Modelling a text corpus using deep Boltzmann machines in Python

Ricardo Pio Monti

PyData London, 2016

work with Giovanni Montana, Christoforos Anagnostopoulos,
Romy Lorenz & Rob Leech

Imperial College London

Who am I?

- PhD student within the Statistics Department, Imperial College London
- research focus is on computational statistics — motivated by the study of neuroimaging data
- I use python

This talk

- Restricted & Deep Boltzmann machines:
  - what are they?
  - advantages/disadvantages
  - training and model selection
- An application to text mining

Introduction

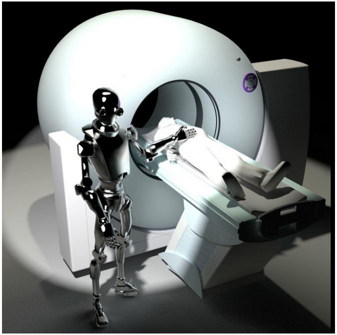
Motivation

- information retrieval/extraction, etc. . .

Motivation

- information retrieval/extraction, etc. . .
- The *Automatic Neuroscientist*

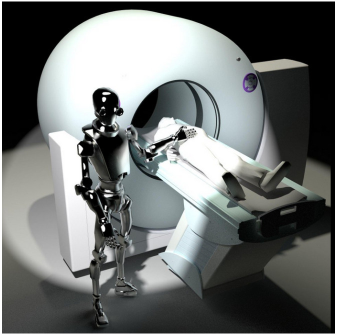
  [Lorenz, Monti, *et al.*, NeuroImage, 2016]

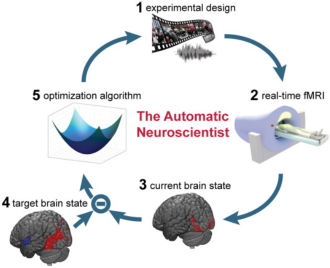


Motivation

- information retrieval/extraction, etc. . .
- The *Automatic Neuroscientist*

  [Lorenz, Monti, *et al.*, NeuroImage, 2016]





NeuroSynth

- Yarkoni *et al.*, **Nature Methods**, 2011
  - collected full text (& activations) for over 10k studies
  - dataset freely available: `neurosynth.org`

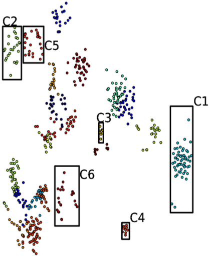
- example entry: `the ability to bind information together such as linking a name with a face or a car with a parking space is a vital process in human episodic memory to identify the neural bases for this binding process we measured brain activity during a verbal associative encoding task ...`

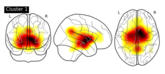
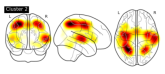
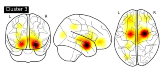
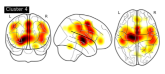
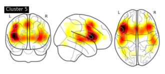
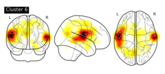
NeuroSynth

- Yarkoni *et al.*, **Nature Methods**, 2011
  - collected full text (& activations) for over 10k studies
  - dataset freely available: `neurosynth.org`
- **our goal**: model text corpus in an unsupervised manner
  - cluster documents
  - extract low-dimensional embeddings (*semantic representations*)

Taster

t-SNE document embeddings



	patients, brain, matter, atrophy, controls, cortical, subjects, motor, disease, regions, cognitive, mri, white, temporal, volume, imaging, clinical, compared, study, analysis, mci, areas, healthy
	neural, processing, activation, attention, cortex, regions, control, activity, brain, gyrus, cognitive, task, frontal, action, motor, emotional, functional, response, study, inferior, memory, sentence, speech
	allele, individuals, genotype, amygdala, brain, responses, activity, group, response, results, activation, healthy, carriers, cortex, subjects, gene, short, functional, significantly, association, adhd
	pain, patients, activation, cortex, functional, response, subjects, healthy, brain, central, compared, control, study, imaging, chronic, matched, processing, activity, controls, motor, inhibition
	task, frontal, activation, temporal, gyrus, inferior, cortex, brain, control, processing, network, performance, social, functional, language, activity, sentences, executive, behavioral, information, tasks
	motor, areas, activation, regions, study, recognition, temporal, connectivity, responses, brain, neural, object, processing, results, showed, gyrus, stimuli, stimulus, emotional, activity, naming

