Dataless Classification

Paper Presentation (CS 6370)

Ameet Deshpande [1] Vedant Somani [2]

April 16, 2018

Outline

1. Building Blocks
   - Bag of Words
   - Explicit Semantic Analysis
   - Naive Bayes Classifier
2. Dataless Classification
   - Motivation
   - Label Expansion
   - On the Fly Classification
   - Leveraging Unlabeled Data
   - Domain Adaptation

Outline

1. Building Blocks
   - Bag of Words
   - Explicit Semantic Analysis
   - Naive Bayes Classifier
2. Dataless Classification
   - Motivation
   - Label Expansion
   - On the Fly Classification
   - Leveraging Unlabeled Data
   - Domain Adaptation

Bag of Words

- Bag of words (BOW) is a Naive way of representing documents.
- It just counts the number of occurrences of each words and does not pay heed to their positioning.
- It is used to serve the purpose of a baseline in this work.
- As might be apparent, BOW vector representations are useful only when the exact words required to be retrieved are present in the document. More on this later.

Bag of Words Vector representations

- A document D is represented by a vector of size $|V|$, where V represents the Vocabulary under consideration.

Bag of Words Vector representations

- A document D is represented by a vector of size $|V|$, where V represents the Vocabulary under consideration.

Consider two documents

- $D1$: I am Raju
- $D2$: I love Kaju

Bag of Words Vector representations

- A document D is represented by a vector of size $|V|$, where V represents the Vocabulary under consideration.

Consider two documents

- $D1$: I am Raju
- $D2$: I love Kaju

The following will be vector representations of the two documents.

Doc	I	am	love	Raju	Kaju
$D1$	1	1	0	1	0
$D2$	1	0	1	0	1

Outline

1. Building Blocks
   - Bag of Words
   - Explicit Semantic Analysis
   - Naive Bayes Classifier
2. Dataless Classification
   - Motivation
   - Label Expansion
   - On the Fly Classification
   - Leveraging Unlabeled Data
   - Domain Adaptation

Explicit Semantic Analysis

- Humans interpret a word in a very large context comprising of background knowledge.

Explicit Semantic Analysis

- Humans interpret a word in a very large context comprising of background knowledge.
- Explicit Semantic Analysis (ESA) is a method which represents text (*words*) in a high dimensional space of Wikipedia derived concepts.

Explicit Semantic Analysis

- Humans interpret a word in a very large context comprising of background knowledge.
- Explicit Semantic Analysis (ESA) is a method which represents text (*words*) in a high dimensional space of Wikipedia derived concepts.
- For each word, an inverted index of concepts - in which the word appeared - is stored.

Explicit Semantic Analysis

- Humans interpret a word in a very large context comprising of background knowledge.
- Explicit Semantic Analysis (ESA) is a method which represents text (*words*) in a high dimensional space of Wikipedia derived concepts.
- For each word, an inverted index of concepts - in which the word appeared - is stored.
- A TFIDF matrix is generated.

Explicit Semantic Analysis

- Humans interpret a word in a very large context comprising of background knowledge.
- Explicit Semantic Analysis (ESA) is a method which represents text (*words*) in a high dimensional space of Wikipedia derived concepts.
- For each word, an inverted index of concepts - in which the word appeared - is stored.
- A TFIDF matrix is generated.
- The concepts associated with the word are then scored based on the TFIDF vector for the input word, and the relevance of the concept to the word.

Explicit Semantic Analysis

- Following is an example of ESA representations.

Word	*Concept*$_1$ *Score*$_1$	*Concept*$_2$ *Score*$_2$	*Concept*$_3$ *Score*$_3$	...
Mars	**planet** 0.9	**Solar System** 0.85	**jupiter** 0.3	...
explorer	**adventurer** 0.89	**pioneer** 0.7	**vehicle** 0.2	...

Explicit Semantic Analysis

- All the words will have some score associated with each concept of Wikipedia. Therefore while computing the true relatedness of words, a cosine similarity measure is used over the word representations.

Explicit Semantic Analysis

- All the words will have some score associated with each concept of Wikipedia. Therefore while computing the true relatedness of words, a cosine similarity measure is used over the word representations.
- However, not all the scores associated with concepts are significant. For example, the score for the concept "Sachin Tendulkar" for the word *neutron* might be very low.

Explicit Semantic Analysis

- All the words will have some score associated with each concept of Wikipedia. Therefore while computing the true relatedness of words, a cosine similarity measure is used over the word representations.
- However, not all the scores associated with concepts are significant. For example, the score for the concept "Sachin Tendulkar" for the word *neutron* might be very low.
- Thus in practice, scores for all the concepts are not stored, instead only the scores associated with a k (say 100) most related concepts are stored.

