Predictive Analysis of Text: Concepts, Features, and Instances

Jaime Arguello
jarguell@email.unc.edu
Concepts from Domingo’s Paper

1. Representation + Parameter Optimization + Evaluation
2. Bias/Variance Trade-off + Overfitting
Predictive Analysis of Text

- **Objective**: developing and evaluating computer programs that automatically detect a particular concept in natural language text
Predictive Analysis
basic ingredients

1. **Training data:** a set of positive and negative examples of the concept we want to automatically recognize

2. **Representation:** a set of features that we believe are useful in recognizing the desired concept

3. **Learning algorithm:** a computer program that uses the training data to learn a predictive model of the concept
Predictive Analysis

basic ingredients

4. **Model**: a function that describes a predictive relationship between feature values and the presence of the concept

5. **Test data**: a set of previously unseen examples used to estimate the model’s effectiveness

6. **Performance metrics**: a set of statistics used to measure the predictive effectiveness of the model
Predictive Analysis
training and testing

training

labeled examples

machine learning algorithm

model

testing

new, unlabeled examples

model

predictions
## Predictive Analysis

**concept, instances, and features**

<table>
<thead>
<tr>
<th>instances</th>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>color</td>
<td>size</td>
</tr>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
</tr>
<tr>
<td>green</td>
<td>big</td>
<td>3</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>inf</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>red</td>
<td>big</td>
</tr>
</tbody>
</table>
Predictive Analysis
training and testing

<table>
<thead>
<tr>
<th>color</th>
<th>size</th>
<th>sides</th>
<th>equal sides</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>green</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>inf</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>4</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

labeled examples

<table>
<thead>
<tr>
<th>color</th>
<th>size</th>
<th>sides</th>
<th>equal sides</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>green</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>inf</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>4</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

new, unlabeled examples

<table>
<thead>
<tr>
<th>color</th>
<th>size</th>
<th>sides</th>
<th>equal sides</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>green</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>inf</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>4</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

predictions

<table>
<thead>
<tr>
<th>color</th>
<th>size</th>
<th>sides</th>
<th>equal sides</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>green</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>inf</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>blue</td>
<td>small</td>
<td>4</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>red</td>
<td>big</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

machine learning algorithm
Predictive Analysis questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
Learning algorithms can recognize some concepts better than others.

What are some properties of concepts that are easier to recognize?
Predictive Analysis

Concepts

- **Option 1:** can a human recognize the concept?
Predictive Analysis

concepts

• **Option 1**: can a human recognize the concept?
• **Option 2**: can two or more humans recognize the concept independently and do they agree?
Predictive Analysis

concepts

• **Option 1:** can a human recognize the concept?
• **Option 2:** can two or more humans recognize the concept independently and do they agree?
• **Option 2** is better.
• In fact, models are sometimes evaluated as an independent assessor
• How does the model’s performance compare to the performance of one assessor with respect to another?
  ▶ One assessor produces the “ground truth” and the other produces the “predictions”
Predictive Analysis measures agreement: percent agreement

- **Percent agreement**: percentage of instances for which both assessors agree that the concept occurs or does not occur

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>no</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

\[
\frac{(A + B)}{(A + B + C + D)}
\]
Predictive Analysis measures agreement: percent agreement

- **Percent agreement**: percentage of instances for which both assessors agree that the concept occurs or does not occur.