Restricted Boltzmann machines

Restricted Boltzmann machines

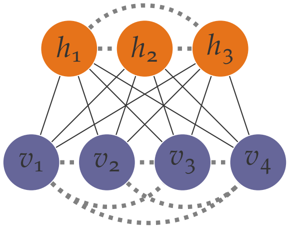
- advantages:
  - generative models: can sample/*dream* new data
  - fully unsupervised
  - stack together to get DBN/DBM $\Rightarrow$ obtain high-level representations of data
  - can be used to pretrain neural networks $\Rightarrow$ fine-tune using backprop

Restricted Boltzmann machines

- advantages:
  - generative models: can sample/*dream* new data
  - fully unsupervised
  - stack together to get DBN/DBM ⇒ obtain high-level representations of data
  - can be used to pretrain neural networks ⇒ fine-tune using backprop
- disadvantages:
  - training is not straight-forward
  - model selection/comparison is difficult

Restricted Boltzmann machines

- special case of **Boltzmann machines**:
  - undirected graphical model
  - very flexible, but difficult to train
- impose restrictions on graph structure



For now we assume binary hidden/visible units: $v \in \{0,1\}^D$ and $h \in \{0,1\}^F$

Restricted Boltzmann machines

- Energy based models: for given (v, h) the energy is:

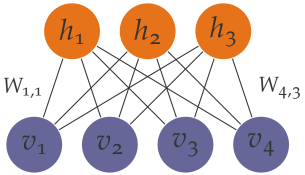
$$E(v,h) = -\sum_{i,j} W_{i,j} v_i h_j$$

- probability of (v, h) given by:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)}$$

- likelihood:

$$p(v) = \sum_h \frac{1}{Z} e^{-E(v,h)}$$



Consequences of bipartite structure

- bipartite graph structure leads to following desirable properties:
  - conditional independence:

$$p(v|h) = \prod_i p(v_i|h) \quad \text{and} \quad p(h|v) = \prod_j p(h_j|v)$$

Consequences of bipartite structure

- bipartite graph structure leads to following desirable properties:
    - **closed-form**, conditional distributions:

$$p(v_i = 1|h) = \sigma\left(\sum_j W_{i,j} h_j\right)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the **sigmoid** activation

Consequences of bipartite structure

- bipartite graph structure leads to following desirable properties:
  - **closed-form**, conditional distributions:

$$p(v_i = 1|h) = \sigma\left(\sum_j W_{i,j} h_j\right)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the **sigmoid** activation

```
## sample visible unit given hidden:
act = sigmoid(numpy.dot(W, hidden))
v_samp = (numpy.random.binomial(num_vis, .5) < act)*1
```

Training RBMs

Training

- tune parameters W via approximate maximum log-likelihood:

$$W_t = W_{t-1} - \alpha_{t-1} \frac{\partial \log p(v)}{\partial W}$$

Training

- tune parameters W via approximate maximum log-likelihood:

$$W_t = W_{t-1} - \alpha_{t-1} \frac{\partial \log p(v)}{\partial W}$$

- derivative is the sum of two expectations:

$$\left(\frac{\partial \log p(v)}{\partial W} \right)_{i,j} = \left\langle v_i h_j \right\rangle_{h|v} - \left\langle v_i h_j \right\rangle_{h,v}$$

Training

- easy to obtain unbiased estimate for first term:
  - sample $h_j|v$ and take $v_i h_j$ as unbiased estimate
- unbiased estimate of second term is trickier:
  - use Gibbs sampling
  - sample $v|h$ and $h|v$ several (k) times
  - often $k = 1$ used

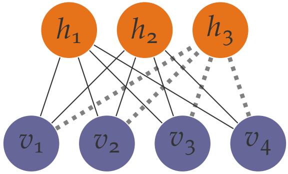
Training

```
## positive phase:
h_act = sigmoid(numpy.dot(visible, W))
h_samp = (numpy.random.binomial(num_hid, .5) < h_act)*1
posGrad = numpy.dot(visible, h_act)

## negative phase:
v_act = sigmoid(numpy.dot(W, h_samp))
h_act_fantasy = sigmoid(numpy.dot(v_act, W))
negGrad = numpy.dot(v_act, h_act_fantasy)

## update paramters:
W += delta * (posGrad - negGrad)
```

Avoiding overfitting: dropout

Overfitting is a concern for unsupervised methods:

- monitor free energy (unnormalized likelihood) of training and validation samples
- use dropout (remove hidden units with some fixed prob.)



