CS233: The Shape of Data Handout # 6
Geometric and Topological Data Analysis
Stanford University Wednesday, 25 May 2016

Homework #4 v4: Joint Data Analysis, Entity Extraction. [100 points]
Due Date: Thursday, 2 June 2016

In this assignment you will familiarize yourself with the paradigm of *joint data analysis* in a setting that involves two-dimensional image data. Concretely, given a collection of photographs that have some common content or object, e.g., a cow or cows appear in all of them, you will aim to discover this shared object in every image. Apart from the fact that all the photos share a common object (a form of very weak supervision), no other source of human supervision will be provide — making the problem very challenging. This problem, of *unsupervised co-localization*, has culminated a lot of research in the last few years ([1], [2], [3]). Your solution will follow one of the approaches, namely, the paradigm of networks of functional maps and their corresponding latent spaces ([3], [4]) which will be covered in class.

Problem 1. Functional Map Ingredients [50 points]

In order to establish a functional map between two data sets, one first needs to derive a few building blocks associated with each individual data set, as well as to commit on certain design options. The first building block you will need to derive for every image of the input collection is a specialized functional basis which will be used to express all the later derived quantities of interest. The specific basis that you will compute, will be comprised by Laplacian-type eigenvectors stemming from super-pixel (SP) graphs, one such graph associated with every image. Given an image, we will compute its decomposition into super-pixels (See Figure 1). The graph associated with this image will be simply the dual graph of this super-pixel decomposition. For every image containing the shared object we will pre-compute its SP decomposition and provide you with the adjacency matrix of its dual graph. Each provided SP graph has on the average 160 nodes representing the superpixels; its edges are weighted by the length of the shared boundary among neighboring super-pixels.

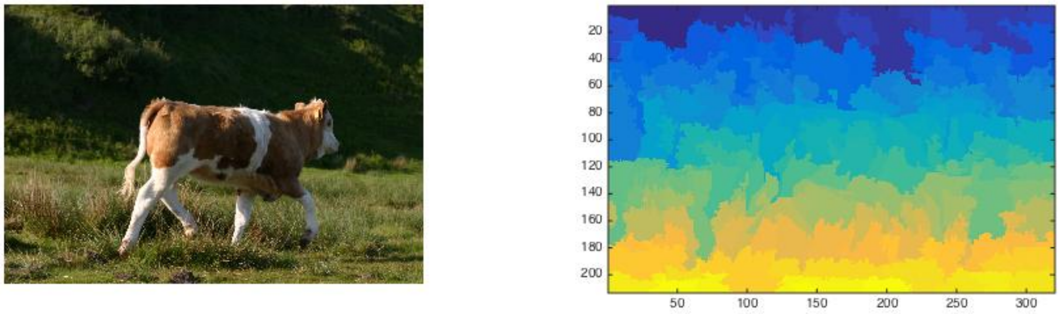

Figure 1: Image portraying a cow (left) and its corresponding super-pixel decomposition (right). The super-pixels are distinguished by their color.

Your first three tasks are as follows:

1. (5 points) Load in memory all the images of the cow collection, along with the adjacency matrices of their corresponding SP graphs. Derive from every adjacency matrix A, its normalized Laplacian: $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where I is the identity matrix, D is a diagonal matrix, with $D_{ii} = \sum_j A_{ij}$.

2. (2.5 points) Compute for every Laplacian, the 64 eigenvectors corresponding to the eigenvalues with the smallest magnitudes.

3. (2.5 points) What is the magnitude of the smallest eigenvalue found? What is its algebraic multiplicity? Can you mention a property of the SP graphs that one can infer by looking at these numbers?

The next building block required by functional maps (F-Maps) will be feature-vectors, associated with every super-pixel of an image. These feature vectors, which we can think of as functions living on the SP graph, will effectively be the cues that an F-map relies upon in order to discover interesting relationships between a pair of images. For instance, a magic feature vector would be a scalar indicator function indicating which super-pixels belong to the shared object or not. Having such a feature for a pair of images would naturally provide us with the functional-relation that would align the shared object and the corresponding backgrounds. Of course, this magic feature **is** the solution to our posed problem and we do not have it. Instead, we will exploit and use some other types of features which hopefully will enable an approximate solution. The first set of features that you will construct belongs in the "bag-of-words" family, which is known to capture several discriminative aspects of the underlying image data.

4. (15 points) Follow the example described here [1] to create a 300-dimensional visual dictionary trained with the images of the three provided classes: cows, dogs and flowers. Use 70% of images from each set as the training data and the remainder, 30%, as the validation data. Train the same SVM classifier as in the example and report the confusion matrix on the validation set.

5. (2.5 points) Redo the previous experiment by considering *all* the images as training data. For the remaining parts of the homework, utilize this latter generated dictionary.

