

Interactive Analytics Library

Emily Wall and Alex Endert
{emilywall},{endert}@gatech.edu

Interaction is data.

<http://gtvalab.github.io/ial/>

Motivation

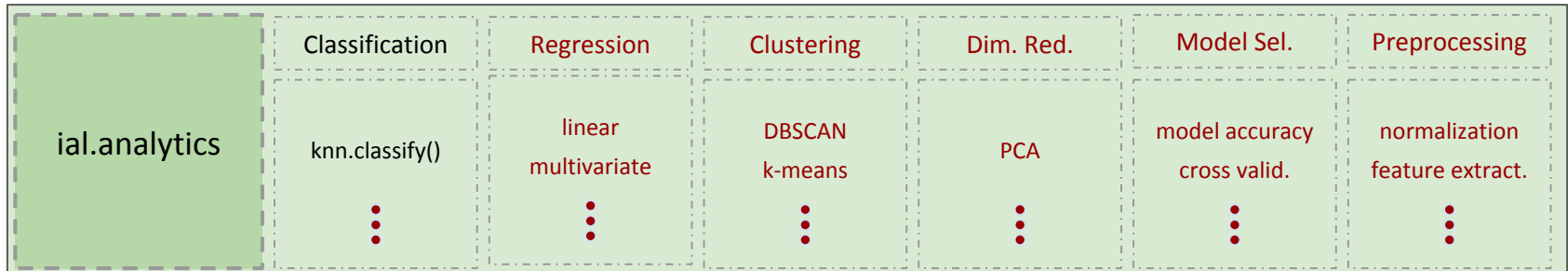
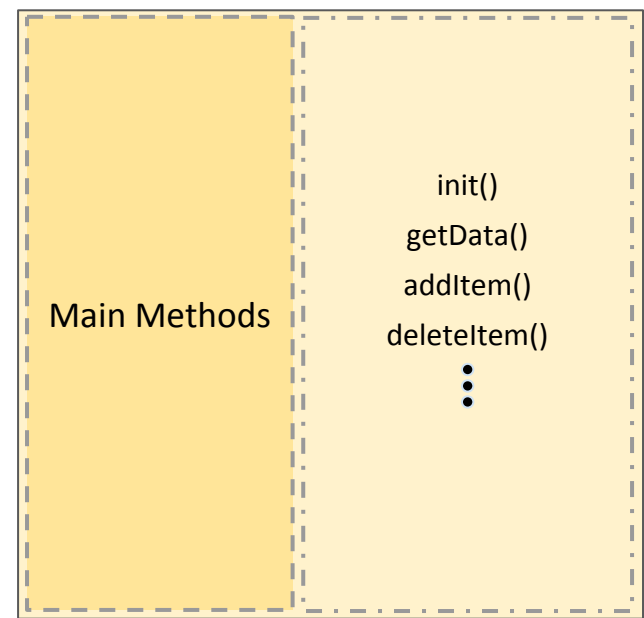
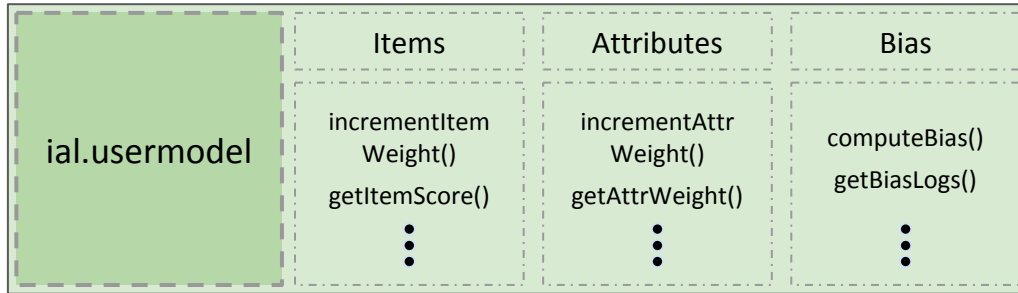
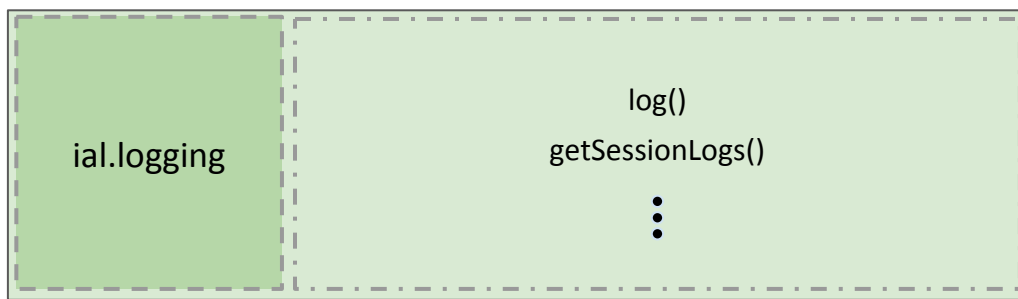
- Interaction is data.
 - What can we tell about a user based on their interactions with data?
 - In the case of streaming data: when the user can't examine all of the data, how can a system leverage the user's goals, interests, etc.?
- Facilitate faster prototyping of web-based visual analytic applications
 - Written in javascript
 - Flexibility to export results for further customized analytic operations in R, python, etc.

