Visualising High Dimensional Data with t-SNE

Harri Edwards

t-SNE???

- t-SNE is a technique for transforming high-dimensional data

  $$x_1, \dots, x_k \in \mathbb{R}^{\mathrm{D}>3}$$

  to lower dimensional 'map points'

  $$y_1, \dots, y_k \in \mathbb{R}^{d \leq 3}$$

  for the purpose of visualisation.

t-SNE???

- t-SNE is a technique for transforming high-dimensional data

$$x_1, \ldots, x_k \in \mathbb{R}^{\mathrm{D}>3}$$

  to lower dimensional 'map points'

$$y_1, \ldots, y_k \in \mathbb{R}^{d \leq 3}$$

  for the purpose of visualisation.
- Desideratum for a mapping is to *preserve the neighbourhoods* as much as possible, *i.e.* if x_i, x_j are 'close' then y_i, y_j should be 'close' and vice-versa.

t-SNE???

- t-SNE is a technique for transforming high-dimensional data

$$x_1, \ldots, x_k \in \mathbb{R}^{\mathrm{D}>3}$$

  to lower dimensional 'map points'

$$y_1, \ldots, y_k \in \mathbb{R}^{d \leq 3}$$

  for the purpose of visualisation.
- Desideratum for a mapping is to *preserve the neighbourhoods* as much as possible, *i.e.* if x_i, x_j are 'close' then y_i, y_j should be 'close' and vice-versa.
- t-SNE comes from the paper 'Visualizing Data using t-SNE' published in 2008 by Laurens van der Maaten and Geoff Hinton (1127 citations!)

SNE

- t-SNE is a modification of an earlier technique called ‘Stochastic Neighbour Embedding’ from Hinton and Rowesis (2002).

SNE

- t-SNE is a modification of an earlier technique called 'Stochastic Neighbour Embedding' from Hinton and Rowesis (2002).

The basic idea of SNE is very simple: for each data point x_i, we specify a probability distribution over its possible neighbours x_j with $j \neq 0$

$$p_{i|j} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{j \neq i} \exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)} \tag{1}$$

SNE

- t-SNE is a modification of an earlier technique called 'Stochastic Neighbour Embedding' from Hinton and Rowesis (2002).

The basic idea of SNE is very simple: for each data point x_i, we specify a probability distribution over its possible neighbours x_j with $j \neq 0$

$$p_{i|j} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{j \neq i} \exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)} \tag{1}$$

Now we do the same for the map points:

$$q_{i|j} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{j \neq i} \exp\left(-\|y_i - y_j\|^2\right)} \tag{2}$$

SNE: Cost

So we have a probability distribution p over the neighbourhood structure of the x's and a probability distribution q over the neighbourhood structure of the map points.

SNE: Cost

So we have a probability distribution p over the neighbourhood structure of the x's and a probability distribution q over the neighbourhood structure of the map points.
Making a visualisation that preserves the neighbourhood structure then consists in learning a q that matches p. The parameters to be learned are the y_i, and we measure the goodness of fit by the KL-divergence

$$C = \sum_i KL(p_{(\cdot|i)} \| q_{(\cdot|i)}) \tag{3}$$

SNE: Cost

So we have a probability distribution p over the neighbourhood structure of the x's and a probability distribution q over the neighbourhood structure of the map points.
Making a visualisation that preserves the neighbourhood structure then consists in learning a q that matches p. The parameters to be learned are the y_i, and we measure the goodness of fit by the KL-divergence

$$C = \sum_i KL(p_{(\cdot|i)} \| q_{(\cdot|i)}) \tag{3}$$

that is

$$C = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{4}$$

SNE: Gradient

The gradient of the cost function has a nice and simple form:

$$\frac{\partial C}{\partial y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j} \right) \left(y_i - y_j \right) \tag{5}$$

SNE: Gradient

The gradient of the cost function has a nice and simple form:

$$\frac{\partial C}{\partial y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j} \right) \left(y_i - y_j \right) \tag{5}$$

The optimisation of C involves initialising the map points from a spherical Gaussian, using gradient descent with momentum, and adding noise to the map points on each iteration with annealing variance.

The Crowding Problem

'High dimensional space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space.' (Euclid)

The Crowding Problem

'High dimensional space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space.' (Euclid)

- The volume of a ball with radius r in dimension k scales as r^k. This means that if you want to faithfully represent distances in 2 dimensions then you are going to have to spread out, *a lot*.

The Crowding Problem

'High dimensional space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space.' (Euclid)

- The volume of a ball with radius r in dimension k scales as r^k. This means that if you want to faithfully represent distances in 2 dimensions then you are going to have to spread out, *a lot*.
- Since we used a Gaussian distribution for our $q_{i|j}$ probabilities, the probability of being neighbours decays square-exponentially.
- Spreading out (*a lot*) + square-exponential decay = very small $q_{i|j}$ for medium $p_{i|j}$, this makes the KL-divergence very sad.

The Crowding Problem

'High dimensional space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space.' (Euclid)

- The volume of a ball with radius r in dimension k scales as r^k. This means that if you want to faithfully represent distances in 2 dimensions then you are going to have to spread out, *a lot*.
- Since we used a Gaussian distribution for our $q_{i|j}$ probabilities, the probability of being neighbours decays square-exponentially.
- Spreading out (*a lot*) + square-exponential decay = very small $q_{i|j}$ for medium $p_{i|j}$, this makes the KL-divergence very sad.
- SNE's solution to this problem? Compress all the map points to a blob so everyone is friends with everyone (in particular nobody is not friends on the map who is friends in x-space.)