Outline

Naive Bayes Classifier

- This serves as a baseline in this work.
- $P(Class|D) = \dfrac{P(d_1|Class)\dots P(d_n|Class) \times P(Class)}{P(D)}$

Naive Bayes Classifier

- This serves as a baseline in this work.
- $P(Class|D) = \dfrac{P(d_1|Class)\dots P(d_n|Class) \times P(Class)}{P(D)}$
- Usually $d_1, d_2, \dots, d_n \in Words(Document)$.

Naive Bayes Classifier

- This serves as a baseline in this work.
- $P(Class|D) = \dfrac{P(d_1|Class)\ldots P(d_n|Class) \times P(Class)}{P(D)}$
- Usually $d_1, d_2, \ldots, d_n \in Words(Document)$.
- But let's convince ourselves that $d_i's$ can be a single element of any vector representation of the document.

Naive Bayes Classifier

- This serves as a baseline in this work.
- $P(Class|D) = \dfrac{P(d_1|Class)\dots P(d_n|Class) \times P(Class)}{P(D)}$
- Usually $d_1, d_2, \dots, d_n \in Words(Document)$.
- But let's convince ourselves that $d_i's$ can be a single element of any vector representation of the document.
- Usually the features of the vector representation are words. But we could use the ESA representation (and it could perhaps be more useful).

Naive Bayes Classifier

- Here is how ESA representations can be used to classify documents.

Naive Bayes Classifier

- Here is how ESA representations can be used to classify documents.

Consider two documents

- $D1$: Tesla launches Falcon
- $D2$: The Eagle has landed

Naive Bayes Classifier

- Here is how ESA representations can be used to classify documents.

Consider two documents

- $D1$: Tesla launches Falcon
- $D2$: The Eagle has landed

The following could be the ESA representations of the two documents.

Doc	Space	Cars	Birds	Musk	Armstrong
$D1$	0.02	0.02	0.01	0.10	0.00
$D2$	0.02	0.00	0.01	0.00	0.10

Naive Bayes Classifier

- Here is how ESA representations can be used to classify documents.

Consider two documents

- $D1$: Tesla launches Falcon
- $D2$: The Eagle has landed

The following could be the ESA representations of the two documents.

Doc	Space	Cars	Birds	Musk	Armstrong
$D1$	0.02	0.02	0.01	0.10	0.00
$D2$	0.02	0.00	0.01	0.00	0.10

- 10-100 examples are used to train a Supervised Classifier and results are compared against the Dataless classifier.

Disclaimer

No Data used?

It is important to remember that a large amount of data has already been used to train the model. Wikipedia articles and ESA are used to get vector representations. This approach is not Dataless in that sense

Disclaimer

No Data used?

It is important to remember that a large amount of data has already been used to train the model. Wikipedia articles and ESA are used to get vector representations. This approach is not Dataless in that sense

On the fly Classification

But post the training procedure, the classifier can categorize documents into labels which it has never been trained on before. It is definitely *Dataless* in that sense.

Disclaimer

No Data used?

It is important to remember that a large amount of data has already been used to train the model. Wikipedia articles and ESA are used to get vector representations. This approach is not Dataless in that sense

On the fly Classification

But post the training procedure, the classifier can categorize documents into labels which it has never been trained on before. It is definitely *Dataless* in that sense.

Example

We will now see a demonstration to get a feel of how this procedure works.

Outline

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Consider the following article

UFO sightings have been reported throughout recorded history and in various parts of the world, raising questions about life on other planets and whether extraterrestrials have visited Earth.

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Consider the following article

UFO sightings have been reported throughout recorded history and in various parts of the world, raising questions about life on other planets and whether extraterrestrials have visited Earth.

- How is it possible for us to classify the article with ease?

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Consider the following article

UFO sightings have been reported throughout recorded history and in various parts of the world, raising questions about life on other planets and whether extraterrestrials have visited Earth.

- How is it possible for us to classify the article with ease?
- We humans seem to use the semantics of the labels representing the classes. We "know" that the label means.

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Consider the following article

UFO sightings have been reported throughout recorded history and in various parts of the world, raising questions about life on other planets and whether extraterrestrials have visited Earth.

- How is it possible for us to classify the article with ease?
- We humans seem to use the semantics of the labels representing the classes. We "know" that the label means.
- But say we were training a supervised classifier for this task. Usual Machine Learning approaches just treat the labels/classes as 0/1.

Motivation

- Say we are posed with the task of classifying documents into two categories, {Alien Invasion, Rocket Launch}

Consider the following article

UFO sightings have been reported throughout recorded history and in various parts of the world, raising questions about life on other planets and whether extraterrestrials have visited Earth.

