Larson Hogstrom
MIT 18.335
Final Project
April 16, 2015

$$H_{aj} \leftarrow H_{aj}\frac{(W^TA)_{aj}}{(W^TWH)_{aj}} \qquad W_{ia} \leftarrow W_{ia}\frac{(W^TA)_{ia}}{(W^TWH)_{ia}} \tag{1}$$

Outline

-Introduction + definition

-cost functions: Euclidian distance, Frobinius 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 $\mathbf{A} \in \mathbb{R}^{\mathbf{mxn}}$, select a low-rank approximation of size k such that there are two nonnegative matrices $\mathbf{W} \in \mathbb{R}^{mxk}$ and $\mathbf{H} \in \mathbb{R}^{kxn}$ which minimizes a function such as

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

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 airises as a special case as $\alpha \to 1$.

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

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: $\mathbf{A} \in \mathbb{R}^{\mathbf{mxn}}$
Result: nonnegative factorization of $\mathbf{A}$ using k components, creating matrices $\mathbf{W} \in \mathbb{R}^{mxk}$ and $\mathbf{H} \in \mathbb{R}^{kxn}$
initialization;
$\mathbf{W} \leftarrow$ random dense (m x k) matrix
$\mathbf{H} \leftarrow$ random dense (k x n) matrix
for *i = 1 to maxiter* **do**
    $\mathbf{H} = \mathbf{H}. * (\mathbf{W^T A})./(\mathbf{W^T W H})$
    $\mathbf{W} = \mathbf{W}. * (\mathbf{A H^T})./(\mathbf{W H H^T})$
end

Algorithm 1: multiplicative update

Often the
requires O(mnk) work per iteration

Figure 1: A picture of a gull.