Deep Feature Synthesis:
Towards Automating Data Science Endeavors

Max Kanter, Kalyan Veeramachaneni
CSAIL, MIT
New challenge: increased rate of new data problems
Diverse problems, but similar process

```
{SELECT ?,
   users, user_id,
   FLOOR (...) FROM mooc demands...
}

{ON users.user_id, week, ...
   AND FLOOR ...
   GROUP BY users.user_id ...
}

{WHERE users.user_id IS NOT NULL,
   ...
   GROUP BY user.user_id ...
}
```
Diverse problems, but similar process

Organize

Extract & Aggregate

{SELECT ?,
users, user_id,
FLOOR (????)
FROM moocdo.users...}

{ON users.user_dropout_week,
..., AND FLOOR II....
GROUP BY users.user_id ... ?

{WHERE users.user_id IS NOT NULL,
..., GROUP BY user.user_id ... }

Interpret & Represent

Model

Features

Entity

Features

Entity
Deep Feature Synthesis
Deep Feature Synthesis Key Ideas

- Features are often derived using relationships in the dataset
Types of relationships

Forward

Logs
CustomerID
Date
EventType
ProductID

Products
ProductID
ProductCategory
Merchant
Price

Customers
CustomerID
Gender
Age
Churned

Orders
OrderID
CustomerID
Date

Backward

OrderProduct
OrderID
ProductID
Types of relationships

Forward

Logsf
- CustomerID
- Date
- EventType
- ProductID

Customersf
- CustomerID
- Gender
- Age
- Churned

Products
- ProductID
- ProductCategory
- Merchant
- Price

Orders
- OrderID
- CustomerID
- Date

OrderProducts
- OrderID
- ProductID

Backward

Types of relationships

Forward

Logs
CustomerID
Date
EventType
ProductID

Products
ProductID
ProductCategory
Merchant
Price

Orders
OrderID
CustomerID
Date

Customers
CustomerID
Gender
Age
Churned

Backward

OrderProduct
OrderID
ProductID
Deep Feature Synthesis Key Ideas

- Features are often derived using relationships in the dataset.
- Across datasets, many features are derived using similar mathematical operations.
3 Feature Abstractions

EFEAT

- 1 entity
- Single row
  - weekday
  - ratios, differences
- Multiple row
  - derivatives
  - bag of words
3 Feature Abstractions

- Direct reflection
- 2 entities
- Forward Relationships
- Examples:
  - reflect metadata
3 Feature Abstractions

- 2 entities
- Backward relationships
- Aggregate rows and apply functions
- Examples
  - count, mean, # distinct, std
- Can condition on categorical feature
Deep Feature Synthesis Key Ideas

- Features are often derived using relationships in the dataset
- Across datasets, many features are derived using similar mathematical operations
- New features are often composed using previously derived features
How to Stack Feature Types

Deep Feature Synthesis

All backward relationships $E_B$

RFEAT

EFTEAT

DFEAT

Deep Feature Synthesis

All forward relationships $E_F$
Algorithm 1 Generating features for target entity

1: function MAKE_FEATURES($E^i, E^{1:M}, E_V$)
2:     $E_V = E_V \cup E^i$
3:     $E_B = \text{BACKWARD}(E^i, E^{1...M})$
4:     $E_F = \text{FORWARD}(E^i, E^{1...M})$
5:     for $E^j \in E_B$ do
6:         MAKE_FEATURES($E^j, E^{1...M}, E_V$)
7:         $F^j = F^j \cup \text{RFFeat}(E^i, E^j)$
8:     end for
9:     for $E^j \in E_F$ do
10:        if $E^j \in E_V$ then
11:           continue
12:        end if
13:        MAKE_FEATURES($E^j, E^{1...M}, E_V$)
14:        $F^i = F^i \cup \text{DFFeat}(E^i, E^j)$
15:     end for
16:     $F^i = F^i \cup \text{EFeat}(E^i)$
### Example

