



SENIOR THESIS IN MATHEMATICS

---

# Data Representation as Low Rank Matrix Factorization

---

*Author:*

Ziv Epstein

ziv.epstein@pomona.edu

*Advisor:*

Dr. Blake Hunter

Submitted to Pomona College in Partial Fulfillment  
of the Degree of Bachelor of Arts

January 28, 2017

# Contents

|          |                                                             |           |
|----------|-------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                         | <b>2</b>  |
| <b>2</b> | <b>Theories for Human Image Understanding</b>               | <b>4</b>  |
| 2.1      | Part-based recognition and visual perception . . . . .      | 4         |
| 2.2      | Knowledge representation as information retrieval . . . . . | 6         |
| 2.3      | The hierarchical structure of perception . . . . .          | 7         |
| 2.4      | Interpretations for Algorithmic Formulation . . . . .       | 8         |
| <b>3</b> | <b>Techniques</b>                                           | <b>9</b>  |
| 3.1      | Singular Value Decomposition . . . . .                      | 9         |
| 3.2      | Neural Network Representation . . . . .                     | 11        |
| 3.3      | Matrix factorization . . . . .                              | 12        |
| <b>4</b> | <b>Types of Data</b>                                        | <b>15</b> |
| 4.1      | Image . . . . .                                             | 15        |
| 4.2      | Sound . . . . .                                             | 16        |
| 4.3      | Graph . . . . .                                             | 16        |
| 4.4      | Natural Language . . . . .                                  | 17        |
| <b>5</b> | <b>Semi-Supervised NNMF</b>                                 | <b>19</b> |
| 5.1      | Approach . . . . .                                          | 19        |
| 5.2      | Visualization . . . . .                                     | 20        |
| <b>6</b> | <b>Hierarchical Representation</b>                          | <b>21</b> |
| 6.1      | Approach . . . . .                                          | 21        |
| 6.2      | Visualization . . . . .                                     | 23        |
| <b>7</b> | <b>Deep Models</b>                                          | <b>25</b> |
| 7.1      | Approach . . . . .                                          | 25        |
| 7.2      | Deep Neural Models . . . . .                                | 25        |
| 7.3      | Visualization . . . . .                                     | 25        |
| <b>8</b> | <b>Discussion</b>                                           | <b>26</b> |

## **Abstract**

A growing body of literature suggests that visual perception and representation is the result of a hierarchical relation network, whereby nodes that contain values on global properties are scaffolded with increasing levels of representational complexity. Here, we explore algorithms that represent data as low-dimensional approximations.

# Chapter 1

## Introduction

There is a growing interest in the computational and mathematical science in the development of algorithms that can automatically create understanding about data corresponding to some phenomenon. As an explosion of data collection and distribution systems have become the mainstream, there is now more data available than humans observers to understand it. Thus the need for automated algorithms that can infer patterns and relationships within data is self evident.

Here, I will consider the case where data is represented as a well-structured and real-valued matrix. This approach can represent data from a large range of domains, such as sound, natural language, visual images and networks. Thus, many of the algorithms and heuristics I explore here are domain general, and can be applied with success to a range of settings.

My particular focus in this thesis is the notion of *low-dimensional representations of data*. A real valued matrix  $X$  that represents a collection of sounds, natural language documents, images or networks taken from real world settings will necessarily be noisy and not optimally structured. One canonical challenge (that we will dub “The Representation Problem”) is to find a  $X' \approx X$  that well approximates the original data’s structure, but with  $\text{rank}(X') \ll \text{rank}(X)$ . A low rank approximation of the data gives a sparse representation that in some sense captures the most important facets of the data.

This approach also has explanatory power. Let  $r = \text{rank}(X')$  and  $b = (b_1, b_2, \dots, b_r)$  be a basis for  $X'$ . Depending on the constraints imposed on the  $b_i$ ’s, they can have many different interpretations. For example, when the  $b_i$ ’s are constrained to be orthonormal to each other, the resulting algorithm is Principal Component Analysis (PCA), which “supplies the user with a lower-dimensional projection of the object when viewed from its most informative viewpoint.”

Since our mission is to emulate human understanding, a natural starting place for inspiration is the human brain. Indeed there is a long-standing tradition in computer and information science to algorithmically model the information processing systems the brain uses to make sense of the world. However, I hope to also go one step deeper. In addition to an overview of many of the important algorithms used today, we will explore the cognitive implications of these algorithms. In particular, I aim to invert the paradigm of using neural architecture to inform algorithms: I will use algorithms to inform notions of neural architecture. By creating a bilateral dialogue between the cognitive and computer sciences, I hope this work will help to bridge the gaps between the two fields by 1) equipping the

cognitive ideas of neural information processing with a notational formalism consist with the algorithms research and II) contextualizing the algorithms research whereby it is clear how it might relate to the acquisition of human knowledge.

## Chapter 2

# Theories for Human Image Understanding

The general class of algorithms I am considering have in the common the fact that they take as input a large amount of noisy, somewhat structured data, and output some unified high-level notion of underlying pattern. Thus in many ways they emulate the most successful



Figure 2.1: How does your brain take the raw retinal activations of this picture and produce a mental representation of a cup?

raw data to high-concept pipeline: the human vision system. When starting at a cup, our brains take in thousands of raw retinal activities, and through a complex schema of neural information processing, result in a high-level conceptualization of a cup. The intricacies of this pipeline are not fully understood, but many of the driving mechanisms are. To motivate the algorithmic approach, I begin with a short overview of the pervading theories for human image understanding.

### 2.1 Part-based recognition and visual perception

A given object can project an uncountable infinitude of possible visual configurations to the human retina. Thus the phenomenon of how raw visual input to the visual system is represented as robust concepts in the human mind has long haunted philosophers, cognitive

scientists, mathematicians and computer scientists. Empirical observation imposes several axioms that any respectable theory of object recognition must take into account, which Biederman enumerates:

1. **Imprecise attention to detail:** “Access to the mental representation of an object should not be dependent on the absolute judgments of quantitative detail. For example, distinguishing among just several levels of the degree of curvature or length of an object typically requires more time than that required for the identification of the object itself.”
2. **Topological fuzziness:** “The information that is the basis for recognition should be relatively invariant with respect to orientation and modest degradation’.”
3. **Plasticity:** “Partial matches should be computable. A theory of object interpretation should have some principled means for computing a match for occluded, partial or new exemplars of a given category.”

