Meeting Notes 11/8/2016

Geoffrey Iyer

November 8, 2016

I spent a lot of time this week thinking about what assumptions we should make about our data set, and what our classification goals should be. For now, I'm working entirely theoretically. I haven't yet thought about how to make this computationally feasible. Here's what I came up with in the end.

Assumptions

We have two data sets, X_1, X_2, both sampling the same underlying scene. There exists a bijective correpondence $X_1 \to X_2$ based on how the data was sampled. For example: we could have X_1, X_2 are both timeseries images of the same scene, taken from different angles. In this case the correspondence would be based on the time the image was taken.

I also assume that we have already determined some number of correspondence pairs m, where $m << |X|$. I notated these as $\{p_1, \ldots, p_m\} \subseteq X_1$ and $\{q_1, \ldots, q_m\} \subseteq X_2$.

Lastly, I assume that the data sets X_1, X_2 have somehow been normalized so that distances between points are roughly comparable between sets.

Goal

With these assumptions, I believe we can construct a latent space X and mappings $X_1, X_2 \to X$ that allow us to extract information from the combined data that we could not get from either individual set. In particular, we are interested in differences between two data points that can be seen through one set, but not the other.

Algorithm

1. First, we attempt to reconstruct the full point correspondence between X_1, X_2 using the distinguished pairs $\{p_j\} \subseteq X_1$ and $\{q_j\} \subseteq X_2$. Let $\rho : X_1 \to X_2$ be a bijection that preserves our distinguished pairs $\{p_j\}, \{q_j\}$. We'll define an energy $E(\rho)$ that we can minimize to find the optimal bijection. For notation, let

$$X_1 - \{p_1, \ldots, p_m\} = \{x_1, \ldots, x_n\}.$$

Then define the energy

$$E(\rho) = \sum_{i=1}^{m} \sum_{j=1}^{n} \max\left(\|p_i - x_j\|, \|q_i - \rho(x_j)\|\right).$$

I'm honestly not too confident about this choice for E. It seems like it will give reasonable results, but it's hard to say if this is the best (especially compared to using $+$ over max). The idea here is that when we sample from our source to obtain X_j, we lose some information. So if two elements are close together in X_j, that doesn't necessarily mean that they are close in the original sample. However, if points are far apart in X_j, then they should be considered far apart in the original sample.

If we somehow find a ρ that minimizes the above energy, we proceed to step 2.

2. Now we have constructed a full correspondence ρ we move on to the creation of our latent space X. Our goal is to embed X_1, X_2 into X while preserving

$$\max\left(\|x_i - x_j\|, \|\rho(x_i) - \rho(x_j)\|\right).$$

That is, if our points are considered close together in X_1, and far apart in X_2, they should be thought of as far apart in our latent space X. I didn't get very far in my experiment with this, but it seems like we should be able to do this by mimicking the idea from [1]. Knowing the correspondence pairs let's us put extra weight in our Graph Laplacian towards keeping those points together.

Toy Example

This section is going to be rushed. I didn't have time to make my examples look nice. Here are pictures of the ground truth data (figure 1), X_1 (figure 2), and X_2 (figure 3). The ground truth contains 3 gaussian clouds (with 3 points each). X_1 and X_2 are the projections onto the xy and xz planes (respectively).

For starting assumptions, I picked a one point from each cloud to be a known correspondence pair (so the m above is 3. One from each cloud). I think minimized the energy from step 1 by explicitely calculating it for each permutation (so 6! total calculations). It usually did okay, but rarely over got the exact correct correspondence from the ground truth. Sounds like I need to rework my idea.

For step 2, I assumed that I already had the correct correspondence from the ground truth (since step 1 wasn't working too well). Then I followed the idea from [1] and built a weight matrix that tracked when points were known to correspond (I called it W_s). As well as the standard graph laplacian on each individual X_j (I called the concatenated matrix W). Constructing the corresponding graph laplacian $L + L_s$ and doing the standard eignvector analysis yielded okay results. It mostly clusted the 3 clouds together, but usually there were some strange outliers. So it's certainly not a perfect method but there was some promise.

