Table of Contents

1. The Open Analytics Platform
2. The Text Processing Extension
3. Importing Text
4. Enrichment
5. Preprocessing
6. Transformation
7. Classification
8. Visualization
9. Clustering
10. Supplementary Workflows
Overview

KNIME Analytics Platform
What is KNIME Analytics Platform?

• A tool for data analysis, manipulation, visualization, and reporting
• Based on the graphical programming paradigm
• Provides a diverse array of extensions:
  • Text Mining
  • Network Mining
  • Cheminformatics
  • Many integrations, such as Java, R, Python, Weka, H2O, etc.
Visual KNIME Workflows

**NODES** perform tasks on data

Nodes are combined to create **WORKFLOWS**
Data Access

- **Databases**
  - MySQL, MS SQL Server, PostgreSQL
  - any JDBC (Oracle, DB2, ...)

- **Files**
  - CSV, txt
  - Excel, Word, PDF
  - SAS, SPSS
  - XML, JSON
  - PMML
  - Images, texts, networks, chem

- **Web, Cloud**
  - REST, Web services
  - Twitter, Google
Big Data

- Spark
- HDFS support
- Hive
- Impala
- Vertica
- In-database processing
Transformation

- Preprocessing
  - Row, column, matrix based
- Data blending
  - Join, concatenate, append
- Aggregation
  - Grouping, pivoting, binning
- Feature Creation and Selection
Analysis & Data Mining

- **Regression**
  - Linear, logistic

- **Classification**
  - Decision tree, ensembles, SVM, MLP, Naïve Bayes

- **Clustering**
  - k-means, DBSCAN, hierarchical

- **Validation**
  - Cross-validation, scoring, ROC

- **Deep Learning**
  - Keras, DL4J

- **External**
  - R, Python, Weka, H2O
Visualization

- Interactive Visualizations
  - Scatter Plot, Box Plot, Line Plot
  - Networks, ROC Curve, Decision Tree
  - Adding more with each release!
- Misc
  - Tag cloud, open street map, molecules
- Script-based visualizations
  - R, Python
Deployment

- Database
- Files
  - Excel, CSV, txt
  - XML
  - PMML
  - to: local, KNIME Server, SSH-, FTP-Server
- BIRT Reporting
Over 1500 native and embedded nodes included:
Overview

• Installing KNIME Analytics Platform
• The KNIME Workspace
• The KNIME File Extensions
• The KNIME Workbench
  • Workflow editor
  • Explorer
  • Node repository
  • Node description
• Installing new features
Install KNIME Analytics Platform

- Select the KNIME version for your computer:
  - Mac, Win, or Linux and 32 / 64bit
- Download archive and extract the file, or download installer package and run it
Start KNIME Analytics Platform

- Use the shortcut created by the installer

- Or go to the installation directory and launch KNIME via the knime.exe

[Image of KNIME Analytics Platform window]
The KNIME Workspace

- The workspace is the **folder/directory** in which workflows (and potentially data files) are stored for the current KNIME session.
- Workspaces are portable (just like KNIME)
Welcome Page

Welcome to KNIME Analytics Platform!

New to KNIME? Looking for resources to get started?
- Register for emails with introductory tips here.
- Explore our Quickstart Guides.
- Check out 5 things to do after installing KNIME Analytics Platform.
- Find more hints and how-tos in the Learning Hub.
- And register for our release and event emails right here.

This page will be displayed upon startup but you can customize the content using the checkboxes at the bottom.

Updates for the following components are available:
- DYMATRIX Update Modeling Extensions
- Palladios for KNIME

Click here in order to install updates.

Where to go from here
- Create new workflow
- Learning Hub
- Browse example workflows
- Get additional nodes
- Go to my workflows
- Mount KNIME Cloud Server

Most recently used workflows
- ModelSelection_WebPortal_Part1
- ModelSelection_WebPortal_Part1
- ModelSelection_BasicWorkflow
- DataCleaning_WebPortal_v2.0
- KNIME_project2
- Sony_ETL_v2.0

Tips & Tricks
Specialist Nodes
Did you know there are a whole variety of specialist nodes available from KNIME Labs and the Community around Scripting, Image Processing, Text Processing, Internet Mining, Network Mining, Cell Biology and Genetics, and Chemistry. To access them, go to Help Menu and choose Install New... Software.

- Show intro text at next start
- Show update notifications at next start
- Show links and most recently used workflows at next start

NEW since 3.2: Workflow Coach recommends matching nodes.
The KNIME Workbench

- KNIME Explorer
- Workflow Editor
- Node Recommendations
- Node Repository
- Node Description
- Console

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
In LOCAL you can access your own workflow projects.

The Explorer toolbar on the top has a search box and buttons to:
- select the workflow displayed in the active editor
- refresh the view

The KNIME Explorer can contain 4 types of content:
- Workflows
- Workflow groups
- Data files
- Metanode templates
Creating New Workflows, Importing and Exporting

- Right-click in KNIME Explorer to create new workflow or workflow group or to import workflow
- Right-click on workflow or workflow group to export
Node Repository

- The Node Repository lists all KNIME nodes
- The search box has 2 modes
  - **Standard Search** – exact match of node name
  - **Fuzzy Search** – finds the most similar node name
- Nodes can be added by drag and drop from the Node Repository to the Workflow Editor.
• Console view prints out error and warning messages about what is going on under the hood.

