SIEMENS

Sam Zheng, Dou Qi, Baoxiang Wang, Tongfang Sun

Research of Machine Learning in User Experience Field

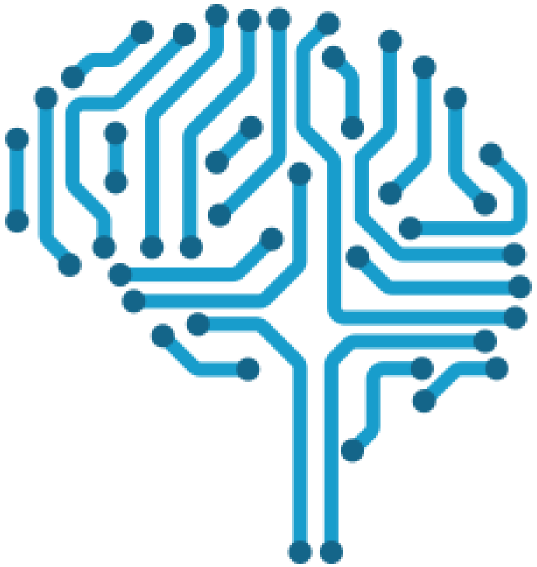
AUG, 16th , 2016
UEX Team – Princeton, NJ

About

Machine Learning (ML)

“**Machine learning** is a subfield of computer science that evolved from the study of **pattern recognition** and **computational learning theory** in artificial intelligence.”

- Wiki

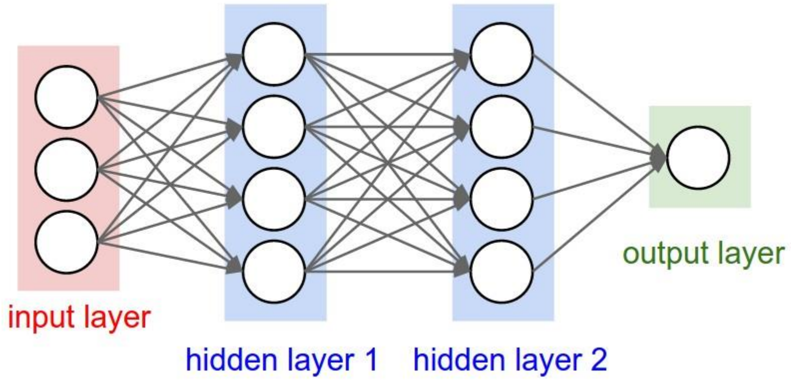


“Field of study that gives computers the **ability to learn** without being explicitly programmed.”

- Arthur Samuel, in 1959

Approaches

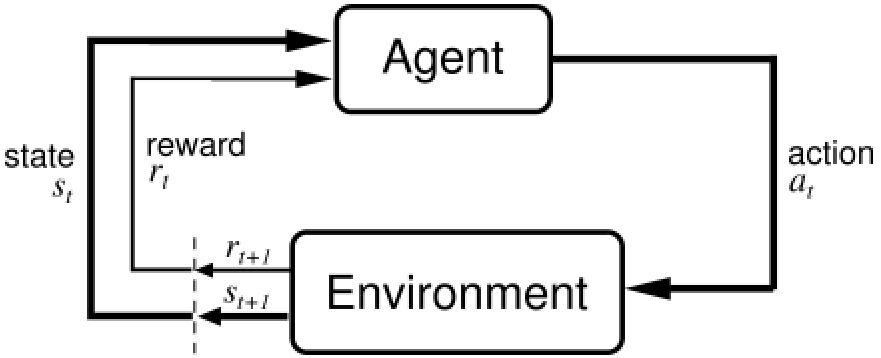
- Decision Tree Learning
- **Artificial Neural Network**
- **Deep Learning**
- Reinforcement Learning
- Clustering
- Bayesian network
- ……



e.g. Convolutional Neural Network (CNN)

Approaches

- Decision Tree Learning
- Artificial Neural Network
- Deep Learning
- **Reinforcement Learning**
- Clustering
- Bayesian network
- ……