References

[1] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. 2

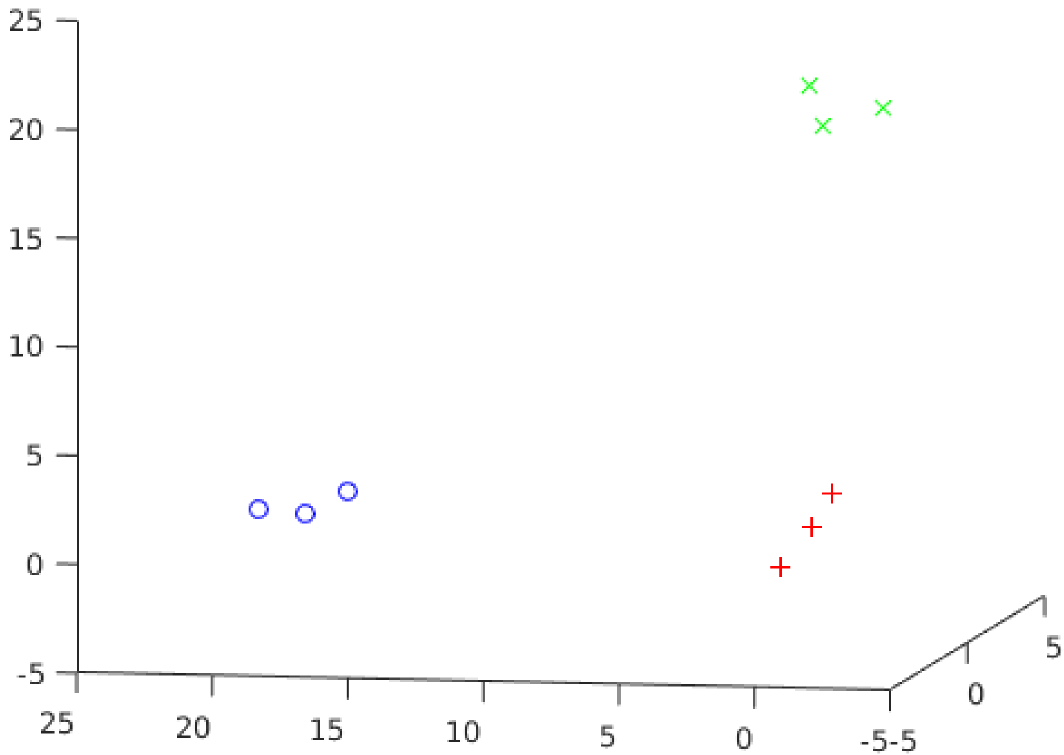