These axioms are central to task of modeling, and will be used by the cognitive and mathematical models put forward here. A large number of explanatory theories have been proposed over the years for this phenomenon of object recognition. One extremal idea is the structuralist position which posits that “the perception of whole figures is nothing more than the concatenation of primitive perceptual elements.” [18]. Another extremal theory, born out of the Gestalt theory, claims that the recognition of complex objects is an indivisible process, which cannot be reconstructed by considering the nature and properties of its constituent parts. The most performant modern theories lie within the simplex defined by these holistic and atomic extrema. One such theory, the Recognition-by-Components (RBC) theory proposed by Biederman proposes a sparse vocabulary of atoms and a schema for how these components can be conglomerated to represent in memory (see Figure 2.1) [2]. First, edges are extracted from the raw sensory data received from the optical system. Local properties of the input, such as luminance, texture and color, aid in providing robust edge segmentations. Then, in parallel, the parsing of the concave regions and the identification of nonaccidental properties is performed. The latter provides “critical constraints on the identity of the components. Then these processes coalesce to form a specific arrangement of components, which are then matched against pre-existing representations in memory. The representation that best “matches” the observed arrangement gives rise to the actual object identification.

Since RBC uniquely relates to the visual system, the components Biederman proposes are a set of generalized cones called **geons** (for “geometric ions”) [2]. A victory for RBC is its explanatory vastness with a small number of components. Biederman suggests that 36 geons is sufficient to represent all objects that humans can differentiate between.

RBC has two heuristics that we will primary draw upon for formalizing this theory of object recognition with formal models. The first is this ability to represent the nuance of human perception with a small number of components. In mathematical terms, these components form a linearly independent **basis**, by which any object,  $O$ , can be approximated:

$$O = \sum_{i=1}^d \alpha_i b_i \tag{2.1}$$

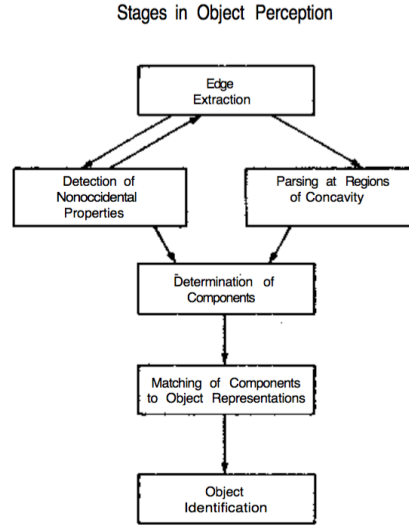


Figure 2.2: Processing stages for object recognition proposed by the RBC Theory. Adapted from [2]

where  $d$  is much smaller than the space of possible objects. Natural questions that arise from this model of human perception are 1) What is sufficient for  $d$ ? Biederman proposes  $d = 36$  is sufficient of visual object recognition, but what about other domains? and 2) How are these bases,  $b_i$  learned? Biederman suggests that  $b_i$  is a volumetric cone, but gives no explanation for how this components emerge from the human visual system.

The second heuristic of RBC we will use the schematic for recognition (see Figure 2.1). This schema gives us a roadmap for a suite of algorithms we will use to generate representational models (see Section 3).

## 2.2 Knowledge representation as information retrieval

The RBC Theory puts forward a highly structured representation of a given input  $O$  as described in Equation 2.1. It then suggests that this representation is then procedural matched with a preexisting notion in memory towards recognition. To understand how this matching occurs, we turn to Minsky’s framework for representing knowledge [16]. Minsky thinks of memory as a collection of **frames**, which is a data structure that represents a stereotyped situation, space or object [16]. In particular, a frame is a hierarchical network of nodes and relations which correspond to that which the frame represents. In particular, the “top levels” of the network are fixed. They represent entities and relations that are always true about the frame’s content. The lower lever has **terminals**, empty “slots” to be filled with raw input or data. These terminals carry with them a set of rules (which are frames themselves) which specify the necessary conditions for assignment that terminal.

Large collections of these frames are aggregated into structures called **frame-systems**, where related frames are proximal. In particular, actions in the world correspond to **transformations** in the frame-system, whereby similar objects have a smooth transformation between them. For example, different frames might correspond to observing a room at dif-



ferent angles, and the transformation between adjacent frames corresponds to the observer moving to change their vantage point [16]. These frame-systems are further connected into a **information retrieval network**.

After the raw sensory input of a stimulus percolates through the RBC processing system outlined in Figure 2, it finally reaches a component representation that must be matched with a frame. This process is done as follows. For a given frame  $F^i$  with top levels  $f^i$  and terminals  $t^i$ , the features of the representation are assigned to the frame's terminals,  $t_i$ . In the assignment sufficiently conforms for the specifications of the terminals (i.e. their corresponding frames  $F_{t_i}$ , and the top level nodes  $f_i$ ), the object is identified as in the last stage of Figure 2.1. However if the proposed frame cannot conform to the terminal specifications, then the information retrieval system repeatedly puts forward another frame until a candidate fits the condition.

This process of selecting an optimal frame for representation is parallel to the optimization component of the algorithms we will discuss.

## 2.3 The hierarchical structure of perception

Biederman puts forward an atomic unit of visual perception [2], and Minsky extends that notion with a generalized framework for representing this atomic units [16]. However neither of these formulations describe explicitly how raw sensory data can be accumulated into something as sophisticated as the high-level recognition of an object. Palmer puts forward a structural schematic for hierarchical perceptual representation involving structural units, values on global properties and structural relationships (see Figure 2.3) [18]. The values of

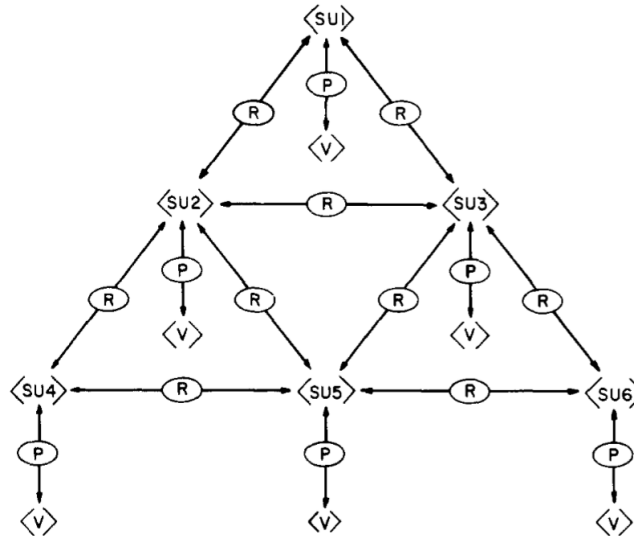


Figure 2.3: The hierarchical relationship network used to represent perceptual information. At each level of the hierarchy, a structural unit (SU) is a combination of values (V) on global properties (P) and structural relationships to other structural units. Adapted from [18]

the global properties at each node of the network are a real-valued vector “along perceptual

dimensions relative to some referent.” [18]. During the process of pattern recognition, these numeric values are used to evaluate the goodness-of-fit between some raw sensory input (that is to be identified) and some pre-existing information encoded in the network. The task of matching the sensory input to the encoded information is analogous to Minsk’s frame retrieval discussed above [16].