Fat Tails to the Rescue

An alternative to the group-hug solution is to use q distributions that are less needy, and more comfortable with long-distance relationships.

Fat Tails to the Rescue

An alternative to the group-hug solution is to use q distributions that are less needy, and more comfortable with long-distance relationships. We are talking of course about the Student t-distribution with one degree of freedom:

$$q_{i,j} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \tag{6}$$

Fat Tails to the Rescue

An alternative to the group-hug solution is to use q distributions that are less needy, and more comfortable with long-distance relationships. We are talking of course about the Student t-distribution with one degree of freedom:

$$q_{i,j} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \tag{6}$$

This behaves like an inverse-square law for larger distances, meaning that the probabilities are somewhat invariant to changes in scale for further apart map points.

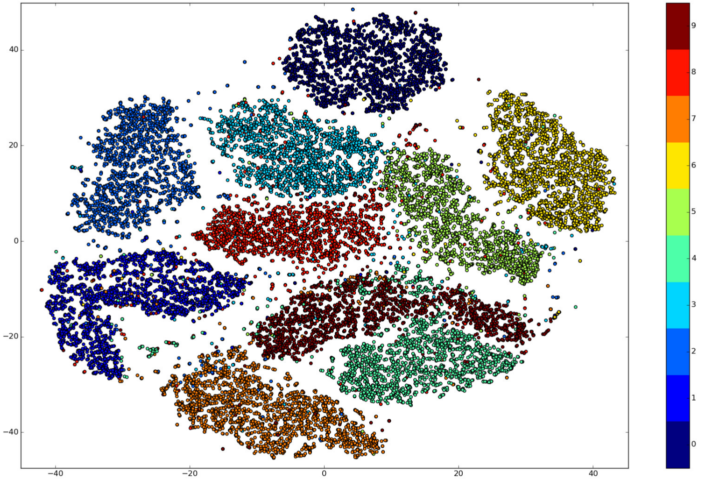
Fat Tails to the Rescue

An alternative to the group-hug solution is to use q distributions that are less needy, and more comfortable with long-distance relationships. We are talking of course about the Student t-distribution with one degree of freedom:

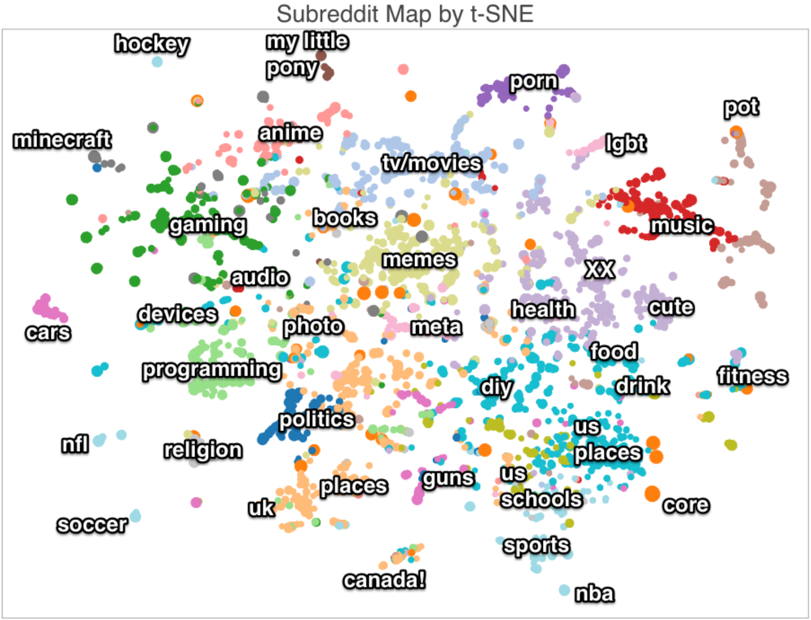
$$q_{i,j} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \tag{6}$$

This behaves like an inverse-square law for larger distances, meaning that the probabilities are somewhat invariant to changes in scale for further apart map points.
Also note that they use a symmetrised $q_{i,j}$ as opposed to $q_{i|j}$, this makes the computations simpler, but is not conceptually important.

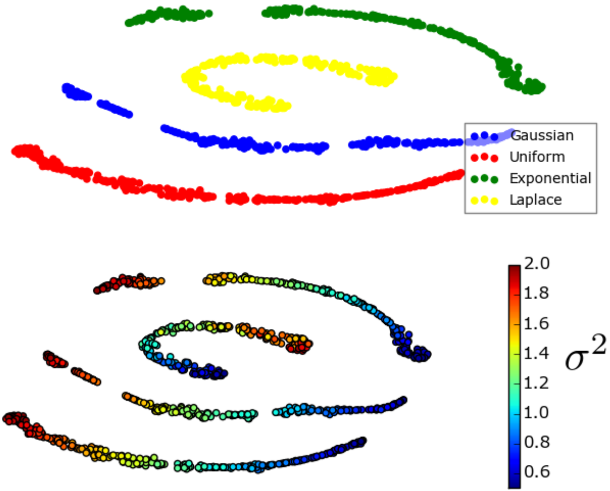
Pictures!!!: MNIST



Pictures!!!: Subreddits



Pictures!!!: Neural Statistician



Acknowledgements

DATASCIENCE
EPSRC CENTRE FOR DOCTORAL TRAINING

EPSRC
Engineering and Physical Sciences
Research Council

THE UNIVERSITY *of* EDINBURGH

THE UNIVERSITY *of* EDINBURGH
informatics