The above mentioned dictionary tries to understand the semantics of an image that are good for discriminating the class of its main portrayed object from other objects. In order to use it and generate a feature for a super-pixel, you will need to call the MATLAB function `encode` (again, see the example). The function `encode` expects as input a dictionary and an image, which for RGB images translates to a 3D-matrix. To find the image content that corresponds to every super-pixel load the provided mask files that label for every pixel of an image to which SP it belongs. One technicality you need to handle stems from the fact that most super-pixels are not rectangular regions and thus you can't just use their pixel content with encode as is. Instead:

[1] http://www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html

6. (7.5 points) Use the tightest axis-aligned bounding box that encloses an SP when calling encode. In such a bounding-box pixels that are not part of the SP should take the the RGB value [0,0,0] (See Figure 2)

Another useful family of image features that you will explore, is solely based on the color content of an SP.

7. (2.5 points) For every SP derive a 3-dimensional feature capturing the mean RGB values of its content.

8. (5 points) Make a color-histogram with 36 bins that represents the color values present in each SP. For this task, treat the content (RGB) of an SP as a single set of values. Also, normalize the histogram to represent a probability distribution (e.g., see `histcounts`).

By aggregating the previously derived features into a single vector, every super-pixel will have a corresponding 339-dimensional feature vector. To fix some notation, assume that for a given image i, we have stacked the vectors of its SPs in a single matrix $F_i \in \mathbb{R}^{K_i \times 339}$, where K_i is the number of SPs of image i. Denote also the reduced basis you computed for that image with B_i.[2] In order to speed up the computations (among other reasons), when deriving an F-Map we will utilize a compressed version of F_i. Namely, we will use $\hat{F}_i = B_i \circ F_i$, where $B_i \circ F_i$ indicates the projection of F_i in the vectors B_i.

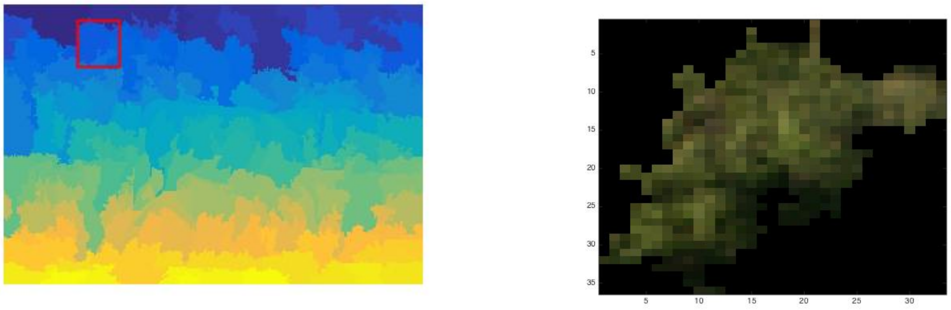

Figure 2: Axis aligned tightest bounding-box of a super-pixel (red box on left image). Zoomed image of color-content of the super-pixel. Pixels of the box that were not part of the SP are colored black.

9. (5 points) For every image i, compute $\hat{F}_i$ by varying the number of eigenvectors constituting B_i from 1 to 64. Start with the eigenvector corresponding to the smallest eigenvalue and at each step include in B_i the eigenvector with the next smallest eigenvalue. Plot the average over all cow images of the quantity $E = \frac{||B_i \times \hat{F}_i - F_i||}{||F_i||}$ as you vary the eigenvectors from 1 to 64. E represents the reconstruction error we have with each basis. Use the Frobenius matrix norm when computing E and also separately report the average reconstruction error with 32 eigenvectors.

[2] The nodes of the provided SP-graphs are ordered in the same manner as the SPs.

10. (2.5 points) Computing the projection in the previous task is a rather simple action, due to a basic property of the Laplacian eigenvectors. Which property is this? In other words, can you compute it without using the backslash (\) MATLAB operator?

Problem 2. Functional Maps in Action [50 points]

Given all the ingredients you have computed in the first problem you are now ready to derive a Functional Map between any pair of images in your collection. How to combine the ingredients to derive an F-Map is closer to an art than to science. Nevertheless, a way to do it which has shown good properties in practice ([3]) is by solving the following convex program:

$$X_{ij} = \underset{X}{\arg\min} ||X\hat{F}_i - \hat{F}_j||_F^2 + \mu ||XD_i - D_jX||_F^2 \tag{1}$$

where:

- X_{ij} is the F-map from image i to image j.
- $\hat{F}_i$ are the projected SP features of image i in the corresponding basis B_i in accordance to Problem 1.
- D_i is a diagonal matrix storing the corresponding eigenvalues of B_i. The eigenvalues are sorted by the same order as the eigenvectors.
- μ is a constant controlling the importance between the two summands.

A natural question to ask is for which pairs of images one should compute an F-Map. If the two images are significantly different, a meaningful map from one another seems unlikely. On the other hand, two identical images have only a trivial map. In this assignment we used the GIST [5] descriptor, which attempts to assess the similarity of two arbitrary scenes, to answer the aforementioned question. Concretely, we have provided you with a sparse graph connecting every image to its top-5 GIST neighbors (`gist_graph.mat`).