## 2.4 Interpretations for Algorithmic Formulation

The theories put forward above have several key formulations of the Representation Problem that I will leverage and formalize throughout this thesis.

### **Object Classification as Supervised Manifold Learning**

As we have seen, the hierarchies proposed above aim to match high-order labels to a series of raw retinal activations. For example, let  $(r_1, r_2, \dots, r_n)$  be the retinal activation of observing Figure 2. Previous to observing the image, there exists somewhere in the perceptual hierarchy of your brain the label of cup  $\ell_i$ . Thus the task of object classification can be formulated as the supervised learning of a manifold  $f$ , whereby  $f(r_1, r_2, \dots, r_n) = \ell_i$ .

### **Geons as Topics**

### **Hierarchical Organization of Topics**

finish  
these and  
make fig-  
ures

# Chapter 3

## Techniques

### 3.1 Singular Value Decomposition

The first technique to represent a matrix  $X$  is to factorize it using singular value decomposition.

**Theorem 1.** Suppose  $X \in \mathbb{R}^{m \times n}$ , then there exists a factorization, called the singular value decomposition of  $X$ , of the form

$$X = U\Sigma V^T$$

where  $U \in \mathbb{R}^{m \times m}$  is unitary,  $\Sigma \in \mathbb{R}^{m \times n}$  with non-negative real numbers on the diagonal and  $V^T \in \mathbb{R}^{n \times n}$  is unitary.

The diagonal entries  $\sigma_i$  of  $\Sigma$  are called the **singular values** of  $X$ .

*Proof.* (Bast) Observe that  $X^T X \in \mathbb{R}^{n \times n}$  is symmetric. Thus, there exist  $n$  eigenvectors  $v_1, \dots, v_r$  with corresponding eigenvalues  $\lambda_1, \dots, \lambda_r$  that are pairwise orthogonal (i.e.  $v_i^T v_j = 0 \ \forall i \neq j$ ) with  $r = \text{rank}(X^T X) = \text{rank}(X)$ . Furthermore, we have that  $X^T X = V D V^T$ , where  $V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$  and

$$d = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

Since  $v_i$  is an eigenvector for  $X^T X$ , observe that  $X X^T X v_i = X \lambda_i v_i = \lambda_i X v_i$ . Which is to say that if  $v_i$  is an eigenvector for  $X^T X$ , then  $X v_i$  is an eigenvector for  $X X^T$  with the same eigenvalue  $\lambda_i$ . Now we can compute the squared norm of  $X v_i$  as follows

$$\|X v_i\|^2 = (X v_i)^T (X v_i) = v_i^T X^T X v_i = v_i^T \lambda_i v_i = \lambda_i v_i^T v_i$$

So the norm of  $X v_i = \sqrt{\lambda_i} = \sigma_i$ . Now let  $u_i = X v_i / \sigma_i$ . Then we have

$$u_i^T X v_j = (X v_i / \sigma_i)^T X v_j = \frac{v_i^T X^T X v_j}{\sigma_i} = v_i^T v_j \times \frac{\lambda_i}{\sigma_i}$$

which is to say  $u_i^T X v_j = 0$  if  $i \neq j$  and  $u_i^T X v_j = \sigma_i$  if  $i = j$ . Now writing these equations for  $i = 1, \dots, r$  yields

$$\begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_r^T \end{bmatrix} \cdot X \cdot [v_1 \cdots v_r] = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}$$

Since  $u_i \in \mathbb{R}^m$ , the lefthand matrix we have is a  $r \times m$  matrix. Thus we generate  $u_i$  for  $i = r+1, \dots, m$  to expand the matrix to be  $m \times m$  by arbitrarily picking unit vectors that are pairwise orthogonal and orthogonal to  $u_1, \dots, u_r$ . Similarly for the  $v_i$ 's, we generate  $v_i$  for  $i = r+1, \dots, n$  to expand the matrix to be  $n \times n$  by arbitrarily picking unit vectors that are pairwise orthogonal and orthogonal to  $v_1, \dots, v_r$ . This extension gives us in matrix form

$$\begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_m^T \end{bmatrix} \cdot X \cdot [v_1 \cdots v_n] = \Sigma$$

Now let  $U = [u_1 u_2 \cdots u_m]$  and  $V = [v_1 v_2 \cdots v_n]$  to write the above equation in a more clear way:  $U^T X V = \Sigma$ . Since  $U$  and  $V$  are unitary, we have  $V^T = V^{-1}$  and  $U^T = U^{-1}$ . Thus we can solve the equation for  $X$ :

$$X = U \Sigma V^T$$

□

An extension of the SVD can be used to create a low-rank approximation for the data. For a low rank  $r < \min(m, n)$ , we can find an optimal approximation using the following theorem.

**Theorem 2** (Eckart-Young). Suppose  $X \in \mathbb{R}^{m \times n}$  has a singular value decomposition of  $X = U \Sigma V^*$  with

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

where  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$  are the singular values of  $A$  and where  $U \in \mathbb{R}^{m \times m}$  and  $V^* \in \mathbb{R}^{n \times n}$  are unitary. Then the matrix  $X_r = U \Sigma_r V^*$ , where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & 0 \\ 0 & 0 & \cdots & \sigma_r & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

is a global minimizer of the problem

$$\min_{\hat{X}, \text{rank}(\hat{X}) \leq r} \frac{1}{2} \|X - \hat{X}\|_f^2$$

and its error is

$$\frac{1}{2} \|X - \hat{X}\|_f^2 = \frac{1}{2} \sum_{i=r+1}^n \sigma_i^2$$

Optimization can be algorithmically challenging, and is often solved using local techniques, such as gradient descent. The fact that the Eckart-Young low-rank approximation guarantees a global minimum is a very nice result.

