Machine Learning: training and classification

What’s Machine Learning?

- **Machine**: any mechanical device
- **Learning**: changing behaviour, on the basis of previous experience, to better perform in the future
- **Data mining**: going from data (registration of facts) to information (relevant patterns underlying the data)
- **Data mining (application) vs. machine learning (algorithm)**

Main references

What is Machine Learning?

- Learning Problem example:
  - $T =$ apples/pears classification task
  - $E =$ samples of apples and pears
  - $P =$ accuracy (number of apples correctly recognized over the whole set)

- A computer program is said to learn from experience $E$ with respect to some task $T$ and some performance measure $P$, if its performance on $T$, as measured by $P$, improves with experience $E$.

  (Tom Mitchell, 1998)

How does Machine Learning work?

- Supervised learning
  Right answers provided (e.g. classification problem: predict the correct class of an item... «spam» or «not spam»?)

- Unsupervised learning
  No answers provided: guess the best possible association / pattern

Examples of supervised learning:

- Regression problem (predict continuous valued output)

Goal: continuous values predictions on the basis of discrete examples

Examples of supervised learning:

- Classification problem (predict discrete class association)

![Graph](image-url)
How does Machine Learning work?

- **Learning methods:**
  - **Supervised learning:**
    - **Classification problem** (predict discrete class association combining more features)

- **Other examples of supervised learning:**
  - **Forecast prediction**
    - **Learning data:** series of classified (time and location) atmospheric observations
      (temperature, pressure, humidity, wind, solar radiation...)
    - **Task:** predicting atmospheric states (set of continuous variables) in specific locations at specific times
  - **Face recognition**
    - **Learning data:** series of classified (named) pre-elaborated pictures
    - **Task:** classification task (given a finite set of classes, which one(s) better represents a new picture?)

- **SOME EXAMPLE of Machine Learning**
  - **Machine Translation**
    - **Learning data:** multilingual, aligned corpora
    - **Task:** predicting the best association pattern of words in a source language with another pattern of words in the target language
  - **Diagnosis / best care**
    - **Learning data:** series of symptoms and their gravity associated to illness / machine fault
    - **Task:** predicting a previous state/trend (configuration of continuous variables in time) that caused a precise state

- **Diagnosis: Play Forecast**
  - **Learning data:**
    - **Game:**
      - **outlook:** sunny, overcast, rainy
      - **temperature:** 85, 80, 70
      - **humidity:** 85, 90, 96
      - **windy:** TRUE, FALSE
      - **play:** no, yes
  - **Task:** predicting the outcome of a precise state
**Some Example of Data Set**

- **Play Forecast:** guess the game!

<table>
<thead>
<tr>
<th>outlook</th>
<th>temperature</th>
<th>humidity</th>
<th>windy</th>
<th>play</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Authorship Detection:** guess the author!

<table>
<thead>
<tr>
<th>the</th>
<th>a</th>
<th>who</th>
<th>that</th>
<th>for</th>
<th>fuck</th>
<th>peerless</th>
<th>...</th>
<th>author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1454</td>
<td>765</td>
<td>232</td>
<td>313</td>
<td>32</td>
<td>3</td>
<td>0</td>
<td>...</td>
<td>Melville</td>
</tr>
<tr>
<td>1278</td>
<td>533</td>
<td>327</td>
<td>324</td>
<td>54</td>
<td>0</td>
<td>6</td>
<td>...</td>
<td>Shakespeare</td>
</tr>
<tr>
<td>1087</td>
<td>609</td>
<td>132</td>
<td>122</td>
<td>34</td>
<td>321</td>
<td>2</td>
<td>...</td>
<td>Eminem</td>
</tr>
</tbody>
</table>

**Classification of Primary Progressive Aphasic Patients**

- **What’s PPA**
  - Primary Progressive Aphasia (PPA) is a specific, progressive disorder of language associated with atrophy of the frontal and temporal regions of the left hemisphere (FTLD) (Mesulam 1982, Mesulam & Weintraub 1992).