#### Deep Feature, $d=3$

<table>
<thead>
<tr>
<th>CustomerID</th>
<th>Gender</th>
<th>Age</th>
<th>AVG(Orders.SUM(Product.Price))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>f</td>
<td>45</td>
<td>$250</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Base Column

<table>
<thead>
<tr>
<th>ProductID</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$100</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>$200</td>
</tr>
<tr>
<td>4</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Deep Feature, $d=2$

<table>
<thead>
<tr>
<th>OrderID</th>
<th>Customer ID</th>
<th>SUM(Product.Price)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>$300</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>$200</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Deep Feature, $d=1$

<table>
<thead>
<tr>
<th>ID</th>
<th>OrderID</th>
<th>ProductId</th>
<th>Product.Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>$200</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>$100</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>$200</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Deep Feature Synthesis Key Ideas

- Features are often derived using relationships in the dataset.
- Across datasets, many features are derived using similar mathematical operations.
- New features are often composed using previously derived features.
- With candidate features for given prediction, there are effective feature selection techniques.
Generalized Machine Learning Pipeline

Feature Selection and dimensionality reduction

Model tuning

Optimized using Gaussian Copula Process
Gaussian Copula Process

Steps:

1. Sample randomly
2. Execute the ML pipeline and assess cross validated performance
3. Model the non-linear function between the parameters and the performance using Gaussian copula process
4. Identify the neighborhood where we should sample next
5. Iterate — goto step 2

Algorithm by Dubois and Veeramachaneni. An implementation by Sébastien Dubois can be found at http://hdi-project.github.io/DeepMining/
Publicly held data science competitions:

**Project Excitement (KDD 2014)**
Using past projects' histories on DonorsChoose.org, predict if a crowd-funded project is "exciting".

**Repeat Buyer Prediction (IJCAI 2015)**
Using past merchant and customer shopping data, predict if a customer making a purchase with a promotion will turn into a repeat buyer.

**Student Dropout (KDD 2015)**
Using student interaction with resources on an online course, predict if they will dropout in next 10 days.
Input to DSM: Relational Datasets

Project Excitement

Dropout Prediction

Repeat Buyer
# Results

<table>
<thead>
<tr>
<th>Competition</th>
<th># Teams</th>
<th>% Top Submission's Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD 2014</td>
<td>473</td>
<td>86.5%</td>
</tr>
<tr>
<td>IJCAI 2015</td>
<td>156</td>
<td>93.7%</td>
</tr>
<tr>
<td>KDD 2015</td>
<td>277</td>
<td>95.7</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>302</strong></td>
<td><strong>92.0%</strong></td>
</tr>
</tbody>
</table>
### Results

<table>
<thead>
<tr>
<th>Competition</th>
<th>% Teams worse</th>
<th>Days spent on worse submissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD 2014</td>
<td>69.3%</td>
<td>929</td>
</tr>
<tr>
<td>IJCAI 2015</td>
<td>32.7%</td>
<td>-</td>
</tr>
<tr>
<td>KDD 2015</td>
<td>85.6%</td>
<td>377</td>
</tr>
<tr>
<td><strong>Avg:</strong></td>
<td><strong>62.5%</strong></td>
<td><strong>Sum:</strong> 1307 days</td>
</tr>
</tbody>
</table>
Results

Lines show the standing of the Data Science Machine in the competition as of May 18th, 2015
Results

Lines show the standing of the Data Science Machine in the competition as of May 18th, 2015.
Conclusions

- Features created by Deep Feature Synthesis can be used to build competitive predictive models, but humans still get best performance.

- The Data Science Machine can be used to get baseline predictive model.