## 3.2 Neural Network Representation

A data matrix  $X \in \mathbb{R}^{m \times n}$  can also be considered as vectorized input to a neural network. Since there has been a growing interest in the representations that neural networks produce, we will explore some examples here. A **neural network** is a network of **neurons**, which

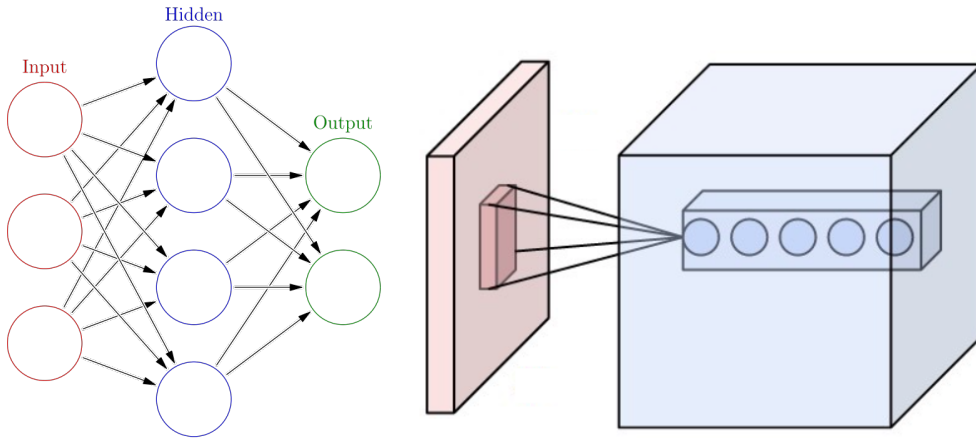


Figure 3.1: Left: a very simple neural network with 3-dimensional input (red), one hidden layer with 4 hidden units (blue), and 2-dimensional output (green). Right: A pictorial representation of a convolutional layer of a neural network.

represent units of computation organized in **layers**. (see Figure 2). For example, say we have input 3-dimensional input  $\vec{x} = [1, 2, 3]$ . Each of those values would be passed to the input neurons in Figure 1. Then, the first set of black arrows moves the input layer to the hidden layers by multiplying them by some weights and summing them together. This is equivalent to multiplying the  $3 \times 1$  vector  $\vec{x}$  by a  $5 \times 3$  matrix of weights,  $W$ . Then a nonlinear function  $f$  is applied elementwise to the 5 dimensional vector  $W\vec{x}$ . This nonlinear function increases the complexity of the space the neural network can explore, and is usually  $f(x) = \tanh(x)$ . This process is repeated with the output layer to achieve a  $2 \times 1$  output vector. In total, the computational process of the neural network can be captured by the equation

$$\vec{output} = f(W^2(f(W^1(\vec{input}))))$$

where  $\vec{input}$  is the  $3 \times 1$  input data,  $W^1$  is the  $5 \times 3$  weight matrix for the first layer,  $W^2$  is the  $2 \times 5$  weight matrix for the second layer,  $f$  is an elementwise nonlinear function, and  $\vec{output}$  is the  $2 \times 1$  output vector. The **activation of a layer**,  $F^\ell$  is the intermediary result of running a given input through the network. For example, the activation of the first layer  $F^1$  is

$$F^1 = f(W^1(\vec{input}))$$

A **convolutional neural network** is a neural network is at least one convolutional layer. This neural architecture is inspired by the organization of the human visual cortex, which small regions of neurons are aggregated into a single value. A convolutional layer is a layer that takes a small window of the input data and computes a single value for that region. For example, a **max pooling** convolutional layer simply computes the maximum value found for a given window (see Figure 2).

Through many layers and complex nonlinearities, a neural network with only a single hidden layer can approximate any function [8] with sufficient weights. But how are these weight matrices  $W^i$  constructed? This requires **training** the neural network to configure these weight matrices. In general, this is achieved by feeding it labeled data,  $(x, y)$  (with  $x_i \in \mathbb{R}^3$  and  $y_i \in \mathbb{R}^2$  for our example) and initializing the weights as random. Then, for each observation  $x_i$ , a prediction  $\hat{y}$  is computed by running  $x_i$  through the network:

$$\hat{y} = f(W^2(f(W^1(x_i))))$$

Then the error in the network can be computed as the difference between the prediction and the actual output:

$$E(x_i) = (\hat{y} - y)^2$$

The weights can then be updated by descending the gradient of the error using **backpropagation**. A review of the backpropagation algorithm is beyond the scope of this paper, but for a full overview, see [19].

For the purposes of image processing, the input data is raw images  $x \in \mathbb{R}^{1000 \times 1000 \times 3}$  assuming they are 1000 pixels by 1000 pixels and are colored. The output  $y$  used to trained the network is often labels corresponding to what the image is of, such as:

$$y \in \{\text{elephant, doughnut, } \dots, \text{child, apple}\}$$

A well documented phenomenon of applying DCNNs to image processing is that lower level layers correspond to pixel or edge level information, and as you look at higher level layers, you get more high order structure. As information passes from the input image pixels to the output classifier, the “scale” of that projection increases. For example, layers towards the input correspond to local pixelated regions, edges and neighboring contours. Layers towards the output correspond to global structure, groups of aggregated edges, etc. The work presented in this paper takes advantage of this important fact. We will refer to this phenomenon as the visual hierarchy phenomenon (VHP).

### 3.3 Matrix factorization

Another famous approach for finding the low-rank approximation of a matrix  $X$  is matrix factorization, which aims to find an  $A$  and  $S$  such that  $X \approx AS$  as shown in Figure 3.3.

Typically, for a input matrix  $X \in \mathbb{R}^{n \times m}$  the size of the inner dimension,  $k$ , is specified ahead of time (such that  $A \in \mathbb{R}^{n \times k}$  and  $S \in \mathbb{R}^{k \times m}$ ). Since  $k$  is chosen such that  $k < \min(m, n)$  we have the was  $\text{rank}(AS) = k$ , which gives us our low-rank approximation of  $S$ .

As discussed further in Section 4.1, the constraints imposed upon  $A$  and  $S$  result in different interpretations of the result. For example, constraining the columns of  $S$  to be a unary vectors, where one element is equal to one and the other elements are zero, results in the Vector Quantization. Or constraining the columns of  $A$  to be orthonormal and the rows of  $S$  to be orthogonal to each results in Principal Component Analysis.

In this thesis, I will focus on a third class of constraints, whereby  $A$  and  $S$  must be non-negative matrices. This constraint is shown to yield a parts-based approach, which is consistent with a “topic model” notion [13].” What results is a family of algorithms dubbed *non-negative factorization* or NNMF.

NNMF was first employed by Paatero and Tapper [17] but was made popular by Lee and Seung [13].

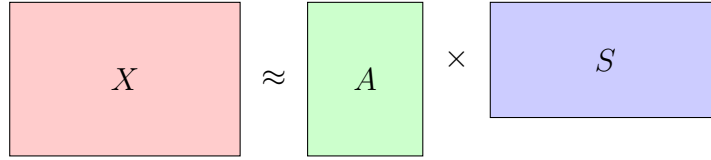


Figure 3.2: A visual representation of the non-negative matrix factorization

The problem of finding such a factorization can be formulated as finding a non-negative  $A$  and  $S$  that minimize the error

$$F = ||X - AS||^2 \quad (3.1)$$

While this optimization problem is not convex in both  $A$  and  $S$ , it is convex in one of them. So for a given, fixed  $S$ , we can find the optimal  $A$  by setting the gradient equal to zero. Since  $||X - AS||^2 = \langle X - AS, X - AS \rangle = X^T X - 2X^T AS + (AS)^T(AS)$  we have

$$\frac{\partial F}{\partial A}(X^T X - 2X^T AS + (AS)^T(AS)) = 0$$

$$\text{implies } S^T AS = 2X^T S$$

which is to say  $\frac{X^T S}{S^T AS} = 1$  at the optimal  $A$ . This equality gives us the below multiplicative

update algorithm.

