process gives as result a function approximation.

In particular, we define a kernel $K^0(x, x') = \kappa \exp(-\|x - x'\|^2/\sigma^2)$, where κ is the Gaussian kernel normalizing constant. We will use the K^0 notation for both the general continuous kernel $K^0(x, x')$ and for its discrete matrix counterpart $K^0_{jk} = K^0(x_j, x_k)$ over the sample points. The smoothing operator P^0 is constructed as the normalized row-stochastic kernel matrix $P^0_{ij} = K^0_{ij}/\sum_k K^0_{ik}$.

A first coarse representation of f is then generated by the convolution $\tilde{f}_0 = f * P^0$ that captures the low-frequencies of the function. For the next steps we fix a parameter $\mu > 1$, and we construct at each level i a sharper Gaussian kernel P^i with scale σ/μ^i. Then, the residual $d_i = f - \tilde{f}_{i-1}$, which captures the error of the approximation to f at the previous $i-1$ step, is used to generate a more detailed representation of f given by $\tilde{f}_i = \tilde{f}_{i-1} + d_i * P^i = \tilde{f}_{i-1} + g_{i-1}$, with $g_\ell = d_\ell * P^\ell$. The iterative algorithm stops once the norm of the residual vector d_i is smaller than a predefined error. Stopping at iteration L, the final LP model has thus the form

$$\tilde{f}_L = \tilde{f}_0 + \sum_{\ell=1}^{L} g_\ell = f * P^0 + \sum_{\ell=1}^{L} d_\ell * P^\ell.$$

Extending this multi-scale representation to a new data point $x \in \mathbb{R}^M$ is now straightforward because we simply set

$$\tilde{f}_L(x) = f * P^0(x) + \sum_{\ell=1}^{L} d_\ell * P^\ell(x) = \sum_j f_j P^0(x, x_j) + \sum_{\ell=1}^{L} \sum_j d_{\ell;j} P^\ell(x, x_j)$$

where we directly extend the P^ℓ kernels for a new x as $P^\ell(x, x_j) = \frac{K^\ell(x,x_j)}{\sum_k K^\ell(x,x_k)}$ with $K^\ell(x, x') = \kappa \exp\left(-\|x - x'\|^2/\sigma_\ell^2\right)$, where $\sigma_\ell = \sigma/\mu^\ell$.

Note, however, that $\tilde{f}_\ell = f * P^\ell + \tilde{f}_{\ell-1} * (I - P^\ell)$ and since it can be shown that $P^\ell \to I$, the $\tilde{f}_\ell$ will overfit the sample points for large ℓ.

3 Auto-adaptive Laplacian Pyramids

k-fold Cross Validation (CV) is usually the standard choice to detect and prevent overfitting. In k-CV we randomly distribute the sample in k subsets and iteratively use $k - 1$ subsets for training and the remaining one for validation. In the extreme case when $k = N$, i.e., we use just one pattern for validation, this results in a Leave One Out Cross Validation (LOOCV) and stop the training iterations when the LOOCV error starts to increase. In our case LOOCV can be easily applied using for training a $N \times N$ normalized kernel matrix $P_{(p)}$ which is just the previous matrix P where we set to 0 the p-th rows and columns when x_p is held out of the training sample and used for validation. The most obvious drawback of LOOCV is its rather high cost, which in our case is $N \times O(LN^2) = O(LN^3)$.

In our context and to alleviate the LOOCV cost, notice first that when we remove x_p from the training sample, the test value at x_p of the $f^{(p)}$ extension built is defined by

$$\begin{aligned} f_L^{(p)}(x_p) &= \sum_{j \neq p} f_j P^0(x_p, x_j) + \sum_{\ell=1}^{L} \sum_{j \neq p} d_{\ell;j}^{(p)} P^\ell(x_p, x_j) \\ &= \sum_j f_j \tilde{P}^0(x_p, x_j) + \sum_{\ell=1}^{L} \sum_j d_{\ell;j}^{(p)} \tilde{P}^\ell(x_p, x_j), \end{aligned}$$

where $d_\ell^{(p)}$ are the different previously defined errors computed using the $P_{(p)}^\ell$ matrices and where $\tilde{P}$ is just the matrix P with its diagonal elements set to 0, i.e. $\tilde{P}_{i,i} = 0$, $\tilde{P}_{i,j} = P_{i,j}$ when $j \neq i$.

