1: Apply the Power Method to Analyze the Document Data

The Power Method is implemented and performed to obtain the Principal Components (PCs). The first 3 PCs are listed in Table 1 with top 10 elements (words followed by values in parenthesis) in the respective eigenvectors. The error is also computed between an eigenvector and that by calling the standard function. The result is plot in Figure 1, where the x-axis represent the iteration index and the y-axis is the error computed based on 2-norm. To show the semantic meaning more visually, values of top elements in eigenvectors are compared. More specifically, the top 20 elements are considered for word cloud presentation; the size of a word is based on the ratio between the value of one element to the value of 20th element. For example, in PC1, the word ”problem” is roughly three times the word ”world” in terms of the value and hence the size. The results are presented in the Figure 2. Based on the presentation, it looks like words in PC1 are more related to information and technology, PC2 religion, PC3 university affairs, and PC4 has more diverse degrees of words.

Table 1: First Three PCs with top 10 elements

PC1	problem (0.298)	help (0.270)	question (0.252)	system (0.232)	fact (0.229)	email (0.206)	university (0.205)	course (0.202)	case (0.201)	world (0.199)
PC2	fact (0.281)	email (-0.258)	windows (-0.255)	god (0.252)	help (-0.219)	world (0.206)	question (0.190)	christian (0.185)	system (-0.185)	course (0.183)
PC3	problem (-0.571)	university (0.543)	email (0.372)	question (0.165)	state (0.165)	research (0.139)	science (0.139)	system (-0.134)	help (-0.119)	computer (0.111)

2: Design a Modified Power Method

Matrix multiplication is one of the classic problems that can be solved efficiently on parallel and distributed platforms. The power method is based on matrix multiplication, so it can be parallelized. The following algorithm shows a parallelized iterative method.

```
1. for $i = 1$ to $K$ do in parallel ($K$ is the number of the computing nodes)
    (a) read $i$-th part of the matrix and store in ”$A_i$” (the piece of data store in node $i$)
2. Central computer does $x =$ randomVector and broadcast $x$
3. While stop condition not satisfied do (e.g., the 2-norm of the difference between $x$'s from two consecutive iterations is less than a predefined value)
    (a) $x = \frac{x}{\|x\|_2}$.
    (b) for $i = 1$ to $K$ do in parallel
        • for $j = 1$ to rows do
            – $y[j] = 0$
            – for $k = 1$ to n do ($n$ is the dimension of the original $A$ matrix, corresponding to the number of the columns)
                i. $y[j] = y[j] + A_i[j][k] * x[k]$
    (c) Central computer gathers $y$'s from all computing nodes to get a full vector $x$
```