• Click on View and select Other... to add different views
  – Node Monitor, Licenses, etc.
The Node Description window gives information about:

- Node Functionality
- Input & Output
- Node Settings
- Ports
- References to literature
Workflow Coach

Recommendation engine

- Gives hints about which node use next in the workflow
- Based on KNIME communities' usage statistics
- Based on own KNIME workflows
Tool Bar

The buttons in the toolbar can be used for the active workflow. The most important buttons:

- Execute selected and executable nodes (F7)
- Execute all executable nodes
- Execute selected nodes and open first view
- Cancel all selected, running nodes (F9)
- Cancel all running nodes
KNIME File Extensions

• Dedicated file extensions for Workflows and Workflow groups associated with KNIME Analytics Platform

• *knwf for KNIME Workflow Files

• *knar for KNIME Archive Files
More on Nodes...

A node can have 3 states:

Not Configured:
The node is waiting for configuration or incoming data.

Configured:
The node has been configured correctly, and can be executed.

Executed:
The node has been successfully executed. Results may be viewed and used in downstream nodes.
Inserting and Connecting Nodes

- Insert nodes into workspace by dragging them from Node Repository or by double-clicking in Node Repository.
- Connect nodes by left-clicking output port of Node A and dragging the cursor to (matching) input port of Node B.
- Common port types:
Node Configuration

- Most nodes require configuration
- To access a node configuration window:
  - Double-click the node
  - Right-click > Configure
Node Execution

- Right-click node
- Select Execute in context menu
- If execution is successful, status shows green light
- If execution encounters errors, status shows red light
Node Views

- Right-click node
- Select Views in context menu
- Select output port to inspect execution results
Getting Started: KNIME Example Server

- Public repository with large selection of example workflows for many, many applications
- Connect via KNIME Explorer
Curved Connections!

Workflow Editor Settings

Modify the settings for the active workflow editor. All settings will be stored with the workflow. To change default settings for new workflow editors go to the preference page. Snap to grid behavior can be toggled by pressing 'Ctrl-Space'.

- **Enable Grid**
  - **Snap to grid** (Alt-key disables snapping temporarily while moving nodes)
  - **Show grid lines**

**Grid Size**

- **Horizontal spacing (px)**: 20
- **Vertical spacing (px)**: 20

**Node Connections**

- **Curved connections**
  - **Connection line width**: 1

---

Copyright © 2018 KNIME AG

Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license

https://creativecommons.org/licenses/by-nc-sa/4.0/

Open for Innovation
Online Node Guide

- Workflows from Example Server also available online
  - https://www.knime.com/nodeguide
## Hot Keys (for future reference)

<table>
<thead>
<tr>
<th>Task</th>
<th>Hot key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node Configuration</strong></td>
<td>F6</td>
<td>opens the configuration window of the selected node</td>
</tr>
<tr>
<td><strong>Node Execution</strong></td>
<td>F7</td>
<td>executes selected configured nodes</td>
</tr>
<tr>
<td></td>
<td>Shift + F7</td>
<td>executes all configured nodes</td>
</tr>
<tr>
<td></td>
<td>Shift + F10</td>
<td>executes all configured nodes and opens all views</td>
</tr>
<tr>
<td></td>
<td>F9</td>
<td>cancels selected running nodes</td>
</tr>
<tr>
<td></td>
<td>Shift + F9</td>
<td>cancels all running nodes</td>
</tr>
<tr>
<td><strong>Move Nodes and Annotations</strong></td>
<td>Ctrl + Shift + Arrow</td>
<td>moves the selected node in the arrow direction</td>
</tr>
<tr>
<td></td>
<td>Ctrl + Shift + PgUp/PgDown</td>
<td>moves the selected annotation in the front or in the back of all overlapping annotations</td>
</tr>
<tr>
<td><strong>Workflow Operations</strong></td>
<td>F8</td>
<td>resets selected nodes</td>
</tr>
<tr>
<td></td>
<td>Ctrl + S</td>
<td>saves the workflow</td>
</tr>
<tr>
<td></td>
<td>Ctrl + Shift + S</td>
<td>saves all open workflows</td>
</tr>
<tr>
<td></td>
<td>Ctrl + Shift + W</td>
<td>closes all open workflows</td>
</tr>
<tr>
<td><strong>Meta-node</strong></td>
<td>Shift + F12</td>
<td>opens meta-node wizard</td>
</tr>
</tbody>
</table>
Additional Resources

**KNIME pages** ([www.knime.org](http://www.knime.org))
- **SOLUTIONS** for example workflows
- **RESOURCES/LEARNING HUB** [www.knime.org/learning-hub](http://www.knime.org/learning-hub)
- **RESOURCES/NODE GUIDE** [https://www.knime.org/nodeguide](https://www.knime.org/nodeguide)

**KNIME Tech pages** ([tech.knime.org](http://tech.knime.org))
- **FORUM** for questions and answers
- **DOCUMENTATION** for docs, FAQ, changelogs, ...
- **COMMUNITY CONTRIBUTIONS** for dev instructions and third party nodes

**KNIME TV** on YouTube [https://www.youtube.com/user/KNIMETV](https://www.youtube.com/user/KNIMETV)
Today’s Example
Today’s Example

• Classification of free-text documents is a common task in the field of text mining.

• It is used to categorize documents, i.e. assign pre-defined topics, or it can be used for sentiment analysis.

• Today we want to construct a workflow that reads and preprocesses text documents, transforms them into a numerical representation and builds a predictive model to assign pre-defined labels to documents.

• Additional tasks:
  – Sentiment analysis
  – Visualization of documents
  – Document clustering
Today’s Example

Burgermeister

Rating

Title

“Great late night burger joint”

I just had a burger there at 3 AM last night. They have great hamburgers, and awesome atmosphere and music. During the day people sit outside of the little burger 'shack' with smiles on their faces. When I got there there were no chairs, it was just people standing inside and eating (I guess due to weather) I liked it...

Author

Fulltext

Was this review helpful? Yes
Today’s Example

Goal:
• Build a **classifier** to distinguish between reviews about Italian or Chinese restaurants.