This observation leads to the modification we propose on the standard LP algorithm given in [3], and which simply consists of the application of the LP procedure described in Section 2, while replacing the P matrix by its 0-diagonal version $\tilde{P}$, computing then $\tilde{f}_0 = f * \tilde{P}^0$ at the beginning, and then the $\tilde{d}_\ell = f - \tilde{f}_\ell$, $\tilde{g}_\ell = \tilde{d}_\ell * \tilde{P}^\ell$ and $\tilde{f}_\ell$ vectors at each iteration. We call this algorithm the Auto-adaptive Laplacian Pyramid (ALP)[1].

[1]The ALP code is publicly available at `http://arantxa.ii.uam.es/~gaa/software.html`.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

According to the previous formula for the $f_L^{(p)}(x_p)$, we can take the ALP values $\tilde{f}_{L,p} = \tilde{f}_L(x_p)$ given by

$$\tilde{f}_L(x_p) = \sum_j f_j \tilde{P}^0(x_p, x_j) + \sum_{\ell=1}^{L} \sum_j \tilde{d}_{\ell;j} \tilde{P}^\ell(x_p, x_j),$$

as approximations to the LOOCV validation values $f_L^{(p)}(x_p)$. The LOOCV square error at each iteration can be approximated via

$$\sum_p (f(x_p) - f_L^{(p)}(x_p))^2 \simeq \sum_p (f(x_p) - \tilde{f}_{L,p})^2 = \sum_p (\tilde{d}_{L;p})^2,$$

which is just the training error of ALP at the current iteration.

The obvious advantage of ALP is that when we evaluate the training error, we are actually estimating the LOOCV error after each LP iteration. Therefore, the evolution of these LOOCV values tells us which is the optimal iteration to stop the algorithm, i.e., just when the training error approximation to the LOOCV error starts growing. Thus, we avoid overfitting and also use the training error as an approximation to the generalization error. This can be seen in Figure 1, which illustrates the application of ALP to the radiation forecasting example described in the next section. Observe how the optimum stopping time for ALP is exactly the same one that would be given by a full LOOCV error and how training error stabilizes afterwards.

Notice that the cost of running L steps of ALP is just $O(LN^2)$ and, thus, we gain the advantage of the exhaustive LOOCV without any additional cost. Moreover, ALP achieves an automatic selection of the width of the Gaussian kernel which makes this version of LP to be auto-adaptive, as it does not require costly parameter selection procedures. In fact, choosing $\mu = 2$, the only required parameter would be the initial σ but provided it is wide enough, its $\sigma/2^\ell$ scalings will yield an adequate final kernel width.

4 ALP for Radiation Forecasting

ALP can be used in manifold learning for extending coordinates but it can also be seen as an interpolation method and thus used for regression. In fact, both applications can be combined to, first, extend manifold learning embedding coordinates, such as Diffusion Maps (DM) [6], to new, unseen, test patterns and, second, to derive predictions over these extended coordinates. We will illustrate its application in one such problem, the prediction of actual solar radiation from Numerical Weather Predictions (NWP). This is directly related with the prediction of solar and, more generally, renewable energy, an area that is getting a growing attention from the Machine Learning community.

We will work with data from the AMS 2013-2014 Solar Energy Prediction Contest hosted by the Kaggle company [2] where the goal was the prediction of

[2]American Meteorological Society 2013-2014 Solar Energy Prediction Contest (`https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest`).

the total daily incoming solar energy in a series of meteorological stations located in Oklahoma.

The complete input data of the contest are, on one hand, an ensemble of 11 NWP from the NOAA/ESRL Global Ensemble Forecast System (GEFS), and on the other hand, the daily aggregated radiation from 98 stations in Oklahoma. For this experiment, we have just used the main NWP ensemble. Therefore, our input patterns contain five time-steps (from 12 to 24 UTC-hours in 3 hour increments) with 15 variables per time-step for all points in a 16×9 grid. The daily NWP forecasts from 1994–2004 yield the training patterns, and the years 2005, 2006 and 2007 are used for testing.

Before applying ALP for radiation forecasting, we have reduced the very large dimension of the NWP training variables using DM, that are able to maintain the essential information contained in the original data, while enabling the embedding coordinates to retain enough information for our predicting purposes. We then apply ALP over the DM coordinates obtained to extend them for the testing subset.