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>no</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

\[
\frac{(A + D)}{(A + B + C + D)}
\]
Percent agreement: percentage of instances for which both assessors agree that the concept occurs or does not occur

% agreement = ???
Predictive Analysis measures agreement: percent agreement

- **Percent agreement**: percentage of instances for which both assessors agree that the concept occurs or does not occur

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
<td>90</td>
</tr>
</tbody>
</table>

\[
\text{% agreement} = \frac{(5 + 75)}{100} = 80\%
\]
Predictive Analysis
measures agreement: percent agreement

- **Problem:** percent agreement does not account for agreement due to random chance.
- **How can we compute the expected agreement due to random chance?**
  - **Option 1:** assume unbiased assessors
  - **Option 2:** assume biased assessors
Predictive Analysis
kappa agreement: chance-corrected % agreement

• **Option 1:** unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td>no</td>
<td>??</td>
<td>??</td>
</tr>
</tbody>
</table>

50 50
Predictive Analysis
kappa agreement: chance-corrected % agreement

- **Option 1**: unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

50 50

50 50
Predictive Analysis
kappa agreement: chance-corrected % agreement

• **Option 1: unbiased assessors**

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

random chance % agreement = ???
Predictive Analysis
kappa agreement: chance-corrected % agreement

• Option 1: unbiased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>no</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

random chance % agreement = \( \frac{25 + 25}{100} \) = 50%
Predictive Analysis
kappa agreement: chance-corrected % agreement

• **Kappa agreement**: percent agreement after correcting for the expected agreement due to random chance

\[ \kappa = \frac{P(a) - P(e)}{1 - P(e)} \]

• \( P(a) \) = percent of observed agreement
• \( P(e) \) = percent of agreement due to random chance
Predictive Analysis

kappa agreement: chance-corrected % agreement

- **Kappa agreement:** percent agreement after correcting for the expected agreement due to unbiased chance

\[
P(a) = \frac{5 + 75}{100} = 0.80
\]
\[
P(e) = \frac{25 + 25}{100} = 0.50
\]

\[
\kappa = \frac{P(a) - P(e)}{1 - P(e)} = \frac{0.80 - 0.50}{1 - 0.50} = 0.60
\]
Predictive Analysis
kappa agreement: chance-corrected % agreement

• Option 2: biased assessors

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no</td>
<td>15</td>
<td>75</td>
</tr>
</tbody>
</table>

biased chance % agreement = ???
Predictive Analysis
kappa agreement: chance-corrected % agreement

• Kappa agreement: percent agreement after correcting for the expected agreement due to biased chance

\[
P(a) = \frac{5 + 75}{100} = 0.80 \quad P(e) = \left( \frac{10}{100} \times \frac{20}{100} \right) + \left( \frac{90}{100} \times \frac{80}{100} \right) = 0.74
\]

\[
K = \frac{P(a) - P(e)}{1 - P(e)} = \frac{0.80 - 0.74}{1 - 0.74} = 0.23
\]
Predictive Analysis
data annotation process

• **INPUT**: unlabeled data, annotators, coding manual

• **OUTPUT**: labeled data

1. using the latest coding manual, have all annotators label some previously unseen portion of the data (~10%)

2. measure inter-annotator agreement (Kappa)

3. **IF** agreement < X, **THEN**:
   - refine coding manual using disagreements to resolve inconsistencies and clarify definitions
   - return to 1

**ELSE**

- have annotators label the remainder of the data
Predictive Analysis
data annotation process

• What is good (Kappa) agreement?
• It depends on who you ask
• According to Landis and Koch, 1977:
  ‣ 0.81 - 1.00: almost perfect
  ‣ 0.61 - 0.70: substantial
  ‣ 0.41 - 0.60: moderate
  ‣ 0.21 - 0.40: fair
  ‣ 0.00 - 0.20: slight
  ‣ < 0.00: no agreement
Predictive Analysis

questions

• Is a particular concept appropriate for predictive analysis?
• What should the unit of analysis be?
• How should I divide the data into training and test sets?
• What is a good feature representation for this task?
• What type of learning algorithm should I use?
Predictive Analysis
turning data into (training and test) instances

• For many text-mining applications, turning the data into instances for training and testing is fairly straightforward

• **Easy case:** instances are self-contained, independent units of analysis
  - **topic categorization:** instances = documents
  - **opinion mining:** instances = product reviews
  - **bias detection:** instances = political blog posts
  - **emotion detection:** instances = support group posts
## Topic Categorization
predicting health-related documents