- How is it possible for us to classify the article with ease?
- We humans seem to use the semantics of the labels representing the classes. We "know" that the label means.
- But say we were training a supervised classifier for this task. Usual Machine Learning approaches just treat the labels/classes as 0/1.
- Can we use an algorithm which does not throw away the meaning in the labels? This could be one way of injecting World Knowledge into the system.

Semantic Representation of Categories

- Now that we have established that a rich semantic representation of a category helps, let's see how we can get such a representation.

Semantic Representation of Categories

- Now that we have established that a rich semantic representation of a category helps, let's see how we can get such a representation.
- Say an oracle provides for each category i, a document w_i.

Semantic Representation of Categories

- Now that we have established that a rich semantic representation of a category helps, let's see how we can get such a representation.
- Say an oracle provides for each category i, a document w_i.
- If the representation of the document to be classified, $\varphi(d)$, is closer to the correct class w_i rather than a wrong class w_j, we can classify it successfully.

Semantic Representation of Categories

- Now that we have established that a rich semantic representation of a category helps, let's see how we can get such a representation.
- Say an oracle provides for each category i, a document w_i.
- If the representation of the document to be classified, $\varphi(d)$, is closer to the correct class w_i rather than a wrong class w_j, we can classify it successfully.

$$||w^i - \varphi(d)|| \leq ||w^j - \varphi(d)|| - \gamma, \forall j \neq i$$

Semantic Representation of Categories

- Since an oracle is imaginary and sadly we have to do all the work, we assume that the label name l_i is a good enough approximation of the oracle document w^i.

Semantic Representation of Categories

- Since an oracle is imaginary and sadly we have to do all the work, we assume that the label name l_i is a good enough approximation of the oracle document w^i.
- But it can be proved that the approximation has room for some error and we can get a good classifier even when the approximation is a little off.

Semantic Representation of Categories

- Since an oracle is imaginary and sadly we have to do all the work, we assume that the label name l_i is a good enough approximation of the oracle document w^i.
- But it can be proved that the approximation has room for some error and we can get a good classifier even when the approximation is a little off.

$$||\varphi(d) - \varphi(\{l_i\})|| \leq ||\varphi(d) - \varphi(\{l_j\})|| - \gamma + 2\eta, \forall j \neq i$$

where η is the error made in approximating the oracle document.

Outline

Label Expansion

- To reduce the error being made in approximating the oracle document, we need to make sure the label names are representative of the category.

Label Expansion

- To reduce the error being made in approximating the oracle document, we need to make sure the label names are representative of the category.
- A procedure called label expansion is used to ensure that. Instead of using a single word to represent a category, a handful of representative words are used.

Label Expansion

- To reduce the error being made in approximating the oracle document, we need to make sure the label names are representative of the category.
- A procedure called label expansion is used to ensure that. Instead of using a single word to represent a category, a handful of representative words are used.

Expanded Labels for Newsgroup Categories

talk.politics.guns $\rightarrow$ {politics, guns}
soc.religion.christian $\rightarrow$ {society, religion, christianity, christian}
comp.sys.mac.hardware $\rightarrow$ {computer, systems, mac, apple, hardware}
sci.crypt $\rightarrow$ {science, cryptography}

Outline

On the Fly Classification

- This is a technique that does not use *any* data post the training procedure.
- Nearest Neighbor Classification is used to predict the correct category.

On the Fly Classification

- This is a technique that does not use *any* data post the training procedure.
- Nearest Neighbor Classification is used to predict the correct category.
- Consider the label set $\{l_1, l_2, \ldots, l_k\}$. Given a document's representation, the label whose representation is closest to that document is predicted.

$$\arg\min_i ||\varphi(l_i) - \varphi(d)||$$

On the Fly Classification

- This is a technique that does not use *any* data post the training procedure.
- Nearest Neighbor Classification is used to predict the correct category.
- Consider the label set $\{l_1, l_2, \ldots, l_k\}$. Given a document's representation, the label whose representation is closest to that document is predicted.

$$\arg\min_i ||\varphi(l_i) - \varphi(d)||$$

- Depending on if Naive Bayes representation is used or ESA representation is used, the classifier is called **NN-BOW** or **NN-ESA**
- The **NN-BOW** classifier can categorize a document successfully only if there are words common between the label and the document (?) but that is not the case with **NN-ESA**.

Outline

Leveraging Unlabeled Data

- There is often a lot of unlabeled data available. Though it is expensive to get labeled data, unlabeled data can be retrieved easily (scraping).

