Learning Distributed Document Representations for Multi-Label Document Categorization

Nitish Gupta

B.Tech - M.Tech Dual Degree

Thesis Defense

Electrical Engineering

IIT Kanpur

May 16, 2015

Outline

Multi-Label Document Categorization

- DC introduction. Applications
- learning algorithm to learn classifer H from previous data to assign categories. Table example.
- Needs document represenation. Background on that.
- background on learning algos
- Motivation for distributed embeddings.
- background on distributed word represenations
- our model
- our dc lr model
- datasets
- results

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class. For E.g. an article on Music Piracy

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class. For E.g. an article on Music Piracy
- Task of assigning documents to one or more predefined categories is called *Multi-Label Document Categorization*

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class. For E.g. an article on Music Piracy
- Task of assigning documents to one or more predefined categories is called *Multi-Label Document Categorization*
- Wide range real-world applications :
  - Web-page tagging
  - Medical Patient Record Management
  - Wikipedia Article Management
  - Document Recommendation etc.

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class. For E.g. an article on Music Piracy
- Task of assigning documents to one or more predefined categories is called *Multi-Label Document Categorization*
- Wide range real-world applications :
  - Web-page tagging
  - Medical Patient Record Management
  - Wikipedia Article Management
  - Document Recommendation etc.
- Multi-label classification belongs to a general class of supervised learning algorithms where :

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class.
  For E.g. an article on Music Piracy
- Task of assigning documents to one or more predefined categories is called *Multi-Label Document Categorization*
- Wide range real-world applications :
  - Web-page tagging
  - Medical Patient Record Management
  - Wikipedia Article Management
  - Document Recommendation etc.
- Multi-label classification belongs to a general class of supervised learning algorithms where :
  - Training instances in the form of document-category pairs are used to learn a classifier $\mathcal{H}$

Intoduction to Multi-Label Document Categorization

- Text Documents usually belong to more than one conceptual class.
  For E.g. an article on Music Piracy
- Task of assigning documents to one or more predefined categories is called *Multi-Label Document Categorization*
- Wide range real-world applications :
  - Web-page tagging
  - Medical Patient Record Management
  - Wikipedia Article Management
  - Document Recommendation etc.
- Multi-label classification belongs to a general class of supervised learning algorithms where :
  - Training instances in the form of document-category pairs are used to learn a classifier $\mathcal{H}$
  - Learned classifier $\mathcal{H}$ is used to assign categories to new test documents

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$
- A set of categories $C = \{c_1, \ldots, c_{|C|}\}$

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$
- A set of categories $C = \{c_1, \ldots, c_{|C|}\}$
- Training data for n ($n < |D|$) documents, $\mathcal{T} = \{l_{d_1}, \ldots, l_{d_n}\}$

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$
- A set of categories $C = \{c_1, \ldots, c_{|C|}\}$
- Training data for n ($n < |D|$) documents, $\mathcal{T} = \{l_{d_1}, \ldots, l_{d_n}\}$
  Each label vector $l_{d_i} \in \{0, 1\}^{|C|}$ denotes relevance of categories to the document d_i

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$
- A set of categories $C = \{c_1, \ldots, c_{|C|}\}$
- Training data for n ($n < |D|$) documents, $\mathcal{T} = \{l_{d_1}, \ldots, l_{d_n}\}$
  Each label vector $l_{d_i} \in \{0, 1\}^{|C|}$ denotes relevance of categories to the document d_i

Example :

Documents	**Sports**	**Music**	**Arts**	**Technology**	**Literature**	**Politics**
d_1	0	0	1	0	1	0
d_2	0	1	1	0	0	1
d_3	1	0	0	1	0	1
d_4	×	×	×	×	×	×
d_5	×	×	×	×	×	×

Introduction to Multi-Label Document Categorization

Given,

- A set of documents $D = \{d_1, \ldots, d_{|D|}\}$
- A set of categories $C = \{c_1, \ldots, c_{|C|}\}$
- Training data for n ($n < |D|$) documents, $\mathcal{T} = \{l_{d_1}, \ldots, l_{d_n}\}$
  Each label vector $l_{d_i} \in \{0, 1\}^{|C|}$ denotes relevance of categories to the document d_i

Example :

Documents	**Sports**	**Music**	**Arts**	**Technology**	**Literature**	**Politics**
d_1	0	0	1	0	1	0
d_2	0	1	1	0	0	1
d_3	1	0	0	1	0	1
d_4	×	×	×	×	×	×
d_5	×	×	×	×	×	×

Using $\mathcal{T}$, D and C the learning algorithm learns a multi-label classifier $\mathcal{H}$ to estimate category label vectors, l_{d_j} ($j > n$) for the test documents.