AlphaGo

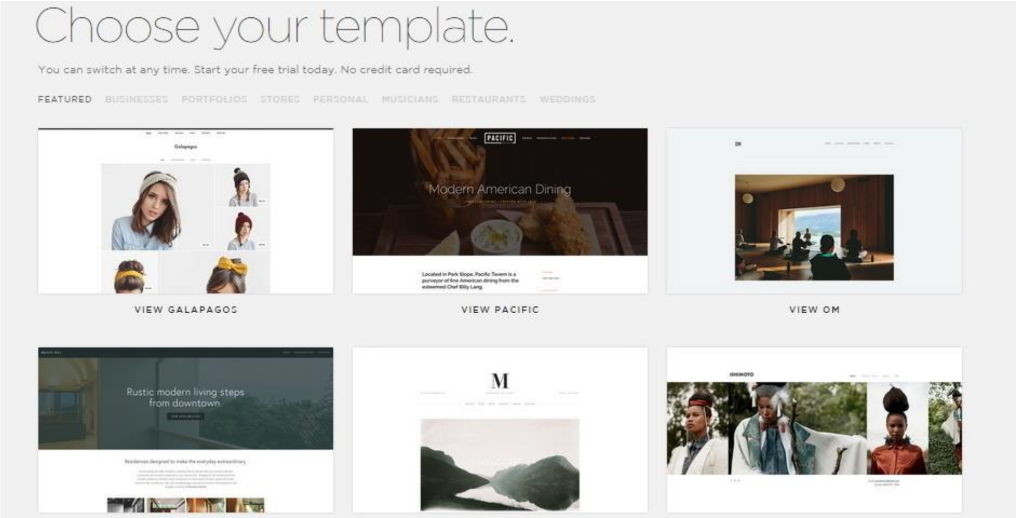
eg. Reinforcement Learning

Why we need this?

Machine Learning is like our **eyes** and **brains**, we want to use that to train our **visual design** and use that to improve **design logic** by learning **user behavior**

Machine Learning and User Experience

What we did?



Website Aesthetics

User Behavior and Game Strategy

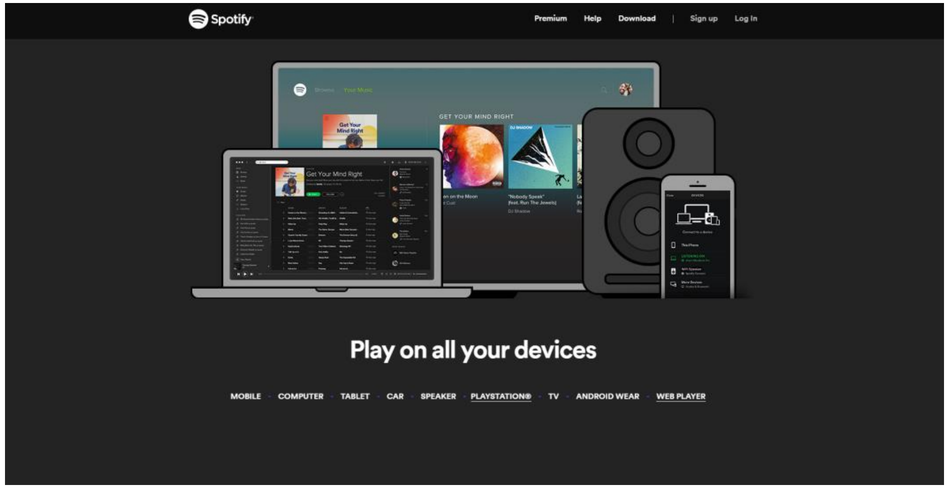
Background

SIEMENS

Webpage Aesthetics are a decisive factor for engaging users online

Users' Aesthetics

- Estimate a **good** design of a webpage
  - Complexity
  - Color scheme
  - Balance
  - Texture
  - …