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td></td>
</tr>
<tr>
<td>w_2</td>
<td></td>
</tr>
<tr>
<td>w_3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>w_n</td>
<td></td>
</tr>
</tbody>
</table>

| instances | | label |
|-----------|-----------------|
|           | health          |
|           | other           |
|           | other           |
|           | other           |
|           | other           |
|           | other           |
|           | other           |

| instances | | label |
|-----------|-----------------|
|           | health          |
Opinion Mining
predicting positive/negative movie reviews

<table>
<thead>
<tr>
<th>instances</th>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>w_2</td>
<td>w_3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Bias Detection
predicting liberal/conservative blog posts

<table>
<thead>
<tr>
<th>features</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>w_2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Predictive Analysis
turning data into (training and test) instances

• A not-so-easy case: relational data
• The concept to be learned is a relation between sets of objects
• May require features that characterize properties of the set
• May require ML algorithms that do not make independent predictions
Predictive Analysis

turning data into (training and test) instances

• Example of relational data in text-mining:
  ‣ topic segmentation: segmenting discourse into topically coherent chunks
Predictive Analysis

topic segmentation example
Predictive Analysis

topic segmentation example: instances
Predictive Analysis

topic segmentation example: independent instances?
Predictive Analysis

topic segmentation example: independent instances?
Predictive Analysis
discourse analysis in MOOCs: independent instances?

- **Question**: requests information about the course content
- **Answer**: contributes information in response to a question
- **Issue**: expresses a problem with the course management
- **Issue Resolution**: attempts to resolve a previously raised issue
- **Positive Ack**: positive sentiment about a previous post
- **Negative Ack**: negative sentiment about a previous post
- **Other**: serves a different purpose
Predictive Analysis
turning data into (training and test) instances

- There are learning algorithms that incorporate relational constraints between predictions
- However, they are beyond the scope of this class
- We’ll be covering algorithms that make independent predictions on instances
- That said, many algorithms output prediction confidence values
- Heuristics can be used to favor certain types of joint outcomes more than others
Predictive Analysis
questions
We want our model to “learn” to recognize a concept
So, what does it mean to learn?
• The machine learning definition of “learning:”

A machine “learns” with respect to a particular task T, performance metric P, and experience E, if the system improves its performance P at task T (on new data) following experience E. -- Tom Mitchell
We want our model to improve its generalization performance!

That is, its performance on previously unseen data!

Generalize: to derive or induce a general conception or principle from particulars. -- Merriam-Webster

In order to test generalization performance, the training and test data cannot be the same.

Why?
Training data + Representation
what could possibly go wrong?
• While we don’t want to test on training data, models usually perform the best when the training and test set are derived from the same “probability distribution”.

• What does that mean?
Predictive Analysis
training and test data

Data

positive instances

negative instances

Training Data

Test Data
Predictive Analysis
training and test data

- Is this a good partitioning? Why or why not?

Data

Training Data

Test Data

positive instances

negative instances
Predictive Analysis
training and test data

Data

positive instances

negative instances

Training Data

Random Sample

Test Data

Random Sample
Predictive Analysis
training and test data

- Usually, random sampling should produce comparable (but not equal) data for training and testing.

Data

Training Data

Test Data

positive instances

negative instances
Models usually perform the best when the training and test set have:

- a similar proportion of positive and negative examples
- a similar co-occurrence of feature-values and each target class value
Predictive Analysis
training and test data

- **Caution:** in some situations, partitioning the data randomly might inflate performance in an unrealistic way!

- How the data is split into training and test sets determines what we can claim about generalization performance

- The appropriate split between training and test sets is usually determined on a case-by-case basis
• Suppose we want to train an email spam classifier

• Obviously, we want it to generalize to new emails (i.e., not in the training set)

• But, what are some other “things” we might want to classifier to generalize beyond?
Predictive Analysis
discussion

• **Spam detection:** should the training and test sets contain email messages from the *same sender, same recipient, and/or same timeframe*?