- **PPA Classification**
  - 3 main variants (Gorno-Tempini et al. 2011):
    - non-fluent / agrammatic (gPPA)
    - semantic (sPPA)
    - logopenic / fonologic (lPPA)

**Clinical Diagnosis of Non-Fluent/Agrammatic Variant PPA**

- At least one of the following core features must be present:
  - Agrammatism in language production
  - Effortful, halting speech with inconsistent speech sound errors and distortions (apraxia of speech)

- At least 2 of the following other features must be present:
  - Impaired comprehension of syntactically complex sentences
  - Spared single-word comprehension
  - Spared object knowledge
Clinical diagnosis of semantic variant PPA

- Both of the following core features must be present:
  - Impaired confrontation naming
  - Impaired single-word comprehension

- At least 3 of the following other diagnostic features must be present:
  - Impaired object knowledge, particularly for low-frequency or low-familiarity items
  - Surface dyslexia or dysgraphia
  - Spared repetition
  - Spared speech production (grammar and motor speech)

Clinical diagnosis of logopenic variant PPA

- Both of the following core features must be present:
  - Impaired single-word retrieval in spontaneous speech and naming
  - Impaired repetition of sentences and phrases

- At least 3 of the following other features must be present:
  - Speech (phonologic) errors in spontaneous speech and naming
  - Spared single-word comprehension and object knowledge
  - Spared motor speech
  - Absence of frank agrammatism

Specific patterns of neuroanatomical damage

- Non-fluent / agrammatic (gPPA) left posterior frontal and insular regions (Grossman et al. 1996, Gorno-Tempini et al. 2004, Nestor et al. 2003)

- Semantic (sPPA) anterior temporal region (Hodges et al. 1996, Mummery et al. 2000)

- Logopenic (lPPA) left temporo-parietal regions (Gorno-Tempini et al. 2004)

Diagnosis protocol and classification

- Minimal assessment
  - Quantitative/qualitative analysis of extended production
  - Naming and comprehension of single words
  - Sentence repetition (AAT, Token Test)
  - Comprehension of syntactically complex sentences (BADA)
Diagnosis protocol and classification

- Very good predictor: **speech analysis** (done by a trained speech pathologist)
  - Both spontaneous and connected speech
  - E.g., picture description task: *picnic* (Western Aphasia Battery, Kertesz 1982)

- Cons:
  - Complex analysis
  - Manual transcription of speech
  - Manual tagging of the transcription (content vs. functional, high vs. low frequency words...)
  - Manual counts (speech rate, # of content words, Type/Token ratio...)

Computational approaches to classification

- Lots of data available for Machine Learning approaches ("there is no data like more data", Jelinek 2004)

- Free software for (semi)automatic analysis of speech and transcriptions:
  - Praat: phonetic analysis [http://www.fon.hum.uva.nl/praat/]
  - TextStat: corpus analysis [http://neon.niederlandistik.fu-berlin.de/textstat/]
  - R: statistics [http://www.r-project.org/]

Computational approaches to classification: some advantages

- Push the automatic analysis as far as possible (more reliable, objective, and robust)

- Faster cues extraction for diagnosis

- Smaller tests (just few seconds of connected speech might be sufficient to extract highly relevant features)

Preparation of a classifier vectorial representation

- What's a vector
  - N components/dimensions:
    - e.g. two-dimensional vector: \( \mathbf{v} = <5, 3> \)

- What's for
  - It represents the transcriptions in a measurable space, this makes comparisons possible
Preparation of a classifier
Simple semantic components

- **Purely statistical approach (bag-of-words):**
  - Classifiers created using simple word frequencies: in the transcription T, the token “the” occurs 123 times, “house” 12 times, “big” 5 times...
  - Supervised Classification task based on word frequency extracted from connected speech (picnic description task) (Garrard et al. 2010): accuracy around 98% (Semantic Dementia (SD) vs. controls matched by age)
  - Supervised Classification (Naïve Bayes) using low and high frequency content word (Garrard et al. 2014): accuracy around 88% (right vs. left hemisphere atrophy pattern)

How a vector representing a transcription looks like
- **ARFF - Attribute-Relation File Format, WEKA tool, Bouckaert et al. 2011:**
  - @relation classification_test
  - @attribute token_A integer
  - @attribute token_B integer
  - @attribute class {gPPA, sPPA, lPPA, NC}
  - @data
    - 0,1,5,3,0,3,3,0,0,1,3,0,0, … , gPPA
    - 5,6,0,0,0,0,2,0,0,5,12,0,0, … , NC
    - ...