Background

SIEMENS

Machine's Aesthetics

- Let computer learn to automatically estimate webpage aesthetics from big data

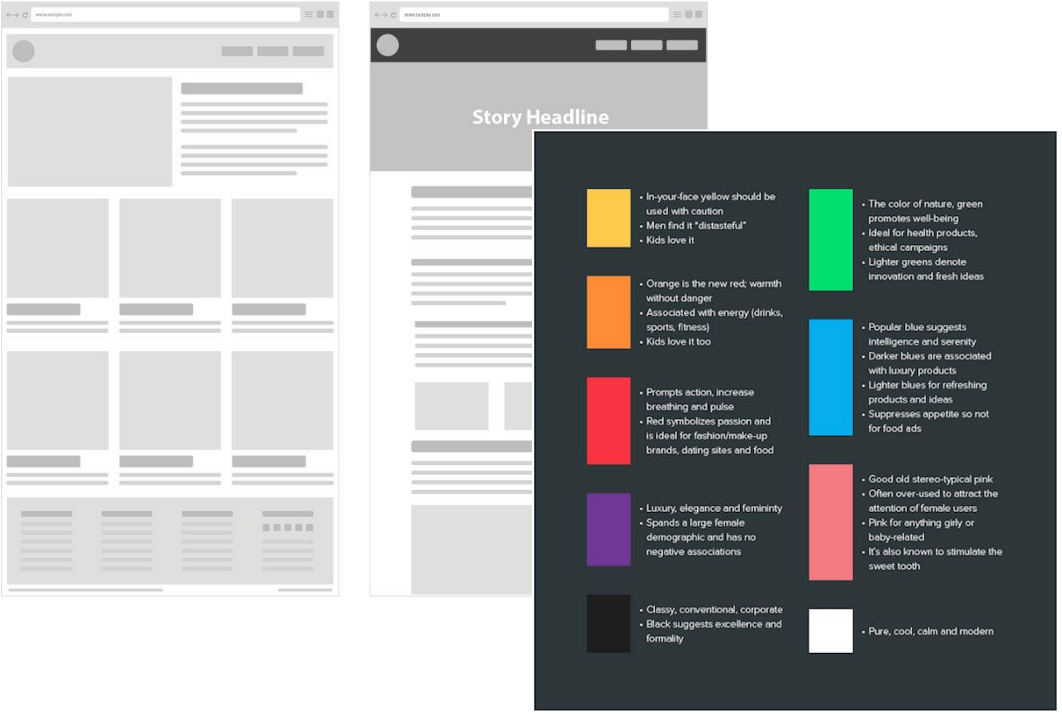
- Identify and quantify key image features that are predictive of favorable aesthetic judgements by individuals

Website Aesthetics
Related Works

Webpage aesthetics rating with **hand-crafted** features from color, context, space layout

- Correlating low-level image statistics with users' rapid aesthetics and affective judgments of webpages , *Xianjun Sam Zheng et al. CHI'09*

- Predicting users' first impression of website aesthetics with a quantification of perceived visual complexity and colorfulness , *Reinecke et al. CHI'13*



Complexity and Colorfulness

Related Works

SIEMENS

Photo aesthetics rating with **deep learning** features from convolutional neural networks

- Deep multi-patch aggregation network for image style, aesthetics, and quality estimation, *Lu et al. ICCV'15*
- Photo aesthetics ranking networks with attributes and content adaptation, *Kong et al. ECCV'17*