Figure 1: Ground Truth

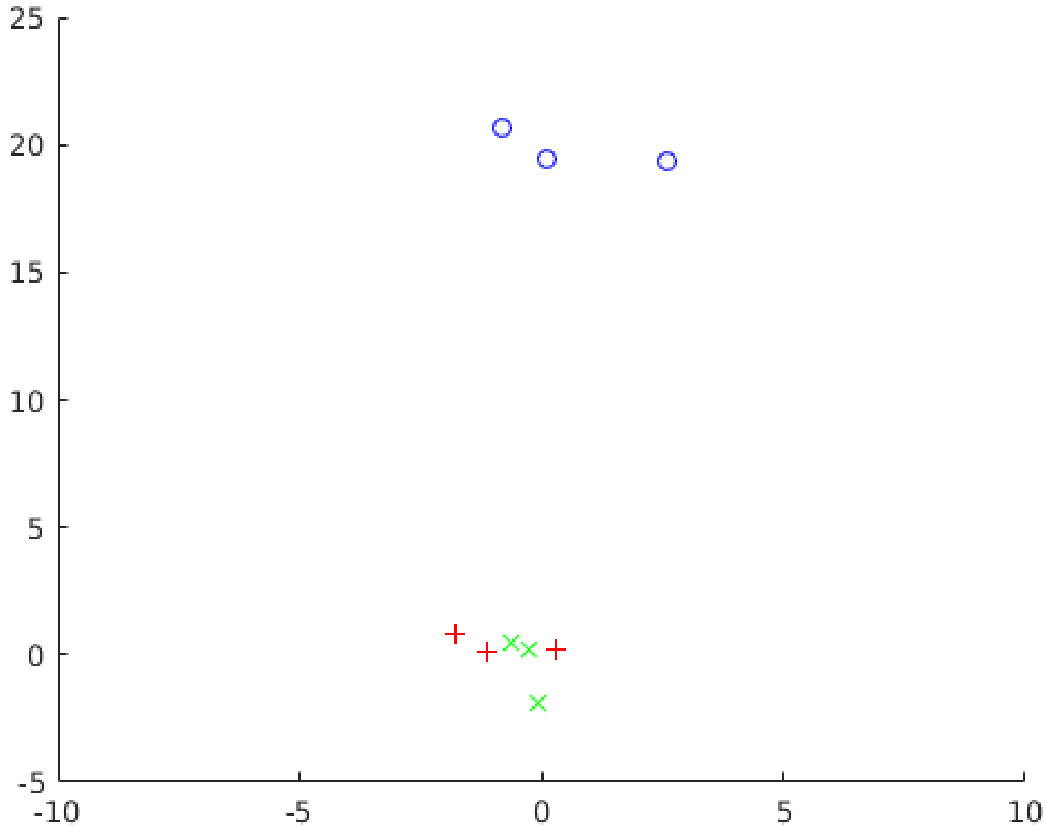

Figure 2: X_1

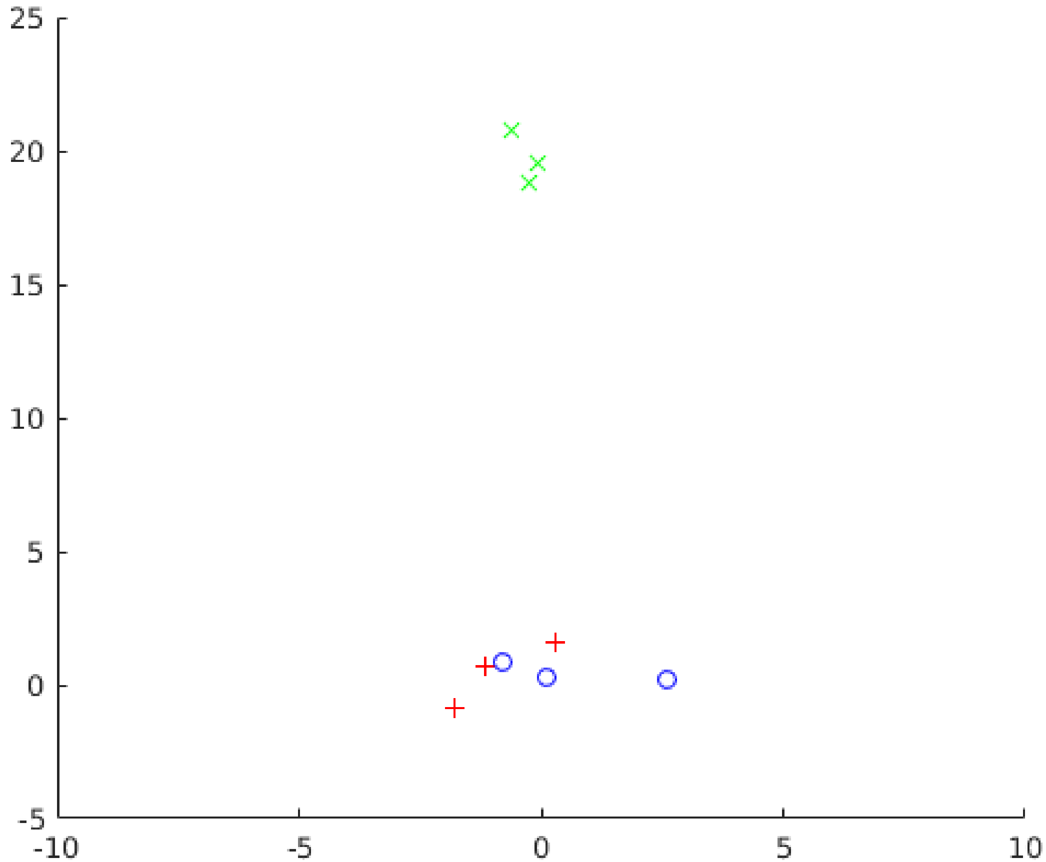

Figure 3: X_2