Your next two tasks are as follow:

1. (25 points) Compute the two F-Maps X_{ij} and X_{ji} corresponding to every edge (i,j) of the provided graph. To solve for the F-Maps optimize (1) e.g., via CVX. Use the 32 eigenvectors corresponding to the smallest eigenvalues to form the basis of every image. Also, use $\mu = 20$.

2. (7.5 points) For some edges of your liking compute again the F-Maps but this time with $\mu = 0$. Visualize and inspect the resulting F-Maps (i.e., plot the matrices with `imagesc`). Can you see a *structural* difference between those and the corresponding F-Maps computed with $\mu = 20$? Include a snapshot of two F-Maps that captures this difference in your write-up. The second term of (1) attempts to enforce in a soft way an invariance discussed class. Can you recognize which one it is?

Having finished all the previous parts you have now a network of images connected with edges that are decorated with pairwise Functional Maps. In this last part of the assignment you will try to extract a set of functions that seem to be highly *transportable* across these edges — in other words, a fixed point of the network. Our hope is that a function that can be transferred across the network in a transparent way captures the commonalities found among the images. The notion of transportability that we will use to assess the quality of function wrt. the network of F-Maps is its *cycle-consistency*. On a high level given, a homogeneous network of data, a function is cycle-consistent if when transported across any closed path of edges (cycle) it is "well preserved". Equation (2) is one possible functional that promotes cycle-consistency. In that equation every edge (i, j) of your network (G), is associated with a term that evaluates how well its corresponding F-Map aligns two sets of latent functions Y_i and Y_j. These functions are latent since they are not given to us, but instead we assume their existence due to the commonalities shared by our images.

$$f^{cons} = \sum_{(i,j)\in G} f^{cons}_{ij} = \sum_{(i,j)\in G} ||X_{ij}Y_i - Y_j||^2_F \tag{2}$$

Specifically, with the F-Maps decorating the edges regarded as fixed, one can find a unique set of functions Y_i for every image i, that would make f^{cons} minimal. To this end one would have to extract the spectra of a block-diagonal matrix W, that glues together all the pairwise F-Maps:

$$W_{ij} = \begin{cases} \sum\limits_{(i,j')\in G} I + X^T_{ij'}X_{ij'} & \text{for } i = j \\ -(X_{ji} + X^T_{ij}) & (i, j) \in G \\ 0 & \text{otherwise} \end{cases}$$

where I denotes the identity matrix of the appropriate size.

Your final tasks is as follows:

3. (17.5 points) Construct W and extract its smallest in magnitude eigenvector. This eigenvector naturally gives rise to 32 co-efficients c_i, who encode a single latent/consistent function $y_i = B_i c_i$, for every image i. Explain how to find c_i for every image and construct y_i. Plot y_i for every image and include in your report the 4 best cases for which y_i seems to delineate the underlying cow. Also include a case where this doesn't happen. Notice, that since B_i by construction expresses functions on super-pixels, you will first need to convert y_i to a function defined on the pixel domain before you plot it. Be creative.

Extra Credit

1. (2.5 points) Describe five sources of variations found in natural images, that make the co-localization problem hard. In other words, if we were looking for exactly the same 'version' of a cow in all images, the problem would be much easier.

2. (2.5 points) What could it be a benefit of using in our analysis super-pixels of images instead of working directly with pixels? I.e., by deriving a basis, features and corresponding F-Maps on the super-pixels.

3. (10 points) One way to sharpen each consistent function y_i that you found on the third question of problem two (P2-3), is by exploiting the discriminative nature of the underlying Laplacian L_i. Redo P2-3, but this time solve for and use the second eigenvector of the matrix $Z = \frac{1}{10}\,\mathrm{diag}(D_i) + W$. Z is a copy of W where we have added to each of its diagonal blocks the corresponding scaled diagonal matrix of eigenvalues D_i. Plot the newly found consistent functions corresponding to the 5 images you reported on P2-3. What is the aforementioned discriminative nature of the Laplacian that helped to sharpen the produced final functions? [Hint: Same reason as why "graph-cuts" [6] work.]

References

[1] F. B. A. Joulin and J. Ponce, “Discriminative clustering for image co-segmentation,” in *CVPR*, 2010.

[2] C. S. M. Cho, S. Kwak and J. Ponce, “Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals.,” in *CVPR*, 2015.

[3] F. Wang, Q. Huang, and L. Guibas, “Image co-segmentation via consistent functional maps,” in *ICCV*, 2013.

[4] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, “Functional maps: A flexible representation of maps between shapes,” in *SIGGRAPH*, 2012.

[5] A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” in *IJCV*, 2001.

[6] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE TPAMI*, vol. 22, pp. 888–905, 2000.