Review about an Italian or a Chinese restaurant?
Today's Example

Reading Textual Data
- Table Reader: Read Tripadvisor data
- Row Filter: no missing values
- Strings To Document: create documents
- Column Filter: only documents
- POS tagger: Assign POS Tags

Enrichment

Preprocessing
- Number Filter
- Punctuation Erasure
- Stop word Filter
- Case converter
- Snowball Stemmer
- Tag Filter

Preprocessing II
- Bag of Words Creator
- Term to String
- Group By
- Row Filter
- Filter Bag of Words: Keep only terms that occur in at least 5 documents

Transformation
- TF
- Document vector
- Document Data Extractor: Extract category for prediction (class label)

Classification
- Color Manager
- Column Filter
- Partitioning: Split into training and test set
- Decision Tree Learner
- Decision Tree Predictor
- Scorer: Score model
Bonus Examples
The KNIME Text Processing Extension
Installation

1.)

2.) KNIME & Extensions -> KNIME Textprocessing
Tip

• Increase maximum memory for KNIME

• Edit knime.ini
  – Add “-Xmx3G” as last line of knime.ini file
  – Replace 3 by the amount of gigabytes allocated for KNIME

• Useful additional extensions
  – Palladian (community extension)
    • Web crawling, Text Mining
  – XML-Processing (KNIME extension)
    • Parsing and processing of XML documents
... perhaps your name is Rumpelstiltskin[Person]? ...
Additional Data Types

• **Document Cell**
  - Encapsulates a document
    • Title, sentences, terms, words
    • Authors, category, source
    • Generic meta data (key, value pairs)

• **Term Cell**
  - Encapsulates a term
    • Words, tags
Data Table Structures

• Document table
  – List of documents

• Bag of words
  – Tuples of documents and terms

• Document vectors
  – Numerical representations of documents
Section Exercise

- Open KNIME
- Import workflows from USB stick
Importing Text
Data Source Nodes

- Typically characterized by:
  - Orange color
  - No input ports, 1 output port
New Node: File Reader

- Workhorse of the KNIME Source nodes
  - Reads text based files
  - Many advanced features allow it to read most ‘weird’ files
File Reader: Configuration

- **Basic Settings**
  - File path
  - Advanced Settings
  - Help button

- **Preview**

---

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
New Node: Excel Reader (XLS)

- Reads .xls and .xlsx file from Microsoft Excel
  - Supports reading from multiple sheets
Excel Reader Configuration

[Image of Excel Reader Configuration dialog box]

- **File path**: Specifies the file path for the Excel file to be read.
- **Sheet specific settings**: Configures specific settings for the sheet.
- **Preview**: Allows previewing of the file content.
New Node: Table Reader

- Reads tables from the native KNIME Format
- Maximum performance
- Minimum configuration
New Node: Database Reader

- Connectors for Common DB types (MySQL, Postgres, SQLite)
- Also works with any jdbc driver
- Common nodes for SQL Query Building (Groupby, Join, Filter, Sort)
Other Interesting Nodes

- PMML Reader – reads standard predictive models
- XML Reader with XPATH support
- REST/SOAP, and many more
Parser Nodes

- Node Repository: Other Data Types/Text Processing/IO
- Available Parser Nodes
  - Flat File Document Parser
  - PDF Parser
  - Word Parser
  - Document Grabber
  - ...
New Node: Tika Parser

- Reads files of various formats from directory
  - Searches for all files with specified extension in directory
  - Creates one document for each file
  - Extracts specified (meta) information
Tika Parser: Configuration

- Directory
- Recursive search
- File extensions
- Extraction of attachments
- Meta data to extract
New Node: Strings To Document

- Creation of document cells from strings
  - Converts string cells to document cells
  - Useful in combination with e.g. File Reader, XLS Reader, database nodes
Strings To Document: Configuration

- **Title**: Column, ID, Empty string
- **Text**: Full text
- **Category**: Use categories from column
- **Authors**: Use authors from column
- **Tokenizer**: Word tokenizer, OpenNLP English WordTokenizer

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
Section Exercise

- Start with “Exercise: Importing text”
  - Import string data from:
    - TripadvisorReviews-SanFranciscoRestaurants-ItalianChineseFood.table
  - Filter rows with missing titles
  - Convert strings to documents
  - Filter all columns except the document column
Section Solution

Import text

- Table Reader
- Row Filter
- Strings to Documents
- Column Filter
Enrichment
Enrichment

- Semantic information is indicated by a tag assignment
  - Part of speech, named entities (persons, organizations, genes, ...), sentiments

- A tag consists of a type and a value
  - **Type** represents the class or set of tags
    - e.g. POS (part of speech)
  - **Value** represents the actual tag value
    - e.g. NN (noun)

Column containing terms with tags

Term “food” with tag value “NN” and type “POS”
Tagging Conflicts

• In case of tag intersections the last node overwrites.

• “Serbian-American inventor Nikola Tesla developed the ...”
  
  1. POS tagger: “Serbian-American\NNP inventor\NNP Nikola\NNP Tesla\NNP developed\VBD the\DT...”
  
  2. NE tagger: “Serbian-American\NNP inventor\NNP Nikola Tesla\Person developed\VBD the\DT ...”
Unmodifiable Terms

- Tagged terms can be set **unmodifiable**
- Unmodifiable terms are **not affected** by any preprocessing node
- Preprocessing nodes can explicitly ignore unmodifiability

Set unmodifiable in tagger nodes

Ignore unmodifiability in preprocessing nodes
Tagger Nodes

• Typically characterized by:
  – Yellow color
  – 1 to 2 input ports (requiring one document column), 1 output port
  – Assignment of semantic information (tags) to terms
Tagger Nodes

- Node Repository:
  Other Data Types/Text Processing/Enrichment

- Available Tagger Nodes
  - Stanford tagger
  - Dictionary (& Wildcard) tagger
  - OpenNLP tagger
  - Abner tagger
  - ...