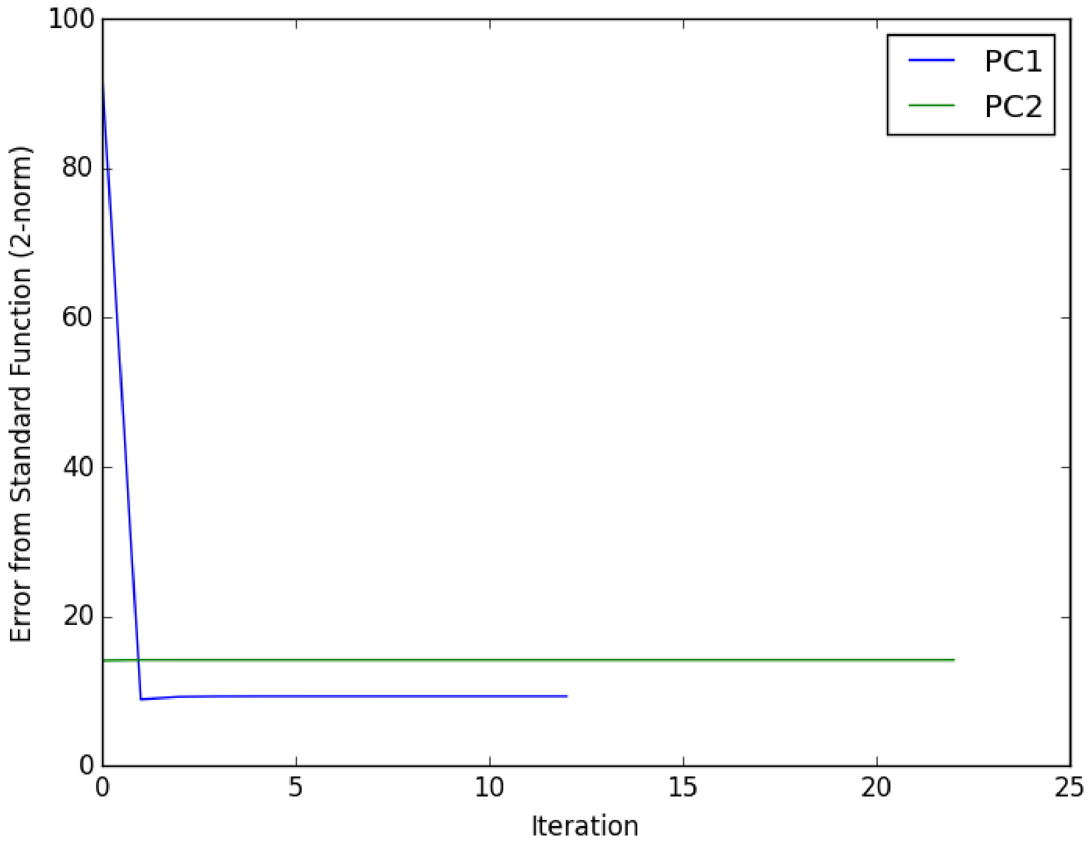


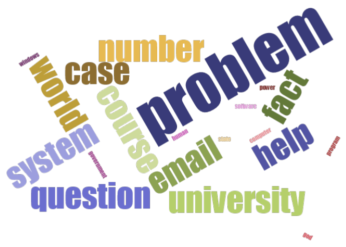
Figure 1: Evolution in Error

   (d) Central computer $\lambda = \|x\|_2$, update $x = \frac{x}{\|x\|_2}$ and broadcast it to all computing nodes

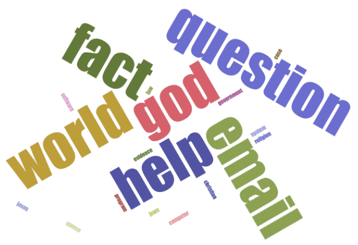
4. Eigenvalue $= \lambda$

5. Eigenvector $= x$

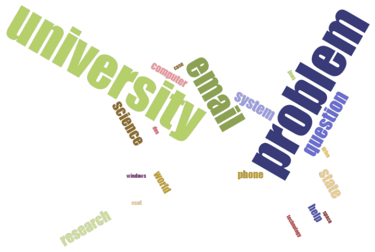
Matrix-vector multiplication follows the common parallelization theory on the well-known idea of dividing the rows of a given matrix according to the number of computing nodes (p), and all rows receive the vector. Thereafter, each node multiplies its associated rows by the given vector. For a matrix $A_{n\times n}$ and a vector $x_{n\times 1}$ each row should be multiplied by the vector's entries and sum of multiplications, replaced by the i-th row in the resulting vector.



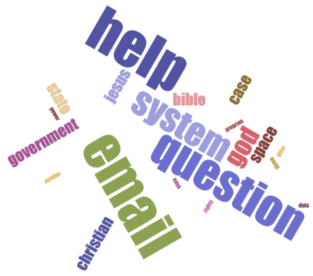
(a) PC1:



(b) PC2



(c) PC3



(d) PC4

Figure 2: Word Cloud Presentation of Relative Ratio of Elements in PCs