t-SNE Notes

Pavlin Poličar

1 Fast KL Divergence

During computation of negative gradients, we do not know the value of the normalization term Z during intermediate steps. Therefore, in order to compute the KL divergence of the embedding, we would need to iterate over all the p_{ij}s again, at the end of the gradient computation step. By rewriting the KL divergence in terms of unnormalized q_{ij}s, we can save on computation time.

$$
\begin{aligned}
KL(P \mid\mid Q) &= \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \\
&= \sum_{ij} p_{ij} \log \left(p_{ij} \frac{Z}{\hat{q}_{ij}} \right)
\end{aligned}
$$

where $\hat{q}_{ij}$ denotes the unnormalized values q_{ij}

$$
= \sum_{ij} p_{ij} \log \frac{p_{ij}}{\hat{q}_{ij}} + \sum_{ij} p_{ij} \log Z
$$

2 KL Divergence with exaggeration

The implemented optimization methods don't have a notion of exaggeration, they simply take an affinity matrix P containing the probabilities of points j appearing close to i. Exaggeration is used to scale P by some constant factor α to help separate clusters in the beginning of the optimization. These methods also compute the KL divergence, and as such, it is incorrect because we don't account for α.

This section derives a quick and simple correction for the KL divergence error term so we can get the true error of the embedding even when P is exaggerated.

$$
KL(P \mid\mid Q) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{1}
$$

We need to introduce the scaling i.e. exaggeration factor α

$$
= \sum_{ij} \frac{\alpha}{\alpha} p_{ij} \log \frac{\alpha p_{ij}}{\alpha q_{ij}} \tag{2}
$$

Exaggeration means that the p_{ij} terms get multiplied by α, so we need to find an expression for the KL divergence that includes only αp_{ij} and q_{ij} and some other factor that will correct for α.

$$= \frac{1}{\alpha} \sum_{ij} \alpha p_{ij} \left(\log \frac{\alpha p_{ij}}{q_{ij}} - \log \alpha \right) \tag{3}$$

$$= \frac{1}{\alpha} \left(\sum_{ij} \alpha p_{ij} \log \frac{\alpha p_{ij}}{q_{ij}} - \sum_{ij} \alpha p_{ij} \log \alpha \right) \tag{4}$$

The first term is computed by the negative gradient method (since it only knows about the scaled P), the second term can easily be computed post-optimization, allowing us to get the correct KL divergence.