Supervised Stochastic Triplet Embedding

November 10, 2015

1 Projected Gradient

After we calculate the gradient using a loss function (in this case, the loss for stochastic triplet embedding, STE), we get an $n \times n$ matrix $\mathbf{K}$. This matrix needs to be positive semidefinite (having no negative eigenvalues) in order to be a valid kernel. STE uses projected gradient descent - after each gradient step at iteration t, where we update the variable $\mathbf{K}^t$, we then project $\mathbf{K}$ to the closest matrix in the set of positive semidefinite matrices, which is a closed convex set (a cone), giving the following optimization problem

$$\begin{array}{cc} \text{minimize} & \rho \left|\left|\mathbf{K} - \mathbf{X}\right|\right|_F^2 \\ \text{s.t} & \mathbf{X} \succeq 0, \end{array} \tag{1}$$

which we can solve by calculating the eigenvalue decomposition of $\mathbf{K}$ and setting all negative eigenvalues to zero.

In metric learning, we want to impose a stronger restriction on $\mathbf{K}$ - instead of being simply psd, we additionally require that the dot products are a function of some features, $\mathbf{K} = \mathbf{F}^T\mathbf{M}\mathbf{F}$. There are many algorithms for metric learning, but it's simple to adapt projected gradient to this problem. Given our feature matrix $\mathbf{F}$, we solve a very similar optimization problem,

$$\begin{array}{cc} \text{minimize} & \rho \left|\left|\mathbf{X} - \mathbf{K}\right|\right|_F^2 \\ \text{s.t} & \mathbf{X} = \mathbf{F}^T\mathbf{M}\mathbf{F} \\ & \mathbf{M} \succeq 0 \end{array} \tag{2}$$

which we can solve in almost as simple a way, after some linear algebra. Below we derive the updates for the objective function, and then we can again require that $\mathbf{M}$ be psd, again by projection.

$$\left|\left|\mathbf{K} - \mathbf{F}^T\mathbf{M}\mathbf{F}\right|\right|_F^2 = \tag{3}$$

$$\operatorname{Tr}\left((\mathbf{K} - \mathbf{F}^T\mathbf{M}\mathbf{F})^T(\mathbf{K} - \mathbf{F}^T\mathbf{M}\mathbf{F})\right) \tag{4}$$

$$\operatorname{Tr}\left(\mathbf{K}^T\mathbf{K} - \mathbf{F}^T\mathbf{M}\mathbf{F}\mathbf{K} - \mathbf{K}^T\mathbf{F}^T\mathbf{M}\mathbf{F} + \mathbf{F}^T\mathbf{M}\mathbf{F}\mathbf{F}^T\mathbf{M}\mathbf{F}\right) \tag{5}$$

$$\operatorname{Tr}\left(\mathbf{K}^T\mathbf{K}\right) - 2\operatorname{Tr}\left(\mathbf{F}^T\mathbf{M}\mathbf{F}\mathbf{K}\right) + \operatorname{Tr}\left(\mathbf{F}^T\mathbf{M}\mathbf{F}\mathbf{F}^T\mathbf{M}\mathbf{F}\right) \tag{6}$$

where we use a few properties of trace and the fact that all the relevant matrices are symmetric.

We can solve this gradient analytically, first by taking derivatives and then setting them to zero

$$-2\mathbf{F}^T\mathbf{K}\mathbf{F} + 2\mathbf{F}^T\mathbf{F}\mathbf{M}\mathbf{F}^T\mathbf{F} = 0 \tag{7}$$

$$\mathbf{F}^T\mathbf{K}\mathbf{F} = \mathbf{F}^T\mathbf{F}\mathbf{M}\mathbf{F}^T\mathbf{F} \tag{8}$$

$$\tag{9}$$

Then we multiply both sides by the inverse of $(\mathbf{F}^T\mathbf{F})$ [1].

