

Larson Hogstrom
MIT 18.335
Final Project
April 16, 2015

$$x_i^{k+1} \in \operatorname{argmin}_{\in X_i}$$

Outline

- Introduction + definition
- cost functions: Euclidian distance, Frobenius norm, KL divergence, Renyi's divergence. Can you construct a situation in which certain norms are better than others?
- update rules: multiplicative update, ALS method, gradient methods
- computational comparison: flop counts, accuracy, size of inputs
- applications to calculating metagenes

Introduction

This report introduces the framework for parts-based representations using NMF and focuses on the algorithms and numerical aspects of computation.

definition For a nonnegative matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, select a low-rank approximation of size k such that there are two nonnegative matrices $\mathbf{W} \in \mathbb{R}^{m \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times n}$ which minimizes a function such as

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\mathbf{A} - \mathbf{WH}\|_F^2$$

Other commonly used objective functions include Euclidian distance and Kullback-Leibler (KL) divergence. KL can be extended to a more general information-based framework using Renyi's divergence. (Devarajan, 2005). Here, a single parameter α is used to represent a continuum of distance measures and KL arises as a special case as $\alpha \rightarrow 1$.

$$KL(V||WH) = \sum_{ij} [V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}]$$

more text

Fundamental Algorithms

One of the first and most widely adopted algorithms for NMF is the multiplicative update rule. This takes the general form:

Data: Input data matrix: $\mathbf{A} \in \mathbb{R}^{m \times n}$

Result: nonnegative factorization of \mathbf{A} using k components, creating matrices $\mathbf{W} \in \mathbb{R}^{m \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times n}$

initialization;

$\mathbf{W} \leftarrow$ random dense ($m \times k$) matrix

$\mathbf{H} \leftarrow$ random dense ($k \times n$) matrix

for $i = 1$ to *maxiter* **do**

$\mathbf{H} = \mathbf{H} \cdot (\mathbf{W}^T \mathbf{A}) ./ (\mathbf{W}^T \mathbf{W} \mathbf{H})$

$\mathbf{W} = \mathbf{W} \cdot (\mathbf{A} \mathbf{H}^T) ./ (\mathbf{W} \mathbf{H} \mathbf{H}^T)$

end

Algorithm 1: multiplicative update

Often the
requires $O(mnk)$ work per iteration

Figure 1: A picture of a gull.