Selecting the «best» features

- **Information Gain (IG) - (Mitchell 1997):**
  - Measure of the information value of using a specific feature value instead of not using it:
    - Total information needed without that feature – Information to encode such feature
  - $\text{Info}(p_1, p_2) = \text{entropy}(p_1, p_2) = -p_1 \log p_1 - p_2 \log p_2$
    - e.g. $\text{Info}([4,1]) = \text{entropy}(4/5, 1/5)$
    - $= -4/5 \log 4/5 - 1/5 \log 1/5$
    - $= 0.32 * 0.8 - 0.2 * 2.32$
    - $= 0.256 + 0.464 = 0.72$
  - $\text{IG} = \text{info}(\text{total}) - \text{info}(\text{attributes})$
Selecting the «best» features

- Information Gain (IG) - (Mitchell, 1997)

<table>
<thead>
<tr>
<th>Value</th>
<th>Information Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>info(0,3) = 0.12</td>
</tr>
<tr>
<td>1</td>
<td>info(1,4) = 0.12</td>
</tr>
<tr>
<td>&gt;2</td>
<td>info(2,0) = 0.37</td>
</tr>
</tbody>
</table>

info_tot = 0.04

IG = info(9,4) - info(0,3) - info(1,4)

= 0.89 - 0.04

= 0.85

Choosing the «best» training algorithm

- Naive Bayes
  - Gaussian: when working with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution (no discretization process needed).
  - Multinomial: Suppose n₁, n₂, ..., nᵢ is the number of times word i occurs in the document D, and Pᵣ, Pᵢ, ..., Pᵢ is the probability of obtaining word i when sampling from all the documents in category C. Assume that the probability is independent of the word’s context and position in the document. These assumptions lead to a multinomial distribution for document probabilities. For this distribution, the probability P of a document D given its class C (i.e. P(D|C))

  \[ P(D|C) = \frac{n!}{n_1!n_2!...n_k!} \prod_{i=1}^{k} \frac{n_i}{N} \]

  where \( N = n_1 + n_2 + ... + n_k \) is the number of words in D.

- Logistic Regression
  - a discriminative classifier; better than linear regression (C is a linear combination of the attributes, with predetermined weights: \( x = w_0 + w_1a_1 + ... + w_ka_k \)), since it builds a linear model based on a transformed target variable;
  - it can model proper probabilities (values range: 0 to 1);
  - No need to assume that the errors are statistically independent and normally distributed with the same standard deviation (Bernoulli distribution is assumed).
Choosing the «best» training algorithm

- **Support Vector Machines (SVMs)**
  - another (very popular) type of linear discriminative classifier;
  - **maximum margin classifiers**: decision boundary between two classes that maximizes the margin between the two classes (Manning et al., 2008).

Classifier validation

- **Training data, Validation data, and Test data.**
  - The **training** data is used for learning schemes generating the classifiers
  - The **validation** data is used to optimize parameters of those classifiers
  - The **test** data (usually 1/3 of the total data) is used to calculate the error rate of the final, optimized, method

Cross validation

- **most practical in limited-data situations** (our case)
- **Safeguard against uneven representations in training and testing data sets**: splitting the data several times with different random samples:
  - decide a fixed number N of partitions (folds) of the data. Split the data into N approximately equal partitions; each in turn is used for testing and the remainder is used for training (N-fold cross validation).