Design

ial.logging

ial.usermodel

ial.analytics



IAL TEST BED DEMO

ial.logging

ial.usermodel


ial.analytics

<http://gtvalab.github.io/ial/>

ial.logging

- What: create customized logs of a user's interactions
- How: add a call to ial.logging within event-handlers in JS

ial.logging

Code  JSON 

```
d3.selectAll('.dot').on('mouseover', function(d) {  
    ial.logging.log(d, new Date(), 'hover');  
    tip.show(d);  
}).on('mouseout', tip.hide);
```

```
{  
  "dataItem": {  
    "Name": "Honda Insight 2dr (gas/electric)",  
    "Type": "Sedan",  
    "AWD": "0",  
    "RWD": "0",  
    "Retail Price": "19110",  
    "Dealer Cost": "17911",  
    "Engine Size (l)": "2",  
    "Cyl": "3",  
    "HP": "73",  
    "City MPG": "60",  
    "Hwy MPG": "66",  
    "Weight": "1850",  
    "Wheel Base": "95",  
    "Len": "155",  
    "Width": "67",  
    "ial": {  
      "id": "143",  
      "weight": 1,  
      "screen_time": 0,  
      "itemScore": 0.6838  
    }  
  },  
  "eventName": "hover",  
  "oldWeight": "",  
  "newWeight": "",  
  "eventSpecificInfo": {},  
  "eventTimeStamp": "2017-05-22T03:33:50.474Z"  
}
```


ial.logging

- Functions in ial.logging:
 - ial.logging.log()
 - ial.logging.getSessionLogs()
 - ial.logging.getItemLogs()
 - ial.logging.getAttributeLogs()
 - ial.logging.printSessionLogs()
 - ial.logging.setMaxQueueSize()
 - ial.logging.enqueue()
 - ial.logging.dequeue()

ial.usermodel

- What: create a model of user's interest based on user interactions
 - Represented as two weight vectors: items and attributes
 - Which data items are of interest? Which attributes are of interest?
- How: assigning weight to interactions and add a call to ial.usermodel within event-handlers in JS

ial.usermodel: items

Name: Mini Cooper

Weight: 8

● single_click ● double_click ● hover



Name: Mini Cooper S

Weight: 5

● single_click ● double_click ● hover



Name: Honda Insight 2dr (gas/electric)

Weight: 5

● single_click ● double_click ● hover



Name: Scion xB

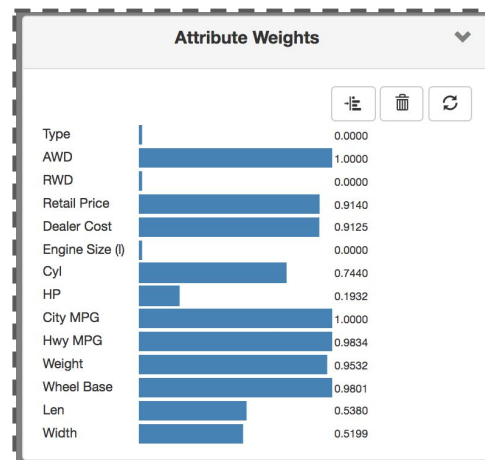
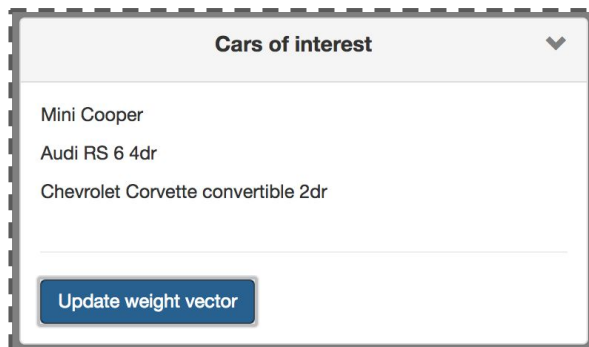
Weight: 4