• **Topic segmentation:** should the training and test sets contain potential boundaries from the *same discourse*?

• **Opinion mining for movie reviews:** should the training and test sets contain reviews for the *same movie*?

• **Sentiment analysis:** should the training and test sets contain blog posts from the *same discussion thread*?
Predictive Analysis
questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?
Predictive Analysis
three types of classifiers

• Linear classifiers
• Decision tree classifiers
• Instance-based classifiers
Predictive Analysis

three types of classifiers

- All types of classifiers learn to make predictions based on the input feature values
- However, different types of classifiers combine the input feature values in different ways
- Chapter 3 in the book refers to a trained model as knowledge representation
Predictive Analysis

linear classifiers: perceptron algorithm

\[ y = \begin{cases} 
1 & \text{if } w_0 + \sum_{j=1}^{n} w_j x_j > 0 \\
0 & \text{otherwise} 
\end{cases} \]
Predictive Analysis
linear classifiers: perceptron algorithm

\[ y = \begin{cases} 
1 & \text{if } w_0 + \sum_{j=1}^{n} w_j x_j > 0 \\
0 & \text{otherwise} 
\end{cases} \]

parameters learned by the model
predicted value (e.g., 1 = positive, 0 = negative)
Predictive Analysis
linear classifiers: perceptron algorithm

test instance

<table>
<thead>
<tr>
<th></th>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

model weights

<table>
<thead>
<tr>
<th></th>
<th>w_0</th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

output = $2 + (0.50 \times -5) + (1.0 \times 2) + (0.2 \times 1)$

output = 1.7

output prediction = positive
Predictive Analysis
linear classifiers: perceptron algorithm

test instance

<table>
<thead>
<tr>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

model weights

<table>
<thead>
<tr>
<th>w_0</th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

output = 2 + (0.50 \times -5) + (1.0 \times 2) + (0.2 \times 1)

output = 1.7

According to this model, f_1 has an inverse relation with “positive”

Prediction = positive
Predictive Analysis
linear classifiers: perceptron algorithm

Test instance

<table>
<thead>
<tr>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Model weights

<table>
<thead>
<tr>
<th>w_0</th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Output = 2 + (0.50 \times -5) + (1.0 \times 2) + (0.2 \times 1)

Output = 1.7

Output prediction = positive

According to this model, f_2 has a positive relation with “positive”
Predictive Analysis
linear classifiers: perceptron algorithm

<table>
<thead>
<tr>
<th>test instance</th>
<th>model weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_1</td>
<td>f_2</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

output = 2 + (0.50 \times -5) + (1.0 \times 2) + (0.2 \times 1)

output = 1.7

output prediction = positive

According to this model, f_3 has a positive, but weaker, relation with “positive”
Predictive Analysis
linear classifiers: perceptron algorithm

2.0 - 0.5PETAL-LENGTH - 0.8PETAL-WIDTH = 0

(two-feature example borrowed from Witten *et al.* textbook)
Predictive Analysis
linear classifiers: perceptron algorithm

Predictive Analysis
linear classifiers: perceptron algorithm

Would a linear classifier do well on positive (black) and negative (white) data that looks like this?
- Draw a decision tree that would perform perfectly on this training data!
predictive analysis
instance-based classifiers

- predict the class associated with the most similar training examples
Predictive Analysis
instance-based classifiers

- predict the class associated with the most similar training examples
Predictive Analysis
instance-based classifiers

• **Assumption:** instances with similar feature values should have a similar label

• Given a test instance, predict the label associated with its nearest neighbors

• There are many different similarity metrics for computing distance between training/test instances

• There are many ways of combining labels from multiple training instances
Predictive Analysis
questions

- Is a particular concept appropriate for predictive analysis?
- What should the unit of analysis be?
- How should I divide the data into training and test sets?
- What is a good feature representation for this task?
- What type of learning algorithm should I use?