An Italian study

- **Goal**: test the reliability of different classifiers, using ML techniques, using transcriptions of connected speech (picnic picture, Western Aphasia Battery, Kertesz 1982)
An Italian study
the clinical population

13 PPA subjects + 6 controls (matched by age)

<table>
<thead>
<tr>
<th>Subject</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>66</td>
<td>52</td>
<td>81</td>
<td>73</td>
<td>64</td>
<td>83</td>
<td>70</td>
<td>70</td>
<td>69</td>
<td>72</td>
<td>74</td>
<td>73</td>
<td>62</td>
</tr>
<tr>
<td>WPS</td>
<td>0.9</td>
<td>1.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.5</td>
<td>1.1</td>
<td>1.1</td>
<td>0.9</td>
<td>1.2</td>
<td>1.5</td>
<td>0.7</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Nouns</td>
<td>29</td>
<td>31</td>
<td>24</td>
<td>8</td>
<td>14</td>
<td>21</td>
<td>12</td>
<td>15</td>
<td>16</td>
<td>21</td>
<td>19</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>MF</td>
<td>81.3</td>
<td>208.8</td>
<td>126.6</td>
<td>30.50</td>
<td>104.8</td>
<td>70.7</td>
<td>111.9</td>
<td>78.5</td>
<td>28.3</td>
<td>432.3</td>
<td>41.1</td>
<td>134.8</td>
<td>171.6</td>
</tr>
</tbody>
</table>

words per second (WPS); number of nouns produced (nouns); mean nouns frequency (MF).

(Semi)automatic pre-processing (similar conventions are discussed in Thompson et al. 2012 and Wilson et al. 2010)

Original transcription

<table>
<thead>
<tr>
<th>Time</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>. io vedo . un picnic con due persone che stanno preparando da mangiare una legge</td>
</tr>
<tr>
<td>2</td>
<td>una beve sta vuotando la bevanda poi [HE] cane vicino un ragazzo con aquilone</td>
</tr>
</tbody>
</table>

Normalized version

<table>
<thead>
<tr>
<th>Time</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>. io vedo . un picnic con due persone che stanno preparando da mangiare una legge</td>
</tr>
<tr>
<td>2</td>
<td>una beve sta vuotando la bevanda poi [HE] cane vicino un ragazzo con aquilone</td>
</tr>
</tbody>
</table>

Speech Transcription samples

<table>
<thead>
<tr>
<th>gPPA</th>
<th>sPPA</th>
<th>lPPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>è [eh] un picnic sul lago</td>
<td>cosa devo dire</td>
<td>allora in un in un posto di campagna</td>
</tr>
<tr>
<td>c’ è una barca a vela</td>
<td>[eh] bhe ci sono d</td>
<td>sembra che abbiamo fatto</td>
</tr>
<tr>
<td>e un uomo che</td>
<td>questo posto</td>
<td>ciò sono presenti alcune persone</td>
</tr>
<tr>
<td>c’ è un uomo che [eh] va</td>
<td>è vicino [FS] al lago</td>
<td>che</td>
</tr>
<tr>
<td>sta volando un</td>
<td>questo posto è vicino al lago</td>
<td>hanno fatto</td>
</tr>
<tr>
<td>un aqualone</td>
<td>allora ci sono anche i due che</td>
<td>ci sono due [FS] due persone che</td>
</tr>
</tbody>
</table>

Vocabulary extraction and token frequency (.CSV) [TextStat]

token, frequency

| un  | 75  |
| che | 60  |
| HE  | 53  |
| cane | 11 |
| cosa | 11 |
An Italian study

**Vectors preparation**

- **Language Model (n-grams)** (ARPA file) [CMU-CSLM toolkit]
  - H-log bigram back-off probability
  - -0.3626 <s> allora 0.1736
  - -1.2765 <s> un 0.0000
  - -0.2359 la casa 0.1011

**Phonetic transcription and POS tagging**

- **Automatic transcription and POS annotation** [G2P + IT lex MaxEnt models]
  - vedo un picnic con due persone...
  - (I) see a picnic with two people...
  - <w v="vedo" p="v e 1 d o" pos="V" />
  - <w v="un" p="u 1 n" pos="D" />
  - <w v="picnic" p="p i k - n i 1 k" pos="N" />
  - <w v="con" p="k o 1 n" pos="P" />
  - <w v="due" p="d u 1 - e" pos="Q" />
  - <w v="persone" p="p e r - s o 1 - n e" pos="N" />

**POS tagging and syntactic critical aspects**

- **Inefficient probabilistic POS annotation**
  - porteranno (they/he) will bring
  - porta door/brings
  - V-fut-3P NN V-pres-3S
  - (MaxEnt: wrong)