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
   - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
   - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$
   - Vectors (v_{d_i}) should encode the semantic content of the documents

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
   - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$
   - Vectors (v_{d_i}) should encode the semantic content of the documents
   - Encoding documents in a k-dimensional space using such represenation is called the *Vector Space Model*

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
   - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$
   - Vectors (v_{d_i}) should encode the semantic content of the documents
   - Encoding documents in a k-dimensional space using such represenation is called the *Vector Space Model*
   - The complete document set D can be represented by a document representation matrix $\mathrm{D} \in \mathbb{R}^{k \times |D|}$

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
    - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$
    - Vectors (v_{d_i}) should encode the semantic content of the documents
    - Encoding documents in a k-dimensional space using such represenation is called the *Vector Space Model*
    - The complete document set D can be represented by a document representation matrix $\mathrm{D} \in \mathbb{R}^{k \times |D|}$

In this thesis, we focus on learning efficient document representations, D

Introduction to Multi-Label Document Categorization

Document Categorization task has the following two components :

1. *Learning Document Representations* : Representing text documents using numerical vectors that are inputs to the multi-label classifier $\mathcal{H}$
   - Each document $d_i \in D$ is represented using a vector $v_{d_i} \in \mathbb{R}^k$
   - Vectors (v_{d_i}) should encode the semantic content of the documents
   - Encoding documents in a k-dimensional space using such represenation is called the *Vector Space Model*
   - The complete document set D can be represented by a document representation matrix $\mathrm{D} \in \mathbb{R}^{k \times |D|}$

In this thesis, we focus on learning efficient document representations, D

2. *Learning Algorithm* : Algorithm to learn the multi-label classifier $\mathcal{H}$

Background on Learning Algorithms

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier
   - k-Nearest Neighbor (k-NN)

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier
   - k-Nearest Neighbor (k-NN)
   - Linear Least Square Fit

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier
   - k-Nearest Neighbor (k-NN)
   - Linear Least Square Fit
   - Decision Trees

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier
   - k-Nearest Neighbor (k-NN)
   - Linear Least Square Fit
   - Decision Trees
   - Generative Probabilistic Models

Background on Learning Algorithms

1. *Learning Multiple Binary Classifiers* :
   Algorithms that treat each category assignment independently and learn multiple binary classifiers, one for each category, to make the category assignments
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Neural Networks
   - Naive Bayes
2. *Learning Single Joint Classifier* :
   Algorithms that jointly assign all the categories to a document d_i, i.e. estimate the complete label vector l_{d_i} using a single classifier
   - k-Nearest Neighbor (k-NN)
   - Linear Least Square Fit
   - Decision Trees
   - Generative Probabilistic Models

   Learning a single joint classifier is usually better as it is able to exploit the category correlations

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf
  - idf

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf
  - idf
  - tf-idf

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
    - tf
    - idf
    - tf-idf

Drawbacks with the Bag-of-Words model

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf
  - idf
  - tf-idf

Drawbacks with the Bag-of-Words model

- r to the i size vectors

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf
  - idf
  - tf-idf

Drawbacks with the Bag-of-Words model

- r to the i size vectors
- each element indicates presence or absence

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

Background on Text Representation

Most common method to learn vector representations for a set of documents is the *Bag of Words (BOW)* model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as
  - tf
  - idf
  - tf-idf

Drawbacks with the Bag-of-Words model

- r to the i size vectors
- each element indicates presence or absence
- weighing techniques such as

TODO: feature selection. Word Embeddings. Our document embedding. Our docu cate. Datasets. Results End

References

[1] G. Salton and C.-S. Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.