Softmax RBMs for text modeling

Softmax RBMs

- modelling text as binary inputs is wasteful

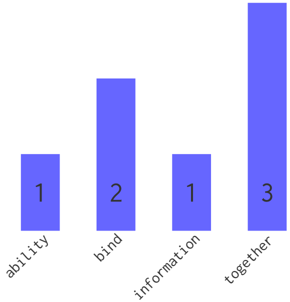
  e.g., `the ability to bind information...` represented as:

	1	2	3	4	5	⋯
`ability`		1				
`bind`				1		
`information`					1	
⋮						

- each document a **large, sparse** matrix.

Softmax RBMs

- a more parsimonious approach is to model word counts:



- each document now a vector.

Softmax RBMs

- model visible visible units as **multinomial** random variables

  each entry in v corresponds to # occurrences word:

  $$\hat{v}_{w_1} \quad \hat{v}_{w_2} \quad \hat{v}_{w_3} \quad \hat{v}_{w_4}$$

Softmax RBMs

- model visible visible units as **multinomial** random variables

  each entry in v corresponds to # occurrences word:

  $$\hat{v}_{w_1} \quad \hat{v}_{w_2} \quad \hat{v}_{w_3} \quad \hat{v}_{w_4}$$

- conditional distr. of visible units give by softmax:

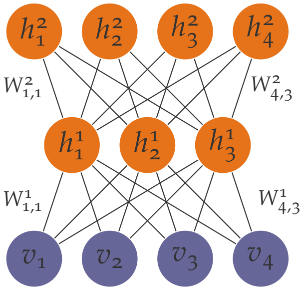
$$p(v_i|h) = \frac{e^{\sum_j W_{i,j} h_j}}{\sum_{j=1}^{D} e^{\sum_k W_{k,j} h_j}}$$

- all properties/training of RBMs unchanged

Deep Boltzmann machines

Deep Boltzmann machines

- RBM with bipartite layers of hidden units
- can learn high-level abstractions of data distribution



- energy function:

$$E(v, h^1, h^2) = -\sum_{i,j} W_{i,j}^1 v_i h_j - \sum_{i,j} W_{i,j}^2 h_i^1 h_j^2$$

Training DBMs

- need to estimate data dependent & indep. expectations
- **data-dependent**:
  - estimate $< vh^1 >_{h^1,h^2|v}$, $< h^1h^2 >_{h^1,h^2|v}$ using mean-field variation inference
  - iterate to convergence:

$$\mu_i^1 = \sigma\left(\sum_j W_{i,j}^1 v_j + \sum_j W_{i,j}^2 h_j^2\right)$$

$$\mu_i^2 = \sigma\left(\sum_j W_{i,j}^2 h_j^1\right)$$

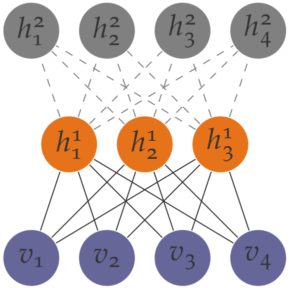
for all j

Training DBMs

- need to estimate data dependent & indep. expectations
- **data-independent**:
  - estimate $< vh^1 >_{h^1,h^2,v}$, $< h^1h^2 >_{h^1,h^2,v}$ using persistent Markov chains
  - store M fantasy particles and update k times at each iteration

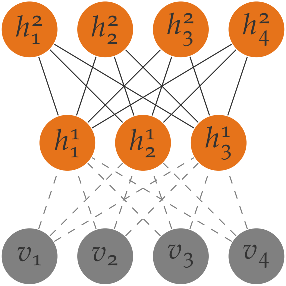
Pre-training DBMs

- iteratively stack RBMs
- sample hidden units and use as training data



Pre-training DBMs

- iteratively stack RBMs
- sample hidden units and use as training data



Results

DBM model

- training:
  - two hidden layers — 50 binary units each[1]
  - pretraining using CD-1 and dropout ($p = 0.10$)
- dataset:
  - abstracts for $\approx 10k$ scientific publications — mean document length of 80 words
  - vocabulary of 1000 words