- **Shallow parsers tend to incorrectly cluster constituents together:**
  - Italian is a null-subject language; it admits post-verbal subjects;
  - Subject/Object Relative Clauses, Passives, clitic forms are hard to be processed correctly but they are crucial constructions

**Special syntactic features**

- **syntag_breaks**
  - counting the hesitation after a functional word:
    - e.g. D(eterminer) [EH]
    - C(omplementizer) [EH]

- **phon_bilabial_rep, phon_fricative_rep**
  - counting simple duplicated mono-phonetic/syllabic patterns:
    - e.g. m m (m)
    - la la (la)

- **false_starts**
  - [FS] before content words (especially nouns)
**An Italian study**

**Classifiers Training**

- **Information Gain, IG** (number of initial attributes: 450)
- Ranked attributes:
  - 1 449 phon_bilabial_rep (sillabic repetition frequencies)
  - 1 448 syntag_breaks (pauses after specific syntactic items)
  - 0.876 79 persone
  - 0.68 23 molo
  - 0.68 12 sta

- Only high IG attributes should be included in the classifiers (for avoiding overfitting)

**Classifiers Evaluation**

- **Supervised training**
  - Algorithm used: Naive Bayes Multinomial (10-fold cross-validation)

- **Discrimination accuracy among classes**
  - (normal, non-fluent - gPPA, semantic - sPPA, logopenic - lPPA):
    - Correctly Classified Instances 17: 92.3077%
    - Incorrectly Classified Instances 1: 7.6923%
    - Kappa statistic 0.8632
    - Mean absolute error 0.076
    - Root mean squared error 0.2125

**Discussion of the preliminary result**

- The automatic classification, using 40 features (among which syntactic «breaks», syllabic cluster repetition, content word e functional words with high IG) shows high reliability (accurate classification above 92%)

- The confusion matrix clearly shows the classifiers difficulties with lPPA (as in the clinical reality):

<table>
<thead>
<tr>
<th></th>
<th>gPPA</th>
<th>sPPA</th>
<th>lPPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>gPPA</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sPPA</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>lPPA</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Extending the population**

- (There is no data like) more data!
- Corpus integration (23 patients, 13 controls):
  - 10 non-fluents/agrammatics (gPPA)
  - 6 semantics (sPPA)
  - 7 logopenics (lPPA)
  - 13 controls (matched by age) (NC)
An Italian study
Classifiers Evaluation

- Supervised training on just old data (algorithm used: Naive Bayes Multinomial, 10-fold cross-validation)
- Discrimination accuracy among classes using simple word frequency on selected (high IG) words (normal, non-fluent - gPPA, semantic - sPPA, logopenic - lPPA):

| Correctly Classified Instances | 25 | 69.4444 % |
| Incorrectly Classified Instances | 11 | 30.5556 % |
| Kappa statistic | 0.5667 |
| Mean absolute error | 0.1597 |
| Root mean squared error | 0.3842 |

The automatic classification, using 43 components (just content word e functional words with high IG) shows medium reliability (accuracy in classification \( \approx 69\% \)).

The confusion matrix clearly shows the classifiers difficulties both with gPPA and with lPPA:

<table>
<thead>
<tr>
<th></th>
<th>gPPA</th>
<th>sPPA</th>
<th>lPPA</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>gPPA</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>sPPA</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>lPPA</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NC</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

An Italian study
More “complexity measures”

- New features to be used (Fraser et al. 2014)
  - Vocabulary/production richness
    - # Words, # Sentences, Type/Token ratio, Word familiarity...
  - Tree-based metrics (excellent predictors of age-related cognitive decline, Cheung and Kemper 1992, and mild cognitive impairment, Roark et al 2011): Syntactic depth measurements (Yngve 1960), non-local dependency distance (FRC, Chesi 2014), # coordinations, # complex NPs, # (reduced) long/short passives, post-verbal subjects, properly bound clitic objects, reflexives