● single_click ● double_click ● hover



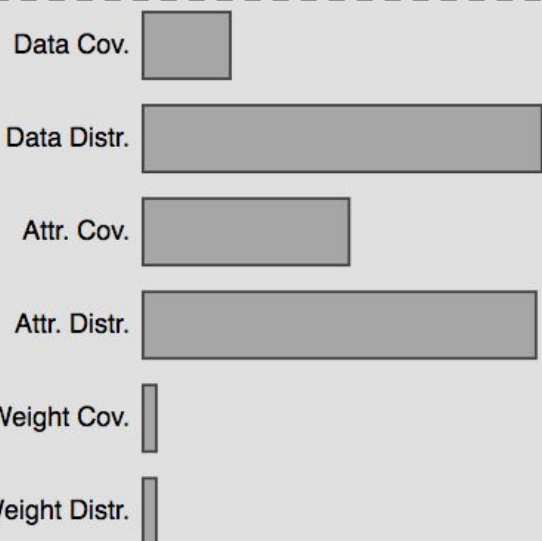
```
var hoverWeight = 0;
var clickWeight = 1;
var dblClickWeight = 2;
d3.selectAll('.dot').on('mouseover', function (d) {
    ial.usermodel.incrementItemWeight(d, hoverWeight,
        true, {'level':'INFO', 'eventType':'hover'});
    tip.show(d);
}).on('mouseout', tip.hide)
.on('click', function (d) {
    ial.usermodel.incrementItemWeight(d, clickWeight,
        true, {'level':'INFO', 'eventType':'single_click'});
    showDetails(d);
}).on('dblclick', function(d) {
    ial.usermodel.incrementItemWeight(d, dblClickWeight,
        true, {'level':'INFO', 'eventType':'double_click'});
});
```

ial.usermodel: attributes



```
var interestObjectsList = [];  
for (var i in interestPointsList) {  
    interestObjectsList.push(getCarObjectByName(interestPointsList[i]));  
}  
  
var derivedWeightVector = ial.usermodel.generateAttributeWeightVectorUsingSimilarity(interestObjectsList);  
ial.usermodel.setAttributeWeightVector(derivedWeightVector, true,  
    {'level': 'INFO', 'eventType': 'set_attribute_weight_vector'});  
updateAttributeWeightDiv();
```

ial.usermodel: bias



```
d3.selectAll('.dot').on('mouseover', function(d) {  
    ial.logging.log(d, new Date(), 'hover');  
    updateBias();  
}).on('click', function(d) {  
    ial.logging.log(d, new Date(), 'click');  
    updateBias();  
});  
  
function updateBias() {  
    var biasResults = ial.usermodel.bias.computeBias();  
    updateBiasVis(biasResults);  
}
```

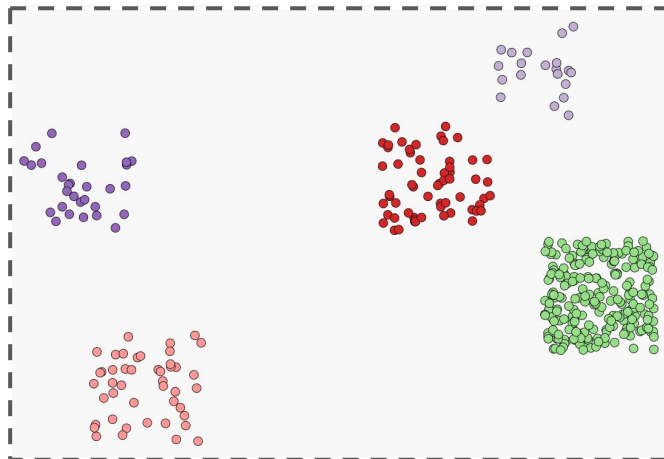
ial.usermodel

- Functions in ial.usermodel:
 - ial.usermodel.setItemWeight()
 - ial.usermodel.incrementItemWeight()
 - ial.usermodel.setAttributeWeight()
 - ial.usermodel.incrementAttributeWeight()
 - ial.usermodel.getItemScore()
 - ial.usermodel.getTopNPointsByInteractionWeights()
 - ial.usermodel.getTopNPointsByScores()
 - ial.usermodel.getNSimilarPoints()
 - ial.usermodel.getSimilarityScore()
 - ial.usermodel.generateAttributeWeightVectorUsingSimilarity()
 - ial.usermodel.generateAttributeWeightVectorUsingDifferences()