Leveraging Unlabeled Data

- There is often a lot of unlabeled data available. Though it is expensive to get labeled data, unlabeled data can be retrieved easily (scraping).
- Is it possible to harness this pool of documents starting with just the labels?

Leveraging Unlabeled Data

- There is often a lot of unlabeled data available. Though it is expensive to get labeled data, unlabeled data can be retrieved easily (scraping).
- Is it possible to harness this pool of documents starting with just the labels? **Bootstrapping**

Bootstrapping Algorithm

Algorithm 1 Bootstrap-φ. *Training a bootstrapped classifier for a feature representation φ, where φ could be Bag of Words or ESA.*

Let training set $T = \emptyset$
for all labels l_i **do**
    Add l_i to T with label i
end for
repeat
    Train a naive Bayes classifier NB on T
    for all d_i, a document in the document collection **do**
        If $y = NB.classify(\varphi(d_i))$ with high confidence
        Add d_i to T with label y
    end for
until No new training documents are added.

Co-training

- An influential paper [3] suggested that if there are two *independent* views of the data, both *self sufficient* in themselves, combining them could give better results in labeling the documents.

Co-training

- An influential paper [3] suggested that if there are two *independent* views of the data, both *self sufficient* in themselves, combining them could give better results in labeling the documents.
- How can this possibly be beneficial?

Co-training

- An influential paper [3] suggested that if there are two *independent* views of the data, both *self sufficient* in themselves, combining them could give better results in labeling the documents.
- How can this possibly be beneficial?
- Let X_1, X_2 represent the independent views of the data (D) and $f_1(X_1), f_2(X_2)$ represent the classifiers built on them.

Co-training

- An influential paper [3] suggested that if there are two *independent* views of the data, both *self sufficient* in themselves, combining them could give better results in labeling the documents.
- How can this possibly be beneficial?
- Let X_1, X_2 represent the independent views of the data (D) and $f_1(X_1), f_2(X_2)$ represent the classifiers built on them.
- Clearly f_1, f_2 depend on the documents they have been trained on. If there is a document on which the predicted labels do not agree, what should be done?

Co-training

- An influential paper [3] suggested that if there are two *independent* views of the data, both *self sufficient* in themselves, combining them could give better results in labeling the documents.
- How can this possibly be beneficial?
- Let X_1, X_2 represent the independent views of the data (D) and $f_1(X_1), f_2(X_2)$ represent the classifiers built on them.
- Clearly f_1, f_2 depend on the documents they have been trained on. If there is a document on which the predicted labels do not agree, what should be done?
- Push the document for later, or ignore it. Let's see what the algorithm looks like.

Co-training Algorithm

Algorithm 2 Co-training *We use the fact that BOW and ESA can independently classify the data quite well to induce a new classifier.*

1: Let training set $T^{BOW} = \emptyset, T^{ESA} = \emptyset$.
2: **for all** labels l_i **do**
3:     Add l_i to both T^{ESA} and T^{BOW} with label i
4: **end for**
5: **repeat**
6:     Train a naive Bayes classifier NB^{BOW} on T^{BOW}.
7:     Train a naive Bayes classifier NB^{ESA} on T^{ESA}.
8:     **for all** d_i, a document in the document collection **do**
9:         **if** Both NB^{BOW} and NB^{ESA} classify d_i with high confidence **then**
10:             Add d_i to T^{BOW} with label from NB^{BOW}
11:             Add d_i to T^{ESA} with label from NB^{ESA}
12:         **end if**
13:     **end for**
14: **until** No new training documents are added

We will look at this algorithm in detail again, for now let's focus on line 9 where the documents are added to respective training sets only if **both** the classifiers output the same label.

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.
- In this work, the document is added to the training set only if both the classifiers label the document with high confidence.

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.
- In this work, the document is added to the training set only if both the classifiers label the document with high confidence.
- In the original work, each classifier chooses p positive examples and n negative examples which it is most confident of (Binary Classification).

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.
- In this work, the document is added to the training set only if both the classifiers label the document with high confidence.
- In the original work, each classifier chooses p positive examples and n negative examples which it is most confident of (Binary Classification).
- The advantage of this is that the classifiers transfer knowledge to each other. How?

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.
- In this work, the document is added to the training set only if both the classifiers label the document with high confidence.
- In the original work, each classifier chooses p positive examples and n negative examples which it is most confident of (Binary Classification).
- The advantage of this is that the classifiers transfer knowledge to each other. How?
- Instead of ignoring a document which is not classified with high confidence by **both**, even if it is classified by one of them (say C_1), we can be sure that it is a legitimate classification.