More on Tree-based metrics

- Syntactic depth measurements (Yngve 1960)
  - sensitive to left vs. right branching structure (e.g. the juice that the child spilled stained the rug max depth: 3, mean depth: 1.67, total depth: 15 vs the child spilled the juice that stained the rug max depth: 2, mean depth: 1.11, total depth: 10)
An Italian study

More “complexity measures”

- More on Tree-based metrics
  - Syntactic depth measurements (Yngve 1960)
  - Non-local dependency distance (FRC, Chesi 2014)
  - Feature-based, memory Retrieval Cost function (more shared features in memory, more complex the retrieval):
    \[
    CFRC(x) = \prod_{i=1}^{n} \frac{1 + \alpha x_i}{1 + \beta x_i}
    \]
  - Syntactic depth (Yngve 1960)
  - Non-local dependency distance (FRC, Chesi 2014)

- #reduced) long/short passives RC
  - e.g. the juice (reduced that was) spilled (short by the boy)

- How hard is automatic extraction of tree-based features?
  - Pretty hard (so far!)
  - There are no Italian parsers dealing with non-local constituency-based dependencies (e.g. relative clauses, pronominal binding)
  - Post-verbal subjects are problematic for any parser

- Accuracy of NLP tools used in English:
  - gPPA (Fraser et al. 2014) 87%
  - sPPA 89.2%
  - lPPA 91.4%

An Italian study

Classifiers Evaluation

- Supervised training on new data
  - (algorithm used: Naive Bayes Multinomial, 10-fold cross-validation)
- Discrimination accuracy among classes
  - (normal, non-fluent - gPPA, semantic - sPPA, logopenic - lPPA):
    - Correctly Classified Instances: 33  91.6666%
    - Incorrectly Classified Instances: 3  8.33333%
    - Kappa statistic: 0.8542
    - Mean absolute error: 0.077
    - Root mean squared error: 0.2125

An Italian study

Discussion

- The automatic classification, using 54 features (word frequencies + phonological cues + tree-based metrics with high IG) shows good reliability
  - (accuracy in classification > 91%)
- The confusion matrix again shows some difficulties with gPPA and with lPPA:

<table>
<thead>
<tr>
<th></th>
<th>gPPA</th>
<th>sPPA</th>
<th>lPPA</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>gPPA</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>sPPA</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>lPPA</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>NC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
</tbody>
</table>
Future research

Future research
direct analysis of the recordings

- Preliminary study by Alessandra Cervone (IUSS Diploma thesis 2014, supervisor: Cristiano Chesi, co-supervisor Stefano Cappa)
- Recorded audio files
  - 7 (44%) non-fluent
  - 4 (25%) semantic
  - 4 (25%) logopenic
  - 1 (6%) mixed
- Poor recording quality (low sample rate, compressed audio)

Future research
Acoustic indicators of Speaking rate

<table>
<thead>
<tr>
<th>Acoustic variables for SPEAKING RATE</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Rate</td>
<td>number of syllables / total duration of the recording</td>
</tr>
<tr>
<td>Articulation Rate</td>
<td>number of syllables / phonation time</td>
</tr>
<tr>
<td>ASD (Average Syllable Duration)</td>
<td>phonation time / number of syllables</td>
</tr>
<tr>
<td>Phonation-time ratio</td>
<td>phonation time / total duration of the recording</td>
</tr>
</tbody>
</table>

Future research
Pauses

<table>
<thead>
<tr>
<th>Acoustic variables for PAUSES</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Pauses Duration</td>
<td>total time of pauses / number of pauses</td>
</tr>
<tr>
<td>Pauses Frequency during recording time</td>
<td>number of pauses / total duration of the recording</td>
</tr>
<tr>
<td>Pauses Frequency during speaking time</td>
<td>number of pauses / phonation time</td>
</tr>
</tbody>
</table>

Future research
Vowel duration

<table>
<thead>
<tr>
<th>Acoustic variable for VOWEL DURATION</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vowel duration</td>
<td>mean vowel length (s) calculated over 15 vowels</td>
</tr>
</tbody>
</table>
Future research

Syllable nuclei detection using Praat

- **Pre-processing** (researcher speech removal)
- **Syllable nuclei** detection using Praat (script by Jong and Ton Wempe 2009)

