

# DimensionalityReduction

May 16, 2018

## 1 Solutions to Dimensionality Reduction

## 2 Dimensionality Reduction Task

- Use PCA from [MultivariateStats.jl](#), to reduce 100 dimensional word embedding down to 3,2 and 1 dimentionions.
- Plot these using [Plots.jl](#), coloring according to class

### 2.1 Tips:

- `plotly` is a good backend for 3D Plotting.
- The command `scatter(xs[1,:], xs[2,:], xs[3,:]; hover=all_words, zcolor=classes)`
- will plot a 3D scatter plot
- coloring each point according to the numerical array `classes`
- and putting a tooltip on each point, according to the string array `all_words`

## 3 First we loadup some data

For the the example presented here, we will use a subhset of Word Embedding, trained using [Word2Vec.jl](#). These are 100 dimentional vectors, which encode syntactic and semantic information about words.

You can download the dataseted from [here](#), and load it up with [JLD](#) as shown below. (or just load it directly if you have cloned the notebooks)

Example code for the loading, together with the words sorted into their original classes is below.

```
In [ ]: using JLD
countries = ["afghanistan", "algeria", "angola", "arabia", "argentina", "austral
usa_cities = ["albuquerque", "atlanta", "austin", "baltimore", "boston", "charlo
world_capitals = ["accra", "algiers", "amman", "ankara", "antananarivo", "athens
animals = ["alpaca", "camel", "cattle", "dog", "dove", "duck", "ferret", "goldfish
sports = ["archery", "badminton", "basketball", "boxing", "cycling", "diving", "e

words_by_class = [countries, usa_cities, world_capitals, animals, sports]
```

```

all_words = vcat(words_by_class...)
classes = vcat(((1:5) .* ones.(length.(words_by_class)))...);
embeddings = load("../assets/ClusteringAndDimensionalityReduction.jld", "em

```

## 4 Extension: T-SNE

- Use [TSne.jl](#), to perform similar dimensionality reduction, and to produce plots.

T-SNE is another popular DR method.

It is mostly maintained though. Be warned: it is sideways – it is row major, so transpose the inputs and outputs

You may have to play with the perplexity to get it to work well.

If you look at the resulting plots, you may note that countries are often paired up with their capital city.