**Input:**  $k=0$ ; Initialize  $A^0, S^0$

**repeat**

$$\begin{aligned} A^{k+1} &= A^k \circ \frac{XS^k}{A^k(S^k)^T A^k} \\ S^{k+1} &= S^k \circ \frac{XA^{k+1}}{S^k(A^{k+1})^T A^{k+1}} \\ k &= k + 1 \end{aligned}$$

**until** *Stopping condition*;

**Algorithm 1:** Multiplicative Update

This optimization scheme naturally leads to a convex optimization function, so the above algorithm can simply be iteratively applied (until for given  $T$  we have  $k > T$  or for a given  $\epsilon > 0$  we have  $\|X - AS\|^2 \leq \epsilon$ ).

**Theorem 3.** The Euclidean distance  $\|X - AS\|^2$  is non-increasing under the updating rules of Algorithm 1.

A key benefit of non-negative matrix factorization is that its results are very interpretable. For a given factorization  $X \approx AS$ , we can think of the  $k$  rows of  $S$  as a basis  $(b_1, b_2, \dots, b_k)$  for the rows of  $X$ . Under this interpretation, the row of  $A$  give us the linear combination used to form the corresponding row in  $X$ :

$$X_i = \sum_{j=1}^k A_{i,j} \times S_j,$$

This interpretation is directly applicable to the notion of “geons as topics” put forward in Section 2.4. In this thesis, we will focus on NNMF as the central algorithm and we will explore how NNMF can be used to build a theory of visual perception.



# Chapter 4

## Types of Data

We have defined several algorithmic approaches for the automated understanding of any real valued matrices. In this chapter, we explore how important types of media and data can be represented as real valued matrices. Each section will focus on NNMF as the lens for analysis, and will describe how NNMF can be interpreted in each domain.

### 4.1 Image

Because visual object recognition is the foundation of many of the cognitive theories discussed here, it makes sense to think about representing images as data. Because of how computers and digital cameras represent images, they are perhaps the most well suited for representation as data. For gray scale, an image is just a  $w \times h$  matrix, where  $w$  is the width of the image and  $h$  is the height of the matrix. For color, an image is a triple  $\{R, B, G\}$  where  $R, B, G \in \mathbb{R}^{w \times h}$  correspond to red, blue and green intensities in the image, respectively. The canonical demonstration of NNMF’s ability to learn a parts-based representation utilizes a dataset of faces,  $X$ , with a row corresponding to a given image and the columns corresponding to a flattened vector representation of the matrix. Lee and Seung compare the NNMF factorization of this corpus with vector quantification (VQ) and principal component analysis (PCA). Indeed all three methods attempt to factorize  $X$  into two matrices  $A$  and  $S$ : the only difference is the constraints imposed on  $A$  and  $S$ . As we have seen, NNMF does not allow nonnegative entries in either  $A$  and  $S$ . For VQ, each column of  $S$  is constrained to be a unary vector. And finally PCA constrains the columns of  $A$  to be orthonormal and rows of  $S$  to be orthogonal to each other [13].

As can be seen in Figure 4.1, the bases each approach learns is radically different. VQ learns a basis of prototypes, where each basis itself is an entire face, and each row of  $X$  is a linear combination (i.e. addition and subtraction) of these protofaces. Alternatively, PCA learns a basis of ‘eigenfaces,’ which correspond to “distorted” versions of complete faces [21]. Instead of learning a “holistic” representation of the faces like VQ and PCA, NMF learns a parts-based model, where the rows of  $S$  correspond to discrete components of a face.

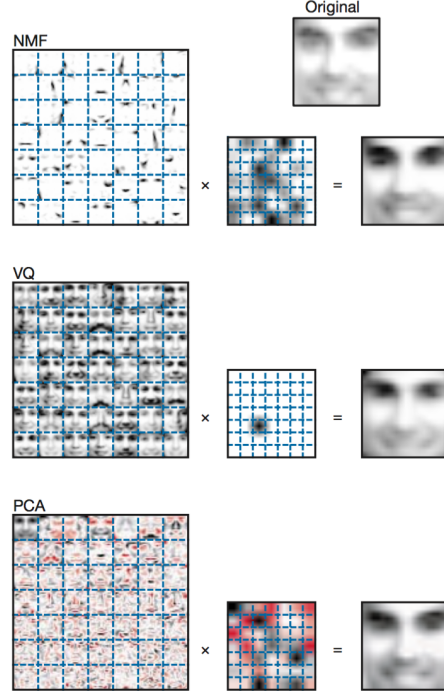


Figure 4.1: A comparison of the basis vectors learned for NMF, VQ and PCA. Adapted from [13]

## 4.2 Sound

Despite not being as discrete as images or graphs, sound also can be represented in a high structured way. There have been many approaches to do so. Perhaps the most success is the Deep Speech models, which feed speech snippets into a RNN paired with a language model for real time dictation [1, 6]. These Deep Speech systems represent sound as follows. An utterance  $x$  is a time series of length  $T$  sliced into time intervals  $t$ . For each time interval, the utterance is represented as a vector of audio features,  $\vec{x}_t$ . Hannun et al, in their famous Deep Speech neural model, use spectrograms as features, such that  $x_{t,p}$  is the power of the  $p$ th frequency bin the the audio frame at time  $t$  [6].

Another approach is to take the modulus of a given signal's the STFT as treat that spectrogram  $X$  as a matrix to be factorized [9, 10]. NMF is particularly useful in this domain because the source components (the results of the factorization) are also interpreted as spectrograms, which must have non-negative entries.