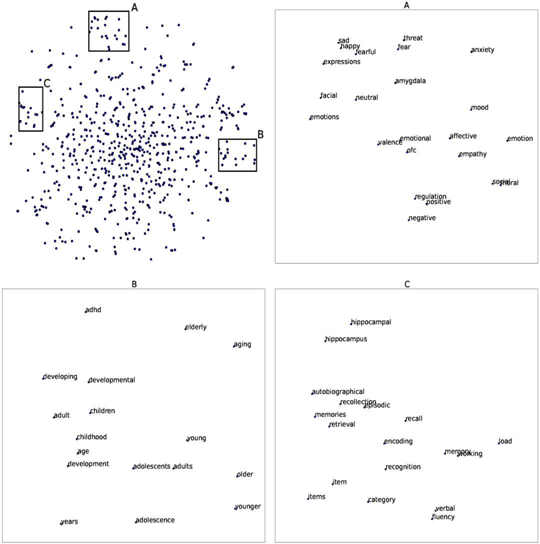
[1]selected by maximizing log-likelihood over validation set

word clustering

- can represent each word as a 50-dim vector

Associated vocabulary
memory, retrieval, encoding, hippocampus, hippocampal, episodic, items, recall, memories, recollection, item, familiarity, autobiographical
language, semantic, words, speech, word, reading, verbal, phonological, lexical, linguistic, naming, fluency, verbs, english
adults, age, children, years, older, young, development, adolescents, developmental, aging, sleep, adult, late, younger, blind, childhood, hearing, adolescence
emotional, amygdala, social, negative, faces, face, emotion, neutral, affective, facial, anxiety, fear, expressions, regulation, emotions, ofc, valence, personality, arousal, fearful, trait, threat, sad, happy, mood, empathy, moral, person, traits, communication
patients, controls, schizophrenia, disorder, deficits, disease, abnormalities, symptoms, impaired, impairment, adhd, alterations, dysfunction, mdd, abnormal, atrophy, patient, ptsd, severity, mci, damage, bipolar, lesions, impairments, deficit, depressive, ocd, mild, syndrome, symptom, elderly, dementia, epilepsy, poor, pathophysiology

word embeddings



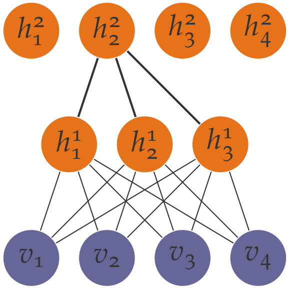
one-step reconstruction

- estimate hidden units and then reconstruct words:

Input	One-step reconstruction
memory	memory, working, recall, performance, retrieval, verbal, load, semantic, recognition, task
emotion	social, emotion, emotional, regions, ofc, brain, affective, gray, traits, amygdala
face	social, facial, faces, face, emotional, processing, regions, functional, brain, cortex
disorder	patients, mdd, disorder, adhd, abnormalities, controls, brain, matter, alterations, structural
mode	network, default, connectivity, brain, regions, cognitive, functional, mode, activity, cortex

hidden layers are discriminative

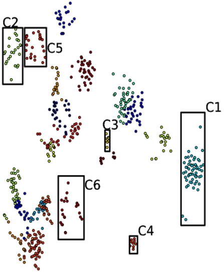
- can obtain word distribution conditional on hidden unit activations

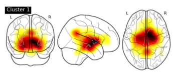
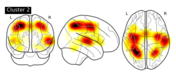
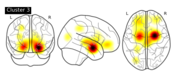
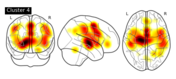
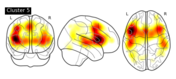
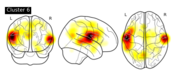


hidden layers are discriminative

- unit 4: younger, older, aging, adults, gender
- unit 27: encoding, hippocampal, hippocampus, retrieval, recollection
- unit 38: reading, words, word, phonological, language, speech
- unit 42: gray, matter, volume, white, thickness

document embedding



	patients, brain, matter, atrophy, controls, cortical, subjects, motor, disease, regions, cognitive, mri, white, temporal, volume, imaging, clinical, compared, study, analysis, mci, areas, healthy
	neural, processing, activation, attention, cortex, regions, control, activity, brain, gyrus, cognitive, task, frontal, action, motor, emotional, functional, response, study, inferior, memory, sentence, speech
	allele, individuals, genotype, amygdala, brain, responses, activity, group, response, results, activation, healthy, carriers, cortex, subjects, gene, short, functional, significantly, association, adhd
	pain, patients, activation, cortex, functional, response, subjects, healthy, brain, central, compared, control, study, imaging, chronic, matched, processing, activity, controls, motor, inhibition
	task, frontal, activation, temporal, gyrus, inferior, cortex, brain, control, processing, network, performance, social, functional, language, activity, sentences, executive, behavioral, information, tasks
	motor, areas, activation, regions, study, recognition, temporal, connectivity, responses, brain, neural, object, processing, results, showed, gyrus, stimuli, stimulus, emotional, activity, naming

Thank you

- any questions?
- full code at `github.com/piomonti`
- contact: `rpm08@ic.ac.uk`