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/

Open for Innovation
Tagger Nodes

• Allows to specify the number of parallel threads.
• Note: each thread will load a separate model into memory!
• Tagged terms are set unmodifiable.
New Node: Stanford tagger

- Assigns part of speech tags to terms
  - Models for English, German, French (from Stanford NLP Group)
  - Alternative node: POS tagger
    - Model only for English (from OpenNLP)
Stanford Tagger: Configuration

Number of parallel threads

Model to use
New Node: Dictionary Tagger

- Assigns selected tag to matching terms
  - Matches terms in documents against terms in dictionary
  - Tag to be assigned to matching terms is specified in the dialog
  - Alternative node: Wildcard tagger
    - Terms in dictionary may contain wild cards and regular expressions
Dictionary Tagger: Configuration

[Diagram of Dictionary Tagger configuration dialog]

- **Dictionary column**
- **Type of tag to be assigned**
- **Tag value to be assigned**
- **Exact match or “contains”**
Section Exercise

• Start with “Exercise: Enrichment”
  – Assign (English) POS tags
Section Solution

Enrichment

- POS tagger
Section Exercise (Bonus)

• Start with “Exercise: Enrichment II”
  – Read files that contain positive and negative words
    • MPQA-OpinionCorpus-PositiveList.csv
    • MPQA-OpinionCorpus-NegativeList.csv
  – Assign positive and negative sentiment tags based on positive and negative word lists
  – Tip: Dictionary Tagger node
Section Solution (Bonus)

Enrichment

- File Reader
- Dictionary Tagger
Custom NER models

• The provided NER models of *OpenNLP NE tagger* and *StandfordNLP NE tagger* are trained for a few types of entities and English language only.

• For more specific applications and other languages custom models are needed.
New Node: StanfordNLP NE Learner

- Trains a NER model based on the input dictionary and corpus
  - Tag type and value can be set in the dialog
  - Creates tagged corpus based in input documents and dictionary. Trains model with tagged corpus.
Stanford Tagger: Configuration

Document corpus

Dictionary column

Tag type and value
New Node: StanfordNLP NE tagger

- Tags documents based on input NER model.
  - NER model can be specified in dialog, built-in or model from input port
StanfordNLP NE tagger: Configuration

Use model from input port or built-in models
Supplementary Workflows: NER Tagger Model Training

• Trains NER model for latin and gallic names based on “De Bello Gallico” from Julius Caesar.
Preprocessing
Preprocessing

• Reduction of feature space (terms)
• Filtering of unnecessary terms
  – Stop words, based on POS tags, dictionaries, regex, ...
• Normalization of terms
  – Stemming, case conversion
Preprocessing Nodes

- Typically characterized by:
  - Yellow color
  - 1 to 2 input ports (requiring one document column), 1 output port
  - For filtering and normalizing terms of documents and bags of words

Stop word Filter

My Node
Preprocessing Nodes

• Node Repository:

Other Data Types/Text Processing/Preprocessing

• Available Preprocessing Nodes
  – Stop Word Filter
  – Snowball Stemmer
  – Tag Filter
  – Case Converter
  – RegEx Filter
  – ...

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
Preprocessing Nodes

- **Preprocessing** tab in node dialog to specify:
  - Append original documents
  - Ignore term **unmodifiability** (set by tagger nodes).
New Node: Stop Word Filter

- Filters stop words
  - Built-in stop word lists: English, French, German, Italian, ...
  - Alternatively load custom stop word list
Stop Word Filter: Configuration

Custom stop word list

Built-in stop word lists

Dialog - 2:42 - Stop word Filter (My Node)

Flow Variables | Job Manager Selection | Memory Policy
---|---|---
Preprocessing

Case sensitive

Use built-in list

Stopword lists:

- English
- Bulgarian
- French
- German
- Hungarian
- Italian
- Polish
- Portuguese

Selected File:
C:\Users\Kilian\Desktop\MyStopWords.txt

OK | Apply | Cancel

File
New Node: Snowball Stemmer

• Reduces terms to word stem
  – For various languages: English, German, French, Italian, ...
  – Integration of Snowball stemming library
  – Alternative nodes: Porter Stemmer, Kuhlen Stemmer
    • For English only
Snowball Stemmer: Configuration

Language selection
New Node: Tag Filter

- Filters terms based on specified tag values
  - For all tag types and values
Tag Filter: Configuration

- Tag type selection
- Tag value selection
Section Exercise

• Start with “Exercise: Preprocessing”
  – Filtering:
    • Numbers
    • Punctuation marks
    • Stop words
    • All terms except: nouns, verbs, adjectives
  – Stemming
  – To lower case
Section Solution

Preprocessing

• Number Filter
• Punctuation Erasure
• Stop Word Filter
• Case Converter
• Snowball Stemmer
• POS Filter
Transformation
Transformation

• Transformation of data table structures
  – List of documents ➔ bag of words
  – Bag of words ➔ document / term vectors
  – Extraction of document fields to string columns
  – Conversion of terms to strings
Transformation Nodes

• Typically characterized by:
  – Yellow color
  – 1 input port, 1 output port
Transformation Nodes

• Node Repository:

  Other Data Types/Text Processing/Transformation

• Available Transformation Nodes
  – Bag of Words Creator
  – Document Vector
  – Strings to Document
  – Sentence Extractor
  – Document Data Extractor
  – …
New Node: Bag of Words Creator

- Transforms list of documents into bag of words
  - Original documents can be appended in a column
Bag of Words Creator: Configuration