- The Data Science Machine can enable non-data scientists to build predictive models.
Website
bit.ly/mitdsm

Contact
Max Kanter - kanter@mit.edu
Kalyan Veeramachaneni - kalyanv@mit.edu
### Example

#### Deep Feature, $d=2$

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Edu Level</th>
<th>AVG(Session.\text{SUM(Resource.Time)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>PHD</td>
<td>120</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Deep Feature, $d=1$

<table>
<thead>
<tr>
<th>SessionID</th>
<th>StudentID</th>
<th>SUM(Resource.Time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Base Column

<table>
<thead>
<tr>
<th>ID</th>
<th>SessionId</th>
<th>ResourceID</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>80</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Generalized Machine Learning Pipeline

<table>
<thead>
<tr>
<th>Param</th>
<th>Default</th>
<th>Range</th>
<th>Function</th>
<th>KDD cup 2014</th>
<th>IJCAI</th>
<th>KDD cup 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>1</td>
<td>[1, 6]</td>
<td>The number of clusters to make</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$n_c$</td>
<td>100</td>
<td>[10, 500]</td>
<td>The number of SVD dimensions</td>
<td>389</td>
<td>271</td>
<td>420</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>100</td>
<td>[10, 100]</td>
<td>The percentage of top features selected</td>
<td>32</td>
<td>65</td>
<td>18</td>
</tr>
<tr>
<td>$rr$</td>
<td>1</td>
<td>[1, 10]</td>
<td>The ratio to re-weight underrepresented classes</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>$n$</td>
<td>200</td>
<td>[50, 500]</td>
<td>The number of decision trees in a random forest</td>
<td>387</td>
<td>453</td>
<td>480</td>
</tr>
<tr>
<td>$m_d$</td>
<td>None</td>
<td>[1, 20]</td>
<td>The maximum depth of the decision trees</td>
<td>19</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$\beta$</td>
<td>50</td>
<td>[1, 100]</td>
<td>The maximum percentage of features used in decision trees</td>
<td>32</td>
<td>51</td>
<td>34</td>
</tr>
</tbody>
</table>
### TABLE III.
The number of rows and the number of synthesized features per entity in each dataset. The uncompressed sizes of the KDD CUP 2014, IJCAI, and KDD CUP 2015 are approximately 3.1 GB, 1.9 GB, and 1.0 GB, respectively.

<table>
<thead>
<tr>
<th>Entity</th>
<th>KDD Cup 2014</th>
<th></th>
<th>Entity</th>
<th>IJCAI</th>
<th></th>
<th>Entity</th>
<th>KDD Cup 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>664,098</td>
<td>935</td>
<td>Merchants</td>
<td>4995</td>
<td>43</td>
<td>Enrollments</td>
<td>2,000,904</td>
</tr>
<tr>
<td>Schools</td>
<td>57,004</td>
<td>430</td>
<td>Users</td>
<td>424,170</td>
<td>37</td>
<td>Users</td>
<td>112,448</td>
</tr>
<tr>
<td>Teachers</td>
<td>249,555</td>
<td>417</td>
<td>Behaviors</td>
<td>54,925,330</td>
<td>147</td>
<td>Courses</td>
<td>39</td>
</tr>
<tr>
<td>Donations</td>
<td>2,716,319</td>
<td>21</td>
<td>Categories</td>
<td>1,658</td>
<td>12</td>
<td>Outcomes</td>
<td>120,542</td>
</tr>
<tr>
<td>Donors</td>
<td>1,282,092</td>
<td>4</td>
<td>Items</td>
<td>1,090,390</td>
<td>60</td>
<td>Log</td>
<td>13,545,124</td>
</tr>
<tr>
<td>Resources</td>
<td>3,667,217</td>
<td>20</td>
<td>Brands</td>
<td>8,443</td>
<td>43</td>
<td>Objects</td>
<td>26,750</td>
</tr>
</tbody>
</table>
| Vendors    | 357          | 13 | ActionType | 5     | 36| ObjectChildren | 26,033 | 3
| Outcomes   | 619,326      | 13 | Outcomes   | 522,341 | 82| EventTypes | 7            |
| Essays     | 664,098      | 9  |            |      |   |            |              |