## 4.3 Graph

Since a network of edges and vertices  $G = (V, E)$  can be represented as an *adjacency matrix*  $A \in \{0, 1\}^{|V| \times |V|}$  with  $A_{i,j} = 1$  if  $(i, j) \in E$  and  $A_{i,j} = 0$  otherwise, graphs are another

domain well suited for low dimensional representations.<sup>1</sup> Factorizing this adjacency matrix is equivalent to clustering the graph, which is a central problem to graph theory [7]. A given clustering  $C = \{C_1, C_2, \dots, C_k\}$  of a graph is a partition of the vertices and can be evaluated based on its *connectivity*:

$$\text{connectivity}(G, C) = \frac{\#\{(i, j) \in E | (i, j) \in C\} + \#\{(i, j) \notin E | (i, j) \notin C\}}{|V|^2}$$

The clustering  $C$  can be formulated as a membership matrix  $M \in \{0, 1\}^{n \times k}$  with  $M_{ir} = 1$  if  $i \in C_r$  and  $X_{ir}$  if  $i \notin C_r$ . Thus, the connectivity of  $G$  under  $C$  as defined above can be rewritten:

$$\begin{aligned} \text{connectivity}(G, C) &= 1 - \frac{\sum_{i,j} \left( A_{ij} - \sum_{r=1}^k X_{ir} X_{jr} \right)^2}{|V|^2} \\ &= 1 - \frac{\|A - XX^T\|_F^2}{|V|^2} \end{aligned}$$

Thus finding a clustering  $C$  that maximizes the connectivity is equivalent to the minimization problem

$$\min_{X \in \{0,1\}^{n \times k}} \|A - XX^T\|_F^2$$

However, finding one such  $X$  is NP-Complete [23]. So we can relax the constraints to try to solve an easier problem we are more used to

$$\min_{X \in \mathbb{R}_+^{n \times k}} \|A - XX^T\|_F^2$$

which just the standard optimization for non-negative matrix factorization for the symmetric case. Since the  $X$  we are solving for in this case is no longer binary, but rather contains reals, it is considered a *weak membership matrix*. That is, the larger  $X_{i,j}$ , the stronger membership of vertex  $i$  in cluster  $j$ .

## 4.4 Natural Language

With the vast amount of digital text being generated across the internet, methods for understanding and processing corpora of human language become necessary. Across mathematics and computer science, many techniques have been put forward that allow one to understand a body of text far too large to read herself. A successful method in this domain is *topic modelling*, whereby semantically cohesive subgroups of words can be identified. In particular, let  $\mathcal{C} = \{d_1, d_2, \dots, d_n\}$  be a collection of documents with a vocabulary  $\mathcal{V}$ . A *topic*  $t_i$  is a vector over the words in the vocabulary that represents a coherent high level notion in the corpus:

$$t_i = \{v_1^i, v_2^i, \dots, v_m^i\}$$

---

<sup>1</sup>For now, we will work with unweighted, undirected graphs, although these results generalize to other types of graphs as well.

where  $m$  is the size of the vocabulary. Topic modelling offers a powerful tool for understanding large amounts of text because they can discover latent semantic structure within text.

There are two primary techniques for learning these topics  $t_i$ . The first is LDA, a generative Bayesian statistical model which views each document  $d_j$  as a mixture of various topics. The second is non-negative matrix factorization, which aims to factor the document/word matrix into a document/topic and a topic/word matrix [13]. The focus of this thesis will be NMF, because of its relation to linear algebra, and its deep visual and conceptual intuition.

Given our  $n$  documents with vocabulary  $\mathcal{V}$  of size  $m$ , we construct a matrix  $X \in \mathbb{R}^{n \times m}$  where  $X_{i,j}$  is the number of occurrences of word  $j$  in document  $i$ . For a given inner dimension  $k$ , we seek to factor  $X$  into two matrices  $A$  and  $S$  such that

$$X \approx AS$$

where  $A \in \mathbb{R}^{n \times k}$  is the document/topic matrix and  $S \in \mathbb{R}^{k \times m}$  is the topic/word matrix. When we impose that  $A$  and  $S$  must be non-negative, a strong intuition emerges. In particular, the  $(i, j)$ th entry of  $A$  corresponds to the proportion of topic  $j$  in document  $i$  and the  $(i, j)$ th entry of  $S$  corresponds to the relevance of word  $j$  in topic  $i$ .

# Chapter 5

## Semi-Supervised NNMF

The methods above are unsupervised, which means the algorithms are designed to reveal latent structure within the unlabeled data. Many times however, data is accompanied with labels and the new algorithmic objective is to assign new unobserved data into appropriate labels. In the semi-supervised case, when only some of the data is labeled, the objective is similar to the unsupervised case. However, the known class labels can be incorporated into the model to improve classification performance [14]. In this section, we will explore an extension of NNMF where some of the class labels are known, and this information is used to enhance the performance of the NNMF. This *semi-supervised* NNMF learns a one-versus-all separating hyperplane for the observations [14].

### 5.1 Approach

In many cases, in addition to  $X \in \mathbb{R}^{n \times m}$  we also have a label matrix  $Y \in \mathbb{R}^{n \times k}$ , where  $k$  is the number of classes and  $Y_{i,j}$  is 1 if document  $i$  is in class  $j$  and 0 otherwise. Given  $B \in \mathbb{R}^{k \times r}$ , a basis matrix for  $Y$ , and  $L \in \mathbb{R}^{k \times n}$ , a weight matrix to handle missing labels, then the energy function for SSNMF is as follows

$$E = ||(X - AS)||^2 + \lambda ||L \circ (Y - BS)||^2$$

where  $\lambda$  is a tradeoff parameter that governs the importance of the supervised term.

In the same vein of Algorithm 1, this energy function yields a convex optimization prob-

lem solved by the following algorithm.

**Input:**  $k=0$ ; Initialize  $A^0, S^0, B^0$

**repeat**

$$A^{k+1} = A^k \circ \frac{X S^k}{A^k (S^k)^T S^k}$$

$$B^{k+1} = B^k \circ \frac{(L \circ Y) S^k}{(L \circ B (S^k)^T) S^k}$$

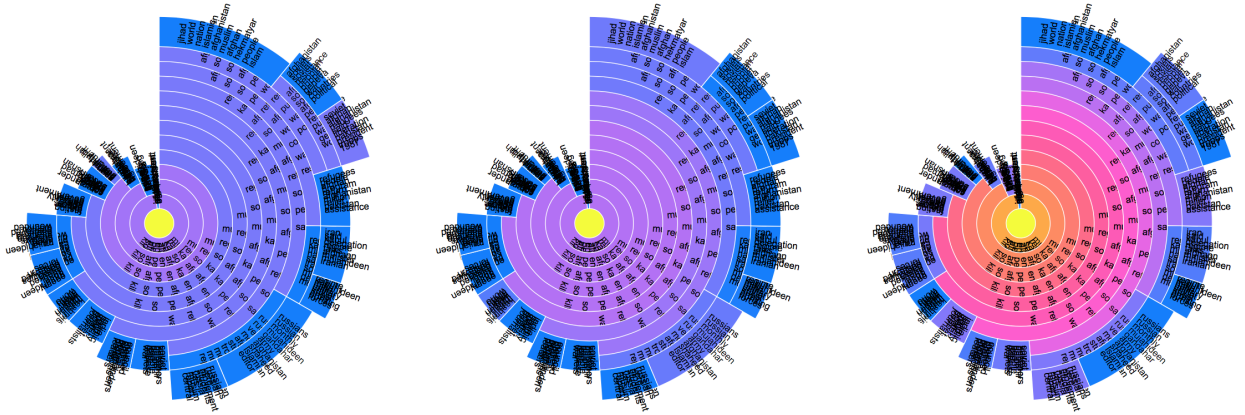
$$S^{k+1} = S^k \circ \frac{(A^{k+1})^T X + \lambda B^T (L \circ Y)}{(A^{k+1})^T A S + \lambda B^T (L \circ B S)}$$