Documents used to create bag of words

Original documents can be appended
New Node: Term to String

- Transforms term cells into string cells
  - Tag information will get lost

Bag of words with string column

```
<table>
<thead>
<tr>
<th>T</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>Document</td>
</tr>
<tr>
<td>Italian [JJ] [POS]</td>
<td>Italian pizza...</td>
</tr>
<tr>
<td>pizza [NN] [POS]</td>
<td>Italian pizza...</td>
</tr>
<tr>
<td>particular [JJ] [POS]</td>
<td>Italian pizza...</td>
</tr>
<tr>
<td>review [NN] [POS]</td>
<td>Italian pizza...</td>
</tr>
<tr>
<td>disappoint [JJ] [POS]</td>
<td>Italian pizza...</td>
</tr>
<tr>
<td>food [NN] [POS]</td>
<td>Italian pizza...</td>
</tr>
</tbody>
</table>
```

My Node
Term to String: Configuration

Terms to transform to strings
Section Exercise

• Start with “Exercise: Preprocessing II”
  – Create bag of words
  – Filter terms that occur in less than 5 documents
  – Tip: Bag of Words, GroupBy, and Reference Row Filter
Section Solution

Preprocessing II
- Bow Creator
- Term to String
- GroupBy
- Row Filter
- Reference Row Filter
New Node: Document Vector

• Transforms bag of words into document vectors
  – Requires numerical (frequency) column
  – Creates bit or numerical vectors
Document Vector: Configuration

Create bit or numerical vector

Documents to append to left of the created vector columns
New Node: Document Vector Applier

- Transforms bag of words into document vectors
  - Creates feature space of reference document vectors
  - Requires numerical (frequency) column
  - Creates bit or numerical vectors

<table>
<thead>
<tr>
<th>T Term</th>
<th>Document</th>
<th>Orig Document</th>
<th>D TF rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Italian</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.027</td>
</tr>
<tr>
<td>With[]</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.014</td>
</tr>
<tr>
<td>Pizza</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.014</td>
</tr>
<tr>
<td>[]</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.041</td>
</tr>
<tr>
<td>'m'</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.014</td>
</tr>
<tr>
<td>a[]</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.027</td>
</tr>
<tr>
<td>particular[]</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.014</td>
</tr>
<tr>
<td>eater[]</td>
<td>Italian With Pizza...</td>
<td>Italian With Pizza...</td>
<td>0.014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document</th>
<th>D dim</th>
<th>D sum</th>
<th>D wife</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Dim Sum No More...&quot;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>&quot;Hole in the wall...&quot;</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;Order of the C...&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;16th birthday su...&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;25 years between...&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;3 dishes, 3 times...&quot;</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Document Vector Applier: Configuration

Use settings from model input

Include and exclude lists of features of the reference vectors
New Node: Document Vector Hashing

- Transforms documents into document vectors
  - Vector indices of terms are determined by term hashing
  - Requires document column only
  - Creates bit or numerical vectors
  - Is streamable

```
Row ID | Document
---------------------
Row 1 | Italian With Pizzazz
Row 2 | Who Doesn't like Italian?
Row 3 | Great Italian Food Served by Italian People
Row 4 | Fantastic Italian
Row 5 | Outstanding Italian
```

Hashed document vector
Document Vector Hashing: Configuration

![Dialog - 0:150 - Document vector hashing](image)

- **Options**: Flow Variables, Job Manager Selection, Memory Policy
- **Document column setting**: Document column: Preprocessed Document
- **Hashing function setting**:
  - Dimension: 2,000
  - Seed: 1013130745
  - Hashing function: murmur3_32bit
  - Vector type: Binary
- **Output column setting**: As collection cell

- **Dimensions of document vectors**
- **Hashing function**
New Node: Document Data Extractor

- Extracts document fields as strings
  - Title, text, categories, ...

Document column

Extracted field as string column
Document Data Extractor: Configuration

Fields to extract
Frequencies

• Frequencies are based on the number of occurrences of terms
  – Locally (in documents): term frequency (TF) absolute or relative
  – Globally (in corpus): inverse document frequency (IDF)
• In order to create document vectors, frequencies have to be computed first
• Frequencies can also be used for term filtering
Frequency Nodes

- Typically characterized by:
  - Green color
  - 1 input port, 1 output port
  - Require bag of words

Append column with relative TF values
Frequency Nodes

- Node Repository:

Other Data Types/Text Processing/Frequencies

- Available Frequency Nodes
  - TF
  - IDF
  - Ngram creator
  - ...
New Node: TF

- Computes the relative or absolute term frequency (tf) of each term within a document.

Appended column with TF values.
New Node: DF

- Computes the number of documents that contain each term
New Node: IDF

- Computes three variants of inverse document frequency (IDF) for each term within the documents
  - Smooth, normalized, and probabilistic

Appended column with IDF values
New Node: Term Co-Occurrence Counter

• Counts the number of pairwise co-occurrences of terms in bag of words within selected parts of document (e.g. sentence, paragraph, title)
New Node: Ngram Creator

- Creates ngrams from documents of input table and counts their frequencies
- Both word and character ngrams are possible
Section Exercise

• Start with “Exercise: Transformation”
  – Compute relative term frequencies
  – Create document vectors
  – Extract class label / category
Transformation

- TF
- Document Vector
- Document Data Extractor
Classification
Classification

• Assigning pre-defined labels to documents
  – Categorization
  – Sentiment analysis
  – Topic assignment
• Supervised learning

• In the last section we transformed textual documents into a numerical representation (document vectors).
• We can use standard KNIME nodes to classify / analyze these vectors.
Classification

Methods:
• Decision Trees
• Neural Networks
• Naïve Bayes
• Logistic Regression
• Support Vector Machine
• Tree Ensembles
Predictive Modeling Overview