Recall that the main advantage of ALP is that its training error after each LP iteration also gives an approximate estimate of the LOOCV error, and we should stop when this training error starts growing. This can be observed for the radiation in Figure 1. The solid blue and dashed green lines represent the LP training error and the true LOOCV error per iteration respectively, and the dashed red line represents the error for our proposed method. As it can be seen, the ALP model requires 15 iterations to attains its minimum, about the same number of iterations that would be suggested by applying full LOOCV to standard LP. The figure also demonstrates the robustness of the ALP model against overfitting.

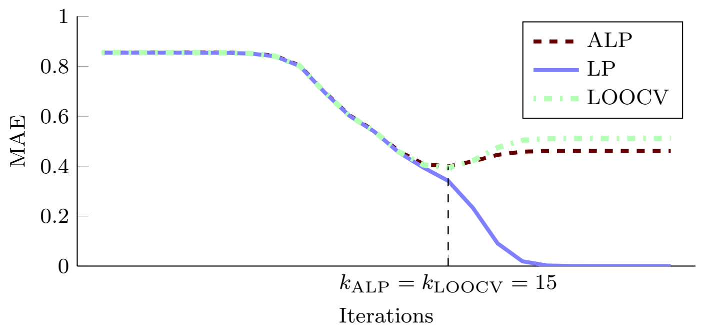

Fig. 1: Training errors for ALP, LP and LOOCV.

In Figure 2, left, we have depicted in red the result of applying this methodology for the second test year and the real radiation in blue. Although the winning models in the Kaggle competition followed different, better approaches, it can be seen that ALP captures radiation's seasonality. In the right plot we zoom in and it can be appreciated how ALP tracks the daily radiation variations and yields a reasonably good approximation to actual radiation values.

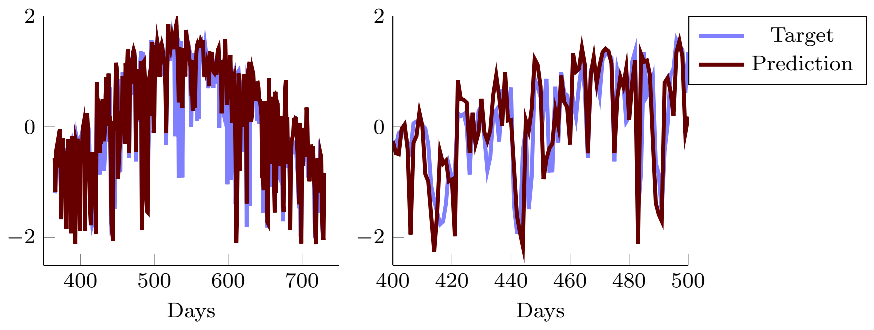

Fig. 2: Solar energy prediction over the second test year and over 100 days.

5 Conclusions

The classical and widely studied Laplacian Pyramid (LP) scheme of Burt and Adelson has been applied to many problems, particularly in image processing. However, it has a considerable overfitting risk and, thus, requires the use of rather costly techniques such as CV to prevent it. In this work we have presented Auto-adaptive Laplacian Pyramids (ALP), a modified, adaptive version of LP training that yields at no extra cost an estimate of the LOOCV value at each iteration, allowing thus to automatically decide when to stop in order to avoid overfitting. We have illustrated the robustness of the ALP method on a radiation forecasting example, where a two step ALP procedure has been applied to build a regression model over a DM embedding on the training set and then to apply it over the ALP estimation of the DM coordinates of the test samples. Further work will concentrate on other applications of ALP to manifold learning problems.

References

[1] A.D. Szlam, M. Maggioni, and R.R. Coifman. Regularization on graphs with function-adapted diffusion processes. *J. Mach. Learn. Res.*, 9:1711–1739, 2008.

[2] R.R. Coifman and S. Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21(1):31–52, 2006.

[3] N. Rabin and R.R. Coifman. Heterogeneous datasets representation and learning using Diffusion Maps and Laplacian Pyramids. In *SDM*, pages 189–199. SIAM/Omnipress, 2012.

[4] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[5] E. Chiavazzo, C.W. Gear, C.J. Dsilva, N. Rabin, and I.G. Kevrekidis. Reduced models in chemical kinetics via nonlinear data-mining. *Processes*, 2(1):112–140, 2014.

[6] R.R. Coifman and S. Lafon. Diffusion Maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.