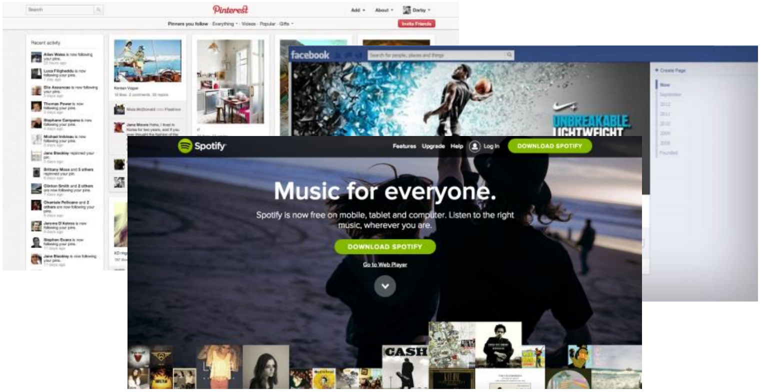
Low Quality

High Quality

Lu et al. ICCV'15

Method

SIEMENS

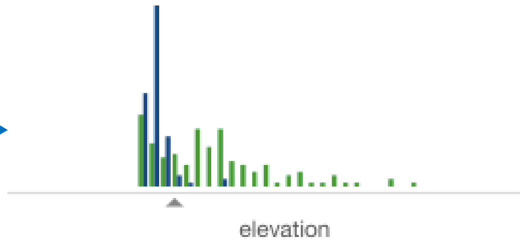


Webpages

Deep Neural Network

- Learn webpage aesthetics from big data
- Rate new webpages automatically

CNN algorithm



Rating

Method

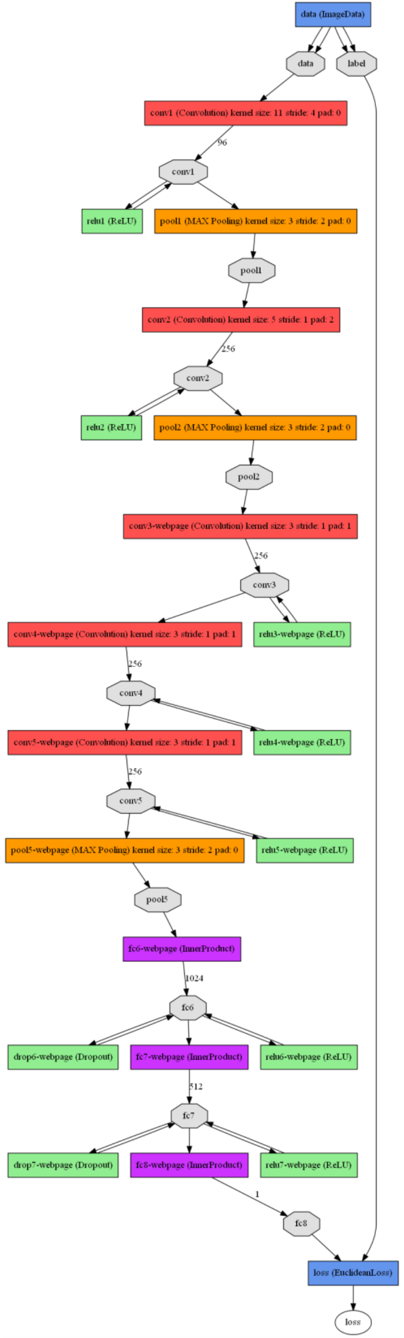
SIEMENS

Deep Neural Network

- 11 layer convolutional neural networks

Table 1: Architecture of the network.

Layer	Kernel	Stride	Channel
conv1	11×11	4	96
pool1	3×3	2	96
conv2	5×5	1	256
pool2	3×3	2	256
conv3	3×3	1	256
conv4	3×3	1	256
conv5	3×3	1	256
pool5	3×3	2	256
fc6	-	-	1024
fc7	-	-	512
Regression	-	-	1