Data Partitioning

Original Data Set → Training Set → Train Model

Test Set

Training and Applying Models

Score Model

Model Evaluation
New Node: Partitioning

- Use it to split data into training and evaluation sets
- Partition by count (e.g. 10 rows) or fraction (e.g. 10%)
- Sample by a variety of methods; random, linear, stratified
Predictive Modeling Overview

Data Partitioning

Original Data Set

Training Set

Test Set

Training and Applying Models

Train Model

Apply Model

Scoring Strategies

Score Model
The Learner-Predictor Motif

- All data mining models use a Learner-Predictor motif.
- The Learner node trains the model with its input data.
- The Predictor node applies the model to a different subset of data.
Decision Tree

J.R. Quinlan, “C4.5 Programs for machine learning”
J. Shafer, R. Agrawal, M. Mehta, “SPRINT: A Scalable Parallel Classifier for Data Mining”

• C4.5 builds a tree from a set of training data using the concept of information entropy.
• At each node of the tree, the attribute of the data with the highest normalized information gain (difference in entropy) is chosen to split the data.
• The C4.5 algorithm then recourses on the smaller sub lists.
New Node: Decision Tree Learner

Decision Tree Learner

Learn Decision Tree

Dialog - 250 - Decision Tree Learner (Lear... -

File

Options

PMML Settings

General

- Class column: Category
- Quality measure: Gini index
- Pruning method: No pruning
- Reduced Error Pruning
- Min number records per node: 2
- Number records to store for view: 10,000
- Average split point
- Number threads: 4
- Skip nominal columns without domain information

Binary nominal splits

- Binary nominal splits
- Max #nominal: 10
- Filter invalid attribute values in child nodes

OK Apply Cancel
If the word “Italian” occurs in a review, the restaurant is very likely an Italian restaurant.
New Node: Decision Tree Predictor

- Consumes a Decision Tree model and new data to classify
- Check the box to append class probabilities
Predictive Modeling Overview

Data Partitioning

Original Data Set

Training Set

Test Set

Training and Applying Models

Train Model

Apply Model

Scoring Strategies

Score Model

Copyright © 2018 KNIME AG

Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license

https://creativecommons.org/licenses/by-nc-sa/4.0/
New Node: Scorer

• Compare predicted results to known truth to evaluate model quality
• Confusion matrix shows the distribution of model errors
• An accuracy statistics table provides additional info
This is the difference between the confusion matrix data table and the confusion matrix view.
### Scorer: Accuracy Measures

From the confusion matrix:

<table>
<thead>
<tr>
<th>Row ID</th>
<th>TruePositives</th>
<th>FalsePositives</th>
<th>TrueNegatives</th>
<th>FalseNegatives</th>
<th>D Recall</th>
<th>D Precision</th>
<th>D Sensitivity</th>
<th>D Specificity</th>
<th>D F-meas...</th>
<th>D Accuracy</th>
<th>D Cohens'...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>58</td>
<td>2</td>
<td>58</td>
<td>5</td>
<td>0.915</td>
<td>0.964</td>
<td>0.915</td>
<td>0.967</td>
<td>0.939</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Italian</td>
<td>58</td>
<td>5</td>
<td>54</td>
<td>2</td>
<td>0.967</td>
<td>0.921</td>
<td>0.967</td>
<td>0.915</td>
<td>0.943</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Section Exercise

- Start with “Exercise: Classification”
  - Append color information based on class labels
  - Split data into training and test set
  - Train decision tree classifier on training set
  - Apply trained model on test set
  - Score model
Section Solution

Classification

• Color Manager
• Column Filter
• Partitioning
• Decision Tree Learner
• Decision Tree Predictor
• Scorer
Classification (Bonus)

• Usually the documents used to train a model are read from a different source than that of the documents to which the model is applied afterwards.

• To apply a trained model on a second set of documents we need to ensure that all features of the training set exist as features of the second set.

• This means that all document vector columns of the training set must exist as document vector columns in the second set.
Classification (Bonus)

All features of the training set must exist as features in the second set.
Exercise: Classification II

- Start with “Exercise: Classification II”
  - Create document vectors for the second set of documents “Boston Tripadvisor Reviews”
  - The feature space of the second set has to contain all features of the training set!
  - Apply the trained model on the second set of documents
Section Solution (Bonus)

Classification II
Sentiment Analysis (Bonus)

• In sentiment analysis predefined sentiment labels, such as "positive" or "negative“, are assigned to texts.

Methods:

• Predictive modeling
• Dictionary based
• Deep parsing
• ...

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
Sentiment Analysis Example (Bonus)

- The Large Movie Review Dataset v1.0
  - 50,000 English movie reviews
  - Associated sentiment labels “positive” and “negative”

- Subset contains 2000 documents
  - 1000 positive reviews
  - 1000 negative reviews
  - ../data/IMDb-sample.csv
Predictive modeling:
• Build **classifier** to distinguish between positive and negative reviews.

  – “Ah, Moonwalker, I'm a huge Michael Jackson fan, I grew up with his music, Thriller was actually the first music video I ever saw apparently. ...”
  – “This film has a very simple but somehow very bad plot. ...”
Section Exercise (Bonus)

• Start with “Exercise: Classification III”
  – Create document cells
  – Preprocess documents
    • Punctuation Erasure, N Chars Filter, Stop Word Filter, Case converter, Snowball Stemmer
    • Filter all terms that occur in less than 20 documents
  – Create document vectors
  – Extract sentiment label and assign colors
  – Partition into training and test set
  – Train decision tree model and score it
Classification
- Strings to document
- Preprocessing nodes
- Bag of words creation, grouping, counting, and filtering
- Vector creation
- Model training and scoring
Sentiment Analysis Example (Bonus)

Dictionary based:

• Use a custom dictionary to count positive and negative words.