$$k = k + 1$$

**until** *Stopping condition*;

**Algorithm 2:** Multiplicative Update for Semi-Supervised NMF

## 5.2 Visualization



# Chapter 6

## Hierarchical Representation

Many times, the data we wish to represent exhibits hierarchical structure. One such example is with the topics found in text data. Since topics correspond to semantic ideas, it makes sense that they would be organized in hierarchically. For example, in the 20 News Group data set, there are two separate topics for the Cincinnati Reds and Toronto Blue Jays, two baseball teams. Thus one could think about a supertopic, baseball, that branches into the two topics the standard NMF finds.

### 6.1 Approach

For a given document matrix  $V$ , we use the python library `scikitlearn` to decompose  $V$  into document/topic matrix  $W$  and topic/word matrix  $H$  such that

$$V \approx WH.$$

The `scikitlearn` implementation uses alternating gradient descent with the following objective function to generate optimal guesses for  $W$  and  $H$ .

$$c(H, W) = \frac{1}{2} \|X - WH\|_{fro}^2 + \alpha\lambda \|W\|_1 + \alpha\lambda \|H\|_1 + \frac{1}{2}\alpha(1 - \lambda) \|W\|_{fro}^2 + \frac{1}{2}\alpha(1 - \lambda) \|H\|_{fro}^2$$

where  $\|\cdot\|_{fro}$  is the Frobenius norm,  $\|\cdot\|_1$  is the L1 norm,  $\lambda$  is the L1 ratio and  $\alpha$  is a free parameter.

From the  $N$  topics  $t_n$  for  $n \in \{1 \cdots N\}$ <sup>1</sup>, we populate an adjacency matrix  $A$  where

$$A_{i,j} = \frac{T_i \cdot T_j}{\|T_i\| \|T_j\|}$$

is the cosine similarity between topics  $i$  and  $j$ . We then define a *threshold vector*  $\sigma$  by sorting all the elements of  $A$ .

$$\sigma = \{\sigma_1, \sigma_2, \cdots \sigma_{N^2} \mid 0 \leq \sigma_i \leq \sigma_j \leq 1 \forall i \leq j \text{ and } \sigma_k \in A\}$$

---

<sup>1</sup>observe that  $t_n$  is simply the  $n$ th row of  $H$

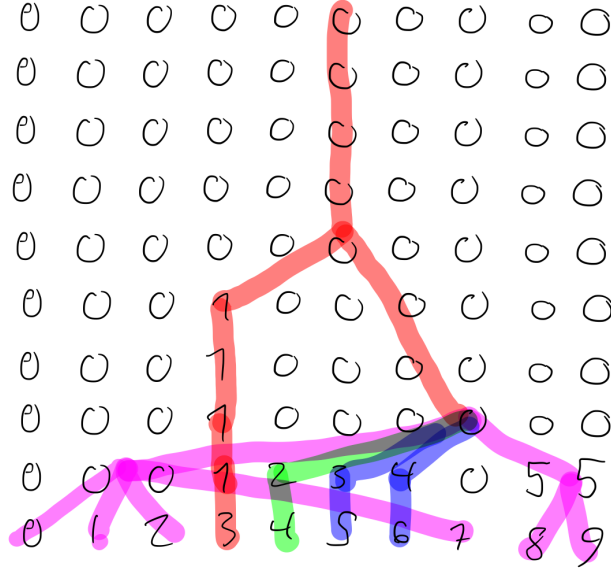


Figure 6.1: How the tree structure is formed for the connected component vectors

We then create an array of graphs  $A^{(k)}$  thresholded using the values of  $\sigma$ , such that

$$A_{i,j}^{(k)} = \begin{cases} 1 & \text{if } A_{i,j} > \sigma_k \\ 0 & \text{otherwise.} \end{cases}$$

Observe that  $A^{(1)}$  is the fully connected graph and  $A^{(N^2)}$  is the completely disconnected graph. By looking at the connected components of a given graph,

$$c(A^{(j)}) = \{c_1^j, c_2^j, \dots, c_i^j, \dots, c_N^j\}$$

where  $c_i = k$  means that the  $i$ th vertex is in the  $k$ th order component, we can formulate a tree structure (see Figure 1). For example, say  $N = 8$  and we have

$$c(A^{(j)}) = \{0, 0, 0, 0, 1, 1, 1, 1\}$$

$$c(A^{(j+1)}) = \{0, 0, 0, 0, 1, 1, 2, 2\}$$

This means that  $A^{(j)}$  has two connected components, ordered 0 (with vertices 1,2,3,4) and 1 (with vertices 5,6,7,8) and that  $A^{(j+1)}$  has three connected components, ordered 0 (with vertices 1,2,3,4), 1 (with vertices 5 and 6) and 2 (with vertices 7 and 8). Thus there is a branch from the connect component 1 in  $A^{(j)}$  to the connected components 1 and 2 in  $A^{(j+1)}$ . By greedily repeating this iterative algorithm starting with  $A^{(1)}$ <sup>2</sup> as the root, we produce the tree of topics. Observe that at this stage, all the leaf nodes correspond to actual topics  $t_n$ . We formulate the topic vectors for the parent nodes by additive percolating up the tree. That is, for a given parent topic  $\tau$  with children  $\tau_1, \dots, \tau_k$  we simply have

$$\tau = \sum_i \tau_i$$

<sup>2</sup>which has by definition only a single connected component and so  $c(A^{(1)}) = \{0, \dots, 0\}$