Co-training Algorithm

- The training procedure in the original paper [3] was slightly different.
- In this work, the document is added to the training set only if both the classifiers label the document with high confidence.
- In the original work, each classifier chooses p positive examples and n negative examples which it is most confident of (Binary Classification).
- The advantage of this is that the classifiers transfer knowledge to each other. How?
- Instead of ignoring a document which is not classified with high confidence by **both**, even if it is classified by one of them (say C_1), we can be sure that it is a legitimate classification.
- This can be used in the next iteration of learning to train even C_2 and the confidence of classification for it will hopefully increase.

Co-training Algorithm

Algorithm 2 Co-training *We use the fact that BOW and ESA can independently classify the data quite well to induce a new classifier.*

```
Let training set T^{BOW} = ∅, T^{ESA} = ∅.
for all labels l_i do
    Add l_i to both T^{ESA} and T^{BOW} with label i
end for
repeat
    Train a naive Bayes classifier NB^{BOW} on T^{BOW}.
    Train a naive Bayes classifier NB^{ESA} on T^{ESA}.
    for all d_i, a document in the document collection do
        if Both NB^{BOW} and NB^{ESA} classify d_i with high confidence then
            Add d_i to T^{BOW} with label from NB^{BOW}
            Add d_i to T^{ESA} with label from NB^{ESA}
        end if
    end for
until No new training documents are added
```

Co-training Algorithm

Algorithm 2 Co-training *We use the fact that BOW and ESA can independently classify the data quite well to induce a new classifier.*

Let training set $T^{BOW} = \emptyset, T^{ESA} = \emptyset$.
for all labels l_i **do**
  Add l_i to both T^{ESA} and T^{BOW} with label i
end for
repeat
  Train a naive Bayes classifier NB^{BOW} on T^{BOW}.
  Train a naive Bayes classifier NB^{ESA} on T^{ESA}.
  for all d_i, a document in the document collection **do**
    if Both NB^{BOW} and NB^{ESA} classify d_i with high confidence **then**
      Add d_i to T^{BOW} with label from NB^{BOW}
      Add d_i to T^{ESA} with label from NB^{ESA}
    end if
  end for
until No new training documents are added

Catch: BOW and ESA representations are not really independent. Nevertheless, this was found to improve the classifier.

Outline

1. Building Blocks
   - Bag of Words
   - Explicit Semantic Analysis
   - Naive Bayes Classifier

2. Dataless Classification
   - Motivation
   - Label Expansion
   - On the Fly Classification
   - Leveraging Unlabeled Data
   - Domain Adaptation

Domain Adaptation

- Different domains tend to use different vocabularies to express the same meaning.

Domain Adaptation

- Different domains tend to use different vocabularies to express the same meaning.

Documents from different Domains

- $D1$: Manchester United floors Manchester City in yesterday's *football* match.
- $D2$: LA Galaxy wins derby 4-3 in Major League *Soccer*.

Domain Adaptation

- Different domains tend to use different vocabularies to express the same meaning.

Documents from different Domains

- *D*1: Manchester United floors Manchester City in yesterday's *football* match.
- *D*2: LA Galaxy wins derby 4-3 in Major League *Soccer*.

- The difference in vocabularies may restrict supervised classifier to be used in different domains. But with ESA representations, the words *Football* and *Soccer* may already be close to each other and this could help in generalization.

Results for Domain Adaptation

- Let's stare at a few results and deduce what is effecting Domain Adaptation.

	Model	Features	Accuracy
20NG → 20 NG	Supervised	BOW	0.97
Yahoo → 20 NG	Supervised	BOW	0.60
20NG → 20 NG	Supervised	ESA	0.96
Yahoo → 20 NG	Supervised	ESA	0.90
$\emptyset$ → 20NG	Dataless	ESA	0.96
Yahoo → Yahoo	Supervised	BOW	0.93
20NG → Yahoo	Supervised	BOW	0.89
Yahoo → Yahoo	Supervised	ESA	0.97
20NG → Yahoo	Supervised	ESA	0.96
$\emptyset$ → Yahoo	Dataless	ESA	0.94

Summary

We have looked at,

- Dataless Classification
- Less Data Classification

References I

Chang, Ming-Wei, et al. "Importance of Semantic Representation: Dataless Classification." AAAI. Vol. 2. 2008.

Gabrilovich, Evgeniy, and Shaul Markovitch. "Computing semantic relatedness using wikipedia-based explicit semantic analysis." IJcAI. Vol. 7. 2007.

Blum, Avrim, and Tom Mitchell. "Combining labeled and unlabeled data with co-training." Proceedings of the eleventh annual conference on Computational learning theory. ACM, 1998.