• Compute sentiment score to predict sentiment label.
Section Exercise (Bonus)

• Start with “Exercise: Classification IV”
  – Create document cells
  – Tag terms based on sentiment dictionaries
    • Tip: Dictionary Tagger
  – Extract and count positive and negative terms
  – Compute sentiment score based on the number of positive and negative terms
  – Predict sentiment labels based on score
  – Score predictions
Classification

- Strings to Documents
- Dictionary Tagger
- Bag of words, TF, and GroupBy for counting
- Pivoting
- Math Formula
- Rule Engine
- Scorer
Visualization
Visualization Nodes

- Typically characterized by:
  - Blue color
  - 1 input port, 1-2 output port (image port)
Visualization Nodes

• Node Repository:
  Other Data Types/Text Processing/Misc

• Available Visualization Nodes
  – Document Viewer
  – Tag Cloud

• KNIME Text Processing provides only two dedicated viz. nodes

• Various other nodes can be used for viz. too.
New Node: Tag Cloud (JavaScript)

• Shows terms visualized in a cloud
  – Colors are specified via the Color Manager
  – Requires a term and a numerical column (usually \(tf\))
  – Creates image, available at image out port

<table>
<thead>
<tr>
<th>Term</th>
<th>TF abs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Italian[J]PO...</td>
<td>230</td>
</tr>
<tr>
<td>tri[VBN][POS]</td>
<td>90</td>
</tr>
<tr>
<td>review[NNS]</td>
<td>43</td>
</tr>
<tr>
<td>disappoint[...]</td>
<td>50</td>
</tr>
<tr>
<td>food[NNS][POS]</td>
<td>569</td>
</tr>
<tr>
<td>melt[VBN][PO...]</td>
<td>5</td>
</tr>
<tr>
<td>mouth[VBP][...]</td>
<td>8</td>
</tr>
<tr>
<td>wow[VBD][POS]</td>
<td>5</td>
</tr>
<tr>
<td>water[NNS][...]</td>
<td>40</td>
</tr>
</tbody>
</table>

Size of words corresponds to frequency
Tag Cloud: Configuration

- Display only top N terms (rows)
- Term column and frequency column
Tag Cloud: View

Scaling of font size: linear, log, exp

Min and max fontsize, angle, ...
Additional Visualizations

• Decision Tree View
  – Inspect trained model
  – See which terms are discriminative
Section Exercise

• Start with “Exercise: Visualization”
  – Inspect decision tree via its view
  – Visualize bag of words using a tag cloud
  – Assign colors to terms in tag cloud (Optional)
    • Green if term occurs mostly in Chinese reviews, blue if terms occurs mostly in Italian reviews
Section Solution

Visualization

- Decision Tree Learner
- Tag Cloud
- (Optional Coloring)
  - TF, Document Data Extractor, Group By, Pivoting, Math Formula, Color Manager
New Node: Document Viewer

- Shows details of documents
  - Title, Full text
  - Meta information
  - Tagged terms can be highlighted and linked
Document Viewer: View

- List of all documents. Double click for details.
- Tagset to hilite: Tagged terms can be hilited.
- Details view with title and full text.
- Author, category, meta information, ...

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
Section Exercise

- Start with “Exercise: Visualization II”
  - View document content
  - View document content and highlight tagged terms
Visualization
• Document Viewer
Bonus Visualizations

- Supplementary Workflows/
  - R Theme River (R plot)
  - Twitter Word Tree (JavaScript view)
Clustering
Clustering

• Find groups (clusters) of similar documents
  – Topic detection
  – Exploration

• Unsupervised learning

• We can use standard KNIME nodes to cluster the numerical document vectors.
Clustering

Methods:
- Hierarchical clustering
- K-Means / Medoids
- Density based
- ...
Hierarchical Clustering

• Creates hierarchy for all data points
  – Agglomerative, bottom-up
  – Combine the “closest” data points/clusters, one at a time
• Hierarchy can be illustrated by dendrogram
• Applicable only on small data sets (<5000)

• **Complete linkage**: combine data object/cluster with minimal maximum distance
  – Finds compact, convex clusters
• **Single linkage**: combine data object/cluster with minimal minimum distance
  – Also finds concave clusters
• **Average linkage**: distance between two clusters c1 and c2 = mean distance between all points in c1 and c2
Prototype-based Clustering

• K-Medoids, K-Means, Fuzzy C-Means, ...
• Data are condensed to a small fixed number of prototypical data points
• Each prototype represents a subset of data points
• Applicable on large data sets
• Number of prototypes (k) must be specified in advance
New Node: Distance Matrix Calculate

- Computes all pairwise distances
- Different distance measures available
  - Euclidean, Manhattan, Cosine, Dice, Tanimoto, ...
- Optional distance model input port

![Distance Matrix Calculate](image)

<table>
<thead>
<tr>
<th>Document</th>
<th>Dim Sum No Mor...</th>
<th>Dim Sum No Mor...</th>
<th>Dim Sum No Mor...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16th birthday su...</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>*25 years between...</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>*3 dishes, 3 times...</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Distance Matrix Calculate: Configuration

- Name of distance column
- Distance measure
- Columns to use for distance computation
New Node: Hierarchical Clustering (DistMatrix)

- Creates hierarchy of input data points
  - Complete Linkage, Average Linkage, Single Linkage
- Requires distance column or model
Hierarchical Clustering (DistMatrix): Configuration

- Distance column
- Linkage type

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
New Node: Hierarchical Cluster View

• Shows:
  – Dendrogram of clustering
  – Distance curve
  – Colors

Hierarchical clustering model

Data points, e.g. document vectors

Dendrogram or distance

Hierarchical Cluster View

My Node
New Node: Hierarchical Cluster Assigner

- Assigns data points to clusters based on
  - Distance threshold
  - Number of clusters