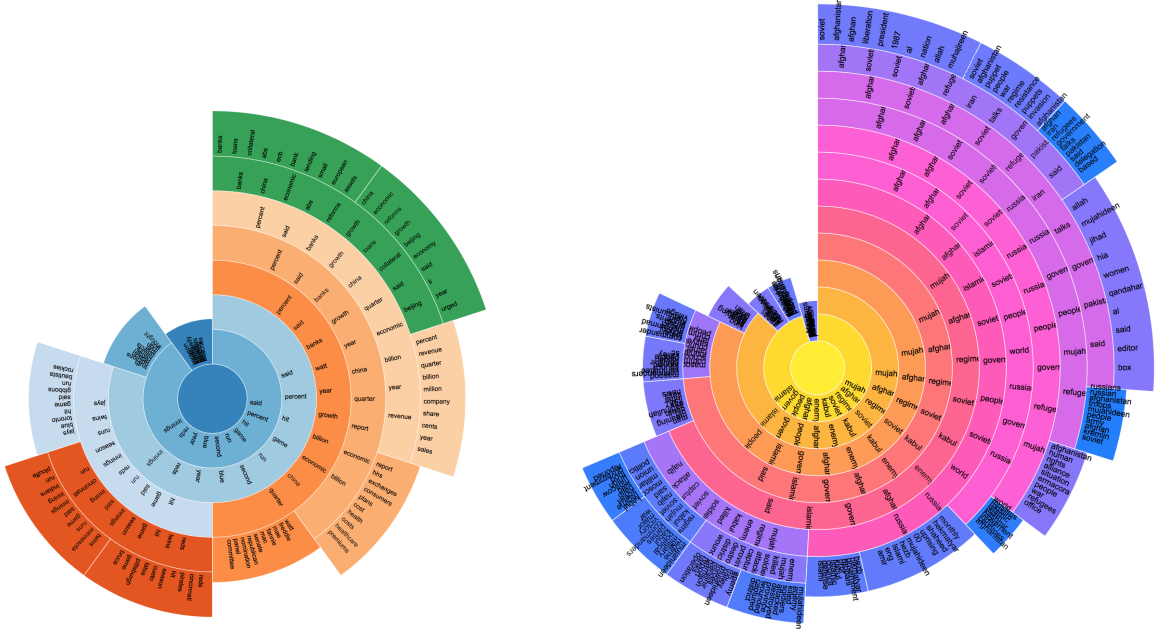


Figure 6.2: Left: Visualization of hierarchical topics in standard NMF; Right: Visualization of hierarchical topics in SSNMF;

## 6.2 Visualization

I use the d3.js Sunburst implementation to visualize the hierarchical topic model. Arcs at the same level represent discrete topics. A topic on an inner layer that encompasses multiple outer topics represent a super-topic. For example, in Figure 3, the two outer green topics that represent European banking and Chinese banking representatively (with top words

{banks, loans, collateral, abs, ecb bank, lending, small, european, asset}

and

{china, economic, reforms, growth, beijing, economy, said, li, year, urged})

merge into the inner super topic with top words

{banks, china, economic, abs, reforms, growth, loans, collateral, said, beijing}.

The visualization is responsive, dynamic and available at <http://www1.cmc.edu/pages/faculty/BHunter/ziv.html>

Next, I extended this visualization to the Semi-Supervised domain using the Afghan Dataset. Here each document has associated with it a class  $C \in \{1, \dots, k\}$  which in this case  $k = 3$ . Recall that the matrix  $B \in \mathbb{R}^{k \times r}$  in the semi-supervised NMF model is multiplied by  $S$  to obtain an approximation for  $Y$ , the label matrix. Thus the  $i, j$ th entry of  $B$  can be interpreted in the weighted importance of topic  $i$  in predicting class  $j$ . Thus I sum over the columns of  $B$  to capture how important topic  $i$  is predicting classes in general. After normalizing, we get a color value  $c_i$  for topic  $i$ , such that

$$c_i = \sum_j B_{i,j} / \sum_{i,j} B_{i,j}$$

where  $c_i = 1$  corresponds to yellow and  $c_i = 0$  corresponds to blue (see Figure 3 right)

# Chapter 7

## Deep Models

### 7.1 Approach

Semi-Supervised NMF as discussed above can be thought of as representing  $A$  in a low dimensional representation as  $S$ . In this framework,  $A$  is the function that maps the low dimensional representation to the original high dimensional representation. However, as the data becomes increasingly complex, it may have many hierarchy of attributes, each of which requires its own mappings. With the motivation in mind, Trigeorgis et al put forward the notion of a Demi-Semi NMF [20].

$$X \approx A_1 A_2 \cdots A_m S_m$$

This representation of the data can be achieved by recursively factorizing the low-dimensional representation at each level [20].

$$\begin{aligned} S_{m-1} &= A_m S_m \\ &\vdots \\ S_2 &\approx A_3 \cdots A_m S_m \\ S_1 &\approx A_2 \cdots A_m S_m \end{aligned}$$

We then consider the algorithms and notions in the domain of Deep Semi NMF as a model for Hierarchical NMF [3, 20]

### 7.2 Deep Neural Models

### 7.3 Visualization

# Chapter 8

## Discussion

# Bibliography

- [1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint arXiv:1512.02595*, 2015.
- [2] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.
- [3] Jennifer Flenner and Blake Hunter. A deep nonnegative matrix factorization.
- [4] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
- [5] DMBTL Griffiths and MIJJB Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.
- [6] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.
- [7] Ngoc-Diep Ho. *Nonnegative matrix factorization algorithms and applications*. PhD thesis, ÉCOLE POLYTECHNIQUE, 2008.
- [8] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [9] T Kawamoto, K Hotta, T Mishima, J Fujiki, M Tanaka, and T Kurita. Estimation of single tones from chord sounds using non-negative matrix factorization. *Neural Network World*, 10(3):429–436, 2000.
- [10] Sara Krause-Solberg and Armin Iske. Non-negative dimensionality reduction for audio signal separation by nnmf and ica. In *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pages 377–381. IEEE, 2015.
- [11] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

- [12] Daniel D Lee and H Sebastian Seung. Unsupervised learning by convex and conic coding. *Advances in neural information processing systems*, pages 515–521, 1997.
- [13] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [14] Hyekyoung Lee, Jiho Yoo, and Seungjin Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2010.
- [15] Nikos K Logothetis and David L Sheinberg. Visual object recognition. *Annual review of neuroscience*, 19(1):577–621, 1996.
- [16] Marvin Minsky. A framework for representing knowledge. 1975.
- [17] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.
- [18] Stephen E Palmer. Hierarchical structure in perceptual representation. *Cognitive psychology*, 9(4):441–474, 1977.
- [19] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.
- [20] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Bjoern Schuller. A deep semi-nmf model for learning hidden representations. In *ICML*, pages 1692–1700, 2014.
- [21] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [22] Shimon Ullman et al. *High-level vision: Object recognition and visual cognition*, volume 2. MIT press Cambridge, MA, 1996.
- [23] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.
- [24] E Wachsmuth, MW Oram, and DI Perrett. Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque. *Cerebral Cortex*, 4(5):509–522, 1994.