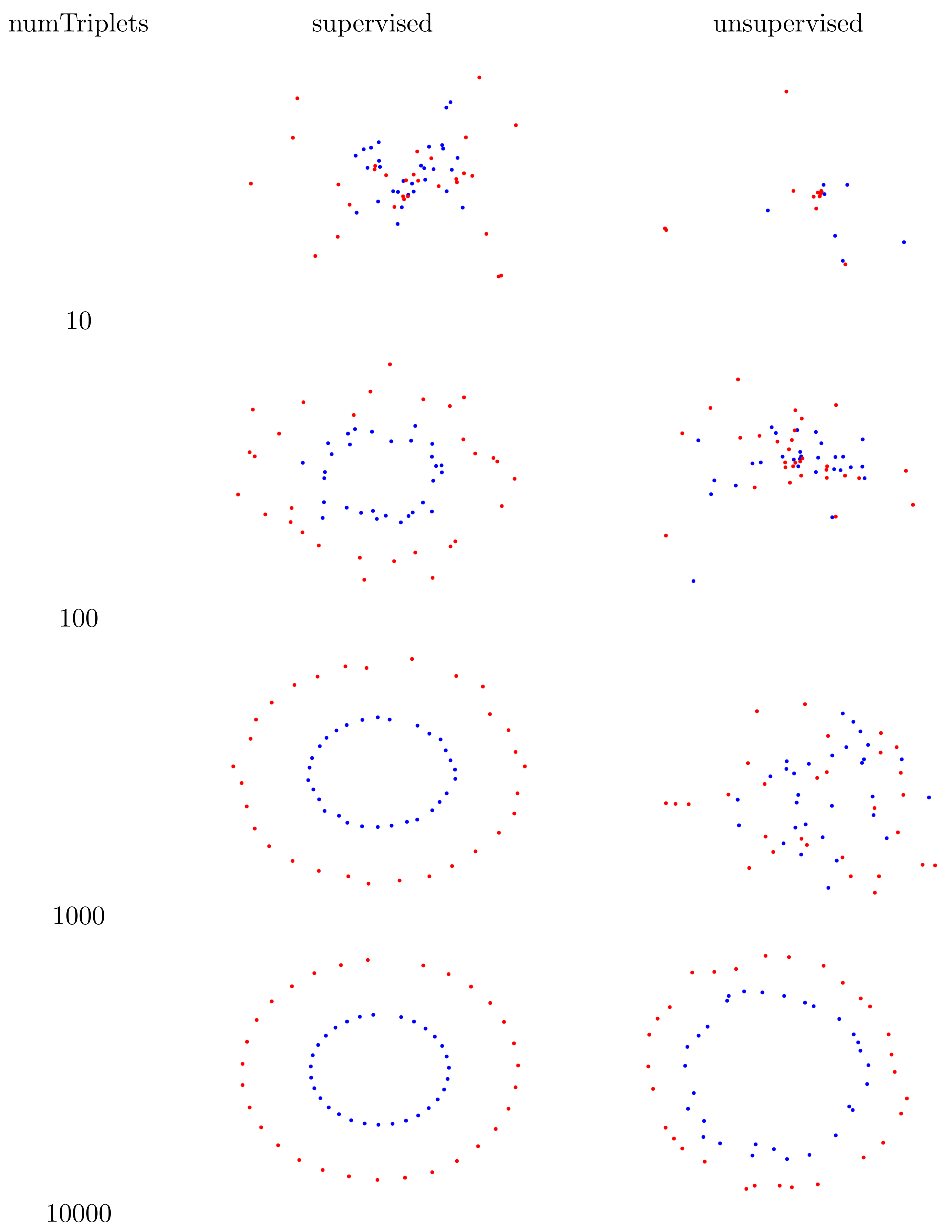
$$\mathbf{M} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\mathbf{K}\mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1} \tag{10}$$

Note that this is the sample covariance, and the inverse is sometimes known as the precision matrix. The update is just a linear function of $\mathbf{K}$, and in general we can't expect $\mathbf{K}$ to be positive semidefinite, so we still need to make sure that $\mathbf{M}$ is positive semidefinite. Note that the main work in this step is inversion of $\mathbf{F}^T\mathbf{F}$, and this never changes as we iterate, so this can be cached.

2 Results

In this section I compare supervised STE with unsupervised STE. Unsurprisingly, it does much better when it has features, because there are fewer parameters to learn: instead of coordinates for each point in a possibly high dimensional space, it has to learn how to rotate and scale the features - the size of the matrix $\mathbf{M}$ only depends on the number of features, not the number of data points.

[1] TODO : check this again with the pseudoinverse to make sure it's still true


numTriplets
supervised
unsupervised
10
100
1000
10000