Robust Projection

August 7, 2018

1 Experiment

Given a data in $X \in \mathcal{R}^{d\times N}$ (where N is the number of data points and d is the data dimension), and K-dimensional subspace $V \in \mathcal{R}^{d\times k}$ that is learnt from the data, the projection of a new data point $x \in \mathcal{R}^d$ on the subspace is VV^Tx. This operation is not robust, as it can be shown that:

$$VV^Tx = \underset{\hat{x}\in span(V)}{\operatorname{argmin}} \|\hat{x} - x\|^2$$

For instance, as the number of outlier-pixels in X grows, the reconstruction error $(I - VV^T)x$ is bigger and has more impact on the projection of inlier-pixels.
We would like to reduce the effect that the outlier-pixels have on the projection of the data point as a whole. We decided to assign weights to each data point (pixel) in the projection process such that outlier-pixels will get low weights and inlier-pixels will get high weight. We get a WLS equation:

$$\hat{\alpha}_{WLS} = \underset{\alpha}{\operatorname{argmin}} \left\| W^{\frac{1}{2}}(V\alpha - x) \right\|^2 \qquad W^{\frac{1}{2}}_{d\times d} = diag(w_1, ..., w_d), \quad \alpha \in \mathcal{R}^k$$

Where its solution holds:

$$V^TWV\hat{\alpha}_{WLS} = V^TWx$$

2 Model

Graph-Cut + Ising Model
We assume that all of the outlier-pixels are related to objects in the image and we will refer to them as foreground pixels (FG).

Using spatial prior (2D ising): we assume that outlier-pixels tend to be close to each other (following the previous assumption).
Using temporal prior (3D ising): we assume that pixels from consecutive frames tend to be similar to each other.
The model is MRF:
Version 1 (2D Ising model):

$$exp\left(\frac{1}{T}\sum_{s\sim s'} x_s^t x_{s'}^t\right) + \mathbb{1}_{x_s=BG}\left(\frac{1}{\sigma^2}\left(y_s - (V\alpha)_s\right)\right) + \mathbb{1}_{x_s=FG}\left(\frac{y_s^2}{\sigma_Y^2}\right)$$

Where:

$$y_s = x_s + \eta_s, \qquad \eta_s \sim \mathcal{N}(0, \sigma_n^2 I_{d\times 1})$$

Version 2 (3D Ising model):

$$exp\left(\frac{1}{T}\sum_{s\sim s'} x_s^t x_{s'}^t\right) + \mathbb{1}_{x_s=BG}\left(\frac{1}{\sigma^2}\left(y_s - (V\alpha)_s\right)\right) + \mathbb{1}_{x_s=FG}\left(\frac{y_s^2}{\sigma_Y^2}\right) + \sum_s x_s^t x_s^{t+1}$$

Version 3 (Use w instead of x):

$$exp\left(\frac{1}{T}\sum_{s\sim s'} w_s^t w_{s'}^t\right) + \mathbb{1}_{x_s=BG}\left(\frac{1}{\sigma^2}\left(y_s - (V\alpha)_s\right)\right) + \mathbb{1}_{x_s=FG}\left(\frac{y_s^2}{\sigma_Y^2}\right) + \sum_s x_s^t x_s^{t+1}$$

Alternate between $P(W|Y,\alpha)$ and $P(\alpha|Y,W)$
Compute $P(W|Y,\alpha)$ using Graph-Cut.
Compute $P(\alpha|Y,W)$ using WLS.

Graph-Cut Computation:
For each pixel s we have 2 parameters: the weight: w_s, and (V*alpha)_s which is the entry in the projection that corresponds to the pixel s. We're looking on a graph that its nodes are the pixels, plus nodes s,t.
For edges between pixels-nodes, need to assign Ising model weights (1st term in the model).
For edges between the source or sink to all nodes, need to assign the terms in the posterior that's related to the data we have. My idea is to use the 2nd term as weights from the source to all nodes, and the 3rd term as weights from all nodes to the sink. They will have 2 different probabilities, which depend if x_s is FG or BG.