Hierarchical clustering model

Data points, e.g. document vectors

Cluster assignment
Hierarchical Cluster Assigner: Configuration

Threshold or cluster count based assignment
Hierarchical Clustering: Example Workflow

Data e.g.: document vectors

Hierarchy of data points

Illustration of dendrogram

Distance Matrix Calculate

Hierarchical Clustering (DistMatrix)

Hierarchical Cluster View

Hierarchical Cluster Assigner

Assignment of clusters

Cosine distance

Complete linkage

Dendrogram

Copyright © 2018 KNIME AG
Licensed under a Creative Commons Attribution-Noncommercial-Share Alike license
https://creativecommons.org/licenses/by-nc-sa/4.0/
New Node: k-Medoids

- Computes k prototypes (medoids)
- Requires distance column or model
- Requires specification of k
- Similar nodes:
  - k-Means
  - Fuzzy c-Means

Data points and distance column

Cluster assignment
k-Medoids: Configuration

- Distance matrix column
- Cluster count $k$
- Random seed for reproducible results
k-Medoids Clustering: Example Workflow

Data e.g.: document vectors

Distance Matrix Calculate

Cosine distance

k-Medoids

Assignment of clusters

k=2
Section Exercise

• Start with “Exercise: Clustering”
  – What groups of documents are in the data?
  – Compute pairwise cosine distances
  – Apply hierarchical clustering
    • View dendrogram to find out the number of clusters (k)
    • Assign k clusters
  – Apply k-Medoids with k as number of clusters
  – Select documents of one cluster in dendrogram, hilite them, and inspect data in interactive table
Section Solution

Clustering

- Distance Matrix Calculate
- Hierarchical Clustering
  - Cluster View
  - Cluster Assigner
- k-Medoids
Supplementary Workflows
R Theme River

Creates theme river using ggplot2.

- ggplot2 has to be installed!
- Change lib path
Twitter Word Tree

Creates a word tree using the JavaScript Google charting library.
Term Co-occurrences

Term co-occurrences of all term pairs are counted on sentence and document level.
Topic Extraction

Extracts two topics from the input documents and 10 words to represent each topic.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Term</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic_0</td>
<td>food</td>
<td>237</td>
</tr>
<tr>
<td>topic_0</td>
<td>italian</td>
<td>195</td>
</tr>
<tr>
<td>topic_0</td>
<td>pizza</td>
<td>177</td>
</tr>
<tr>
<td>topic_0</td>
<td>restaur</td>
<td>147</td>
</tr>
<tr>
<td>topic_0</td>
<td>servici</td>
<td>115</td>
</tr>
<tr>
<td>topic_0</td>
<td>pasta</td>
<td>75</td>
</tr>
<tr>
<td>topic_0</td>
<td>delici</td>
<td>74</td>
</tr>
<tr>
<td>topic_0</td>
<td>excel</td>
<td>72</td>
</tr>
<tr>
<td>topic_0</td>
<td>authenti</td>
<td>66</td>
</tr>
<tr>
<td>topic_0</td>
<td>friend</td>
<td>62</td>
</tr>
<tr>
<td>topic_1</td>
<td>food</td>
<td>250</td>
</tr>
<tr>
<td>topic_1</td>
<td>sum</td>
<td>198</td>
</tr>
<tr>
<td>topic_1</td>
<td>dim</td>
<td>192</td>
</tr>
<tr>
<td>topic_1</td>
<td>restaur</td>
<td>156</td>
</tr>
<tr>
<td>topic_1</td>
<td>servici</td>
<td>106</td>
</tr>
<tr>
<td>topic_1</td>
<td>chines</td>
<td>105</td>
</tr>
<tr>
<td>topic_1</td>
<td>san</td>
<td>63</td>
</tr>
<tr>
<td>topic_1</td>
<td>view</td>
<td>62</td>
</tr>
<tr>
<td>topic_1</td>
<td>chinatown</td>
<td>59</td>
</tr>
<tr>
<td>topic_1</td>
<td>time</td>
<td>57</td>
</tr>
</tbody>
</table>
RESTful Geolocation

REST call to get lat long for IPs

Try Catch Block
RESTful Geolocation

• Translates IPs to geo coordinates via RESTful service

• GET Resource: access RESTful API via GET

• IP to geo coordinates (lat/lon)

• Read REST Representation: parse REST result
  – JSON, XML, CSV, ...

• Try Catch nodes to log errors gracefully
Geographic Analysis

KNIME Desktop Registered Users Report
Example Using Geographical Data

Global Users

KNIME Desktop Registrations by Country

KNIME Desktop Registrations in the USA

OSM Map View

Cities

Heatmap World Cities

Image to Report

World

Date to Report

Top 5 Countries

Aggregate by Count

WorldData

OSM Map View

Image to Report

OSMMap

USA

Aggregate by USA

Date to Report

Top 5 States

Table Reader

Node 121

OSM Map to Image

Node 87

GroupBy

Columns to Grid (deprecated)

Date to Report

Eastern Mass.

Node 105

Node 106

Node 109

Eastern Mass Users
Geographic Analysis

- Reads IPs from download weblog and related geo coordinates
- Aggregates downloads by city, country, and US states
- OSM Map View to visualize geo coordinates
- OSM Map to Image to create image of map view
Social Media Analysis

Leader / Follower analysis of users

Sentiment analysis of users
Social Media Analysis

- Slashdot forum data
- Text Mining: sentiment analysis of users
- Network Mining: leader and follower scoring of users
Romeo and Juliet

- Read epub file
- Tag character names and count frequencies
- Load JPEG and convert to PNG
- Insert PNG images and visualize network
Romeo and Juliet

- Interaction network of characters.
- Border color indicates family assignment.
- Node size is related to TF of character names.
The End

education@knime.com