Future research

Preliminary results

- **Speech rate** (number of syllables/total duration of the recording)
  - higher in the NC group (M=2.51 syllable/s, SD=0.37) compared to the PPAs (M=2.42 syllable/s, SD=0.79).
  - Difference not statistically significant: 
    \[ t(21.45) = 0.34, p = 0.74, CI 95\% (-0.4627, 0.6427) \]

Future research

Preliminary results

- **Articulation rate** (number of syllables/speaking time)
  - articulation rate is significantly faster (t(17.8)= 2.73, \( p = 0.01 \)) in PPA (M=3.65 syllable/s, SD=0.44) than in NC (M=2.8 syllable/s, SD=0.27). CI 95\% (-1.505, -0.1944)

Future research

Preliminary results

- **Average Syllable Duration** (phonation time / number of syllables)
  - Mean syllable duration was on average **slightly longer** in the group of NCs (M=0.36 s, SD=0.03) than in PPA patients (M=0.34 s, SD=0.03).
  - The PPA group, however, did not show a normal distribution according to the Shapiro-Wilk test performed (W= 0.6, \( p = 1.709e-05 \)), thus invalidating the results of a potential Welch’s t-test.
Future research
Preliminary results

- Phonation-time ratio
- NCs were speaking during almost 90% of the time of the recording (M=0.89, SD=0.19) while in general PPA patients spoke only about 70% of the total time (M=0.7, SD=0.08).
- According to the Welch’s t-test performed, this difference is statistically very significant (t(20.42)=2.9, p=0.008, CI 95% (0.0547, 0.33)).

In conclusion

- Machine Learning approaches based on fully automatic features extraction are not (yet) sufficient to guarantee a good classification of PPA variants (at least in Italian).
- But the linguistic features used guarantee a good classification of the 3 main PPA variants vs NC and permitted the identification of the relevant features discriminating among classes (high IG > Decision Tree).
- (Automatic) audio analysis could be pushed further (e.g. fundamental frequency – F0 variations, jitter, shimmer, energy per sentence...)

What’s Weka

- Collection of Machine Learning algorithm for data mining tasks
  - 100+ algorithms for classification
  - data pre-processing
  - Feature selection assistance
  - Clustering, finding association rules etc.

HOW do we use Weka

- Load data into Weka and explore it
- Use filters and preprocess it
- Apply classification algorithms
- Interpret the output
- Understand evaluation methods and their implication
- Understand various representations for models
- Explain how popular machine learning algorithms work
- Be aware of common pitfalls with data mining
To do

- Install Weka
- Explore the “Explorer” interface
- Explore some datasets
- Build a classifier
- Interpret the output
- Use filters
- Visualize your data set

Download from http://www.cs.waikato.ac.nz/ml/weka

The explorer
Comparing classifiers

- **Baseline first**
  - Rule > ZeroR
  - simply it chooses the most likely class
  - fast on the training set
  - if attribute non significant > lower performance than the baseline

- **Simple first**
  - Rule > OneR
  - one rule does the whole job
OVERFITTING

- Using all features for classifying is not good...
  - Select **OneR**, for classifying a dataset with branching numeric attributes
  - Change the option **MinBucketSize** from 6 to 1, run the test using the Training Set:
    - branching from «temperature» in the Weather dataset: perfect prediction, very accurate, but no generalization

More Algorithms: Naïve Bayes

- **Naïve Bayes methods**
  - **OneR** (one attribute does all the work) vs. **Naïve Bayes** (use all the attributes)
  - Two assumptions attributes are
    - equally important a priori
    - statistically independent (given the class value)
      i.e., knowing the value of one attribute says nothing about the value of another (if the class is known, especially in cases where there is not a large amount of training data, Ng & Jordan 2002)

More Algorithms: Naïve Bayes

- Probability of event H (class) given evidence E (instance)
  \[ P[H | E] = \frac{P[E | H] P[H]}{P[E]} \]
  - \( P[H] \) is a priori probability of H (the probability of event before evidence is seen)
  - \( P[H | E] \) is a posteriori probability of H (the probability of event after evidence is seen)
  - "Naïve" assumption; evidence splits into parts that are independent
    \[ P[H | E] = \frac{P[E_1 | H] P[E_2 | H] \ldots P[E_n | H] P[H]}{P[E]} \]