ial.analytics

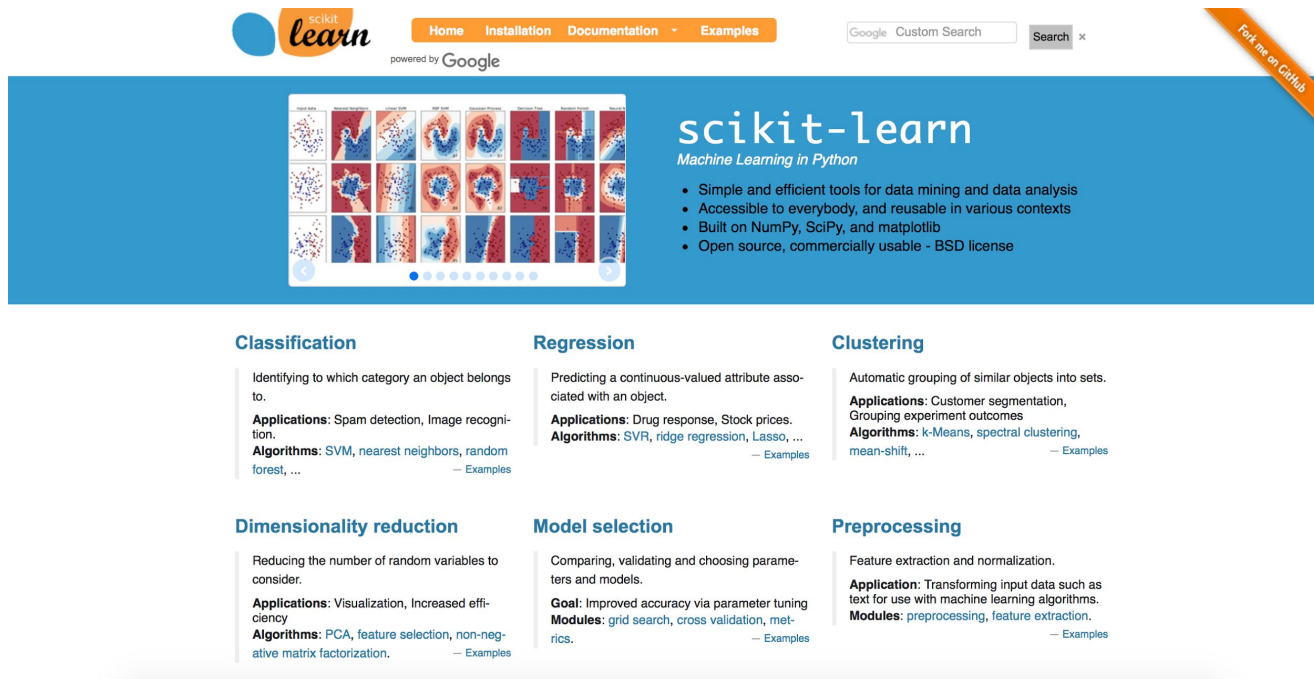
- What: perform common analytic operations using data, logs, or user model
 - Can be unsupervised, or can be steered with user model to create adaptable interfaces
- How: add a call to ial.analytics

ial.analytics



```
$("#createClusterButton").click(function (ev) {  
  var clusterList = ial.analytics.createClusters();  
  activeClusterList = clusterList;  
});  
  
$("#groupClustersButton").click(function (ev) {  
  main.drawKNN(params.width, params.height, activeClusterList);  
});  
  
$("#colorByClusterButton").click(function (ev) {  
  d3.selectAll('.dot').transition().duration(1000).style('fill', function(d) {  
    return color(d.ial.KNNClusterId);  
  });  
});
```


ial.analytics



scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extraction. — Examples

ial.analytics

- Functions in ial.analytics:
 - ial.analytics.createClusters()
 - ial.analytics.classify()
 - ial.analytics.linearRegression()
 - ial.analytics.dbscan()
 - ial.analytics.pca()

Using IAL

- Who can use it?
 - researchers
 - designers
 - etc.
- What to use it for?
 - logging
 - user modeling
 - analytic operations
- Applications?
 - analytic provenance from log analysis
 - recommendation systems
 - adaptive model steering
 - smart sampling of streaming data

<http://gtvalab.github.io/ial/>

BIAS DEMO

ial.logging

ial.usermodel

ial.analytics

PODIUM DEMO

ial.logging

ial.usermodel*

ial.analytics*

<http://gtvalab.github.io/ial/>

Questions?

<http://gtvalab.github.io/ial/>