More Algorithms: Decision trees

- E.g. **J48** (statistical classifier) C4.5 (Quinlan) = J48
  - Top-Down recursive divide and conquer
    - Select attribute for root node (create branch for each possible attribute value)
    - Split instances into subsets (One for each branch extending from the node)
    - Repeat recursively for each branch (using only instances that reach the branch)
    - Stop if all instances have the same class
More Algorithms: Decision trees

- **Aim:** to get the smallest tree
- **Heuristic:** choose the attribute that produces the "purest" nodes (i.e., the greatest information gain)
- **Information theory:** measure information in bits
- **Which is the best attribute?**
  \[
  \text{entropy}(p_1, p_2, \ldots, p_n) = p_1 \log p_1 + p_2 \log p_2 + \ldots + p_n \log p_n
  \]

Pruning: perform worse on the test data, but generalize better
- Don't continue splitting if the nodes get very small (J48 minNumObj parameter, default value 2)
- Build full tree and then work back from the leaves, applying a statistical test at each stage (confidenceFactor parameter, default value 0.25)
- Sometimes it’s good to prune an interior node, raising the subtree beneath it up one level (subtreeRaising, default true)
- Messy … complicated … not particularly illuminating

More Algorithms: Nearest neighbor

- "Rote learning": simplest form of learning
- to classify a new instance, search training set for one that’s “most like” it
  - the instances themselves represent the “knowledge”
  - lazy learning: do nothing until you have to make predictions
- “Instance-based” learning = "nearest-neighbor" learning

k-nearest-neighbors: choose majority class among several neighbors (k of them)
- lazy>IBk (instance-based learning)
- Try k = 1, 5, 20 with 10-fold cross-validation on a rather small vs large dataset
More Algorithms: LINEAR REGRESSION

- Data sets so far: nominal and numeric attributes, but only nominal classes
- Now: numeric classes (classical statistical method from 1805)
- Calculate weights from training data
- Ok if there are more instances than attributes
  - Nominal values must be converted in 1 and 0
- **M5P** (linear regression patches at each node of a Decision Tree)

More Algorithms: classification by REGRESSION

- Two-class problem
  - Training: call the classes 0 and 1
  - Prediction: set a threshold for predicting class 0 or 1
- Multi-class problem: "multi-response linear regression"
  - Training: perform a regression for each class
    - Set output to 1 for training instances that belong to the class,
    - 0 for instances that don’t
  - Prediction: choose the class with the largest output ... or use "pairwise linear regression", which performs a regression for every pair of classes

More Algorithms: classification by REGRESSION

- Investigate two-class classification by regression
- Open file diabetes.arff
- Use the NominalToBinary attribute filter to convert to numeric
  - but first set Class: class (Nom) to No class, because attribute filters do not operate on the class value
- Choose functions>LinearRegression
- Run
- Set Output predictions option
More Algorithms: logistic REGRESSION

- Two-class problem
  - Training: call the classes 0 and 1
  - Prediction: set a threshold for predicting class 0 or 1
- Multi-class problem: “multi-response linear regression”
  - Training: perform a regression for each class
  - Set output to 1 for training instances that belong to the class,
  - 0 for instances that don’t
  - Prediction: choose the class with the largest output … or use “pairwise linear regression”, which performs a regression for every pair of classes

More Algorithms: Support Vector MAchine

- Maximum margine hyperplane among support vectors
- Linearily separable: too easy… real life is harder!
  - «Kernel trick” > different shape
  - S(quential)M(inimal)O(ptimization)
  - libSVM (external download)

More Algorithms: ensembles learning

- Bagging: resampling the training set (meta>Bagging)
- Random forests: alternative branches in decision trees – (trees>RandomForests)
- Boosting: focus on where the existing model makes errors (meta>AdaBoostM1)
- Stacking: combine results using another learner (instead of voting) (meta>Stacking)

Comparing classifiers

1 * = significantly worse
v = significantly better