Method

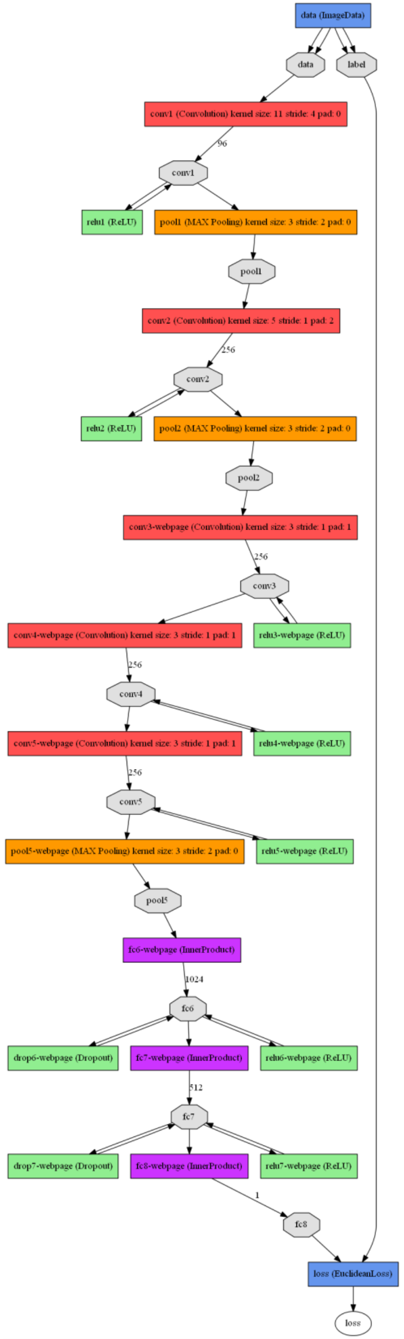
SIEMENS

Deep Neural Network

- 11 layer convolutional neural networks
- Regression to an output rating
- Minimize Euclidean distance

$$\mathcal{L} = \frac{1}{2N} \sum_{n=1}^{N} ||\hat{y}_n - y_n||_2^2$$

- Stochastic gradient descent to update weights in optimization



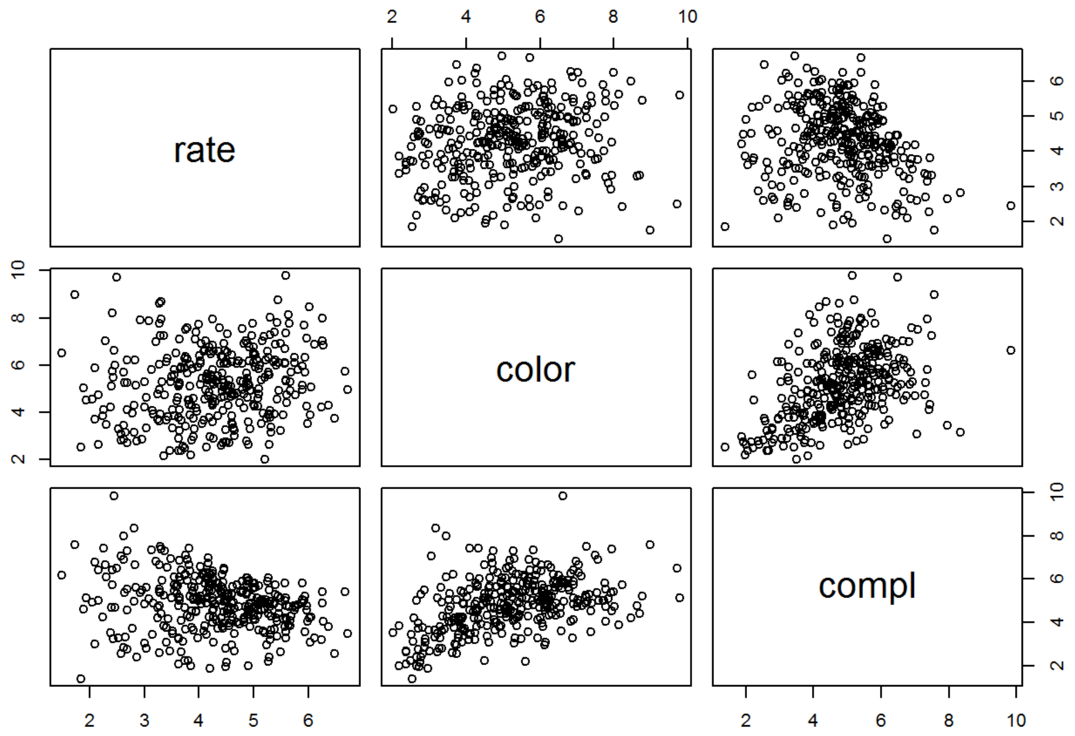
Method

Linear regression (*baseline comparison*)

- hand-crafted complexity and colorfulness
- Rate, colorfulness and complexity ***Pearson Correlation***

```
cor(df[, 1:3], method = "pearson")
```

```
##              rate      color       compl
## rate    1.0000000 0.1540118 -0.2193856
## color   0.1540118 1.0000000  0.4357306
## compl  -0.2193856 0.4357306  1.0000000
```



Method

SIEMENS

Linear regression (*baseline comparison*)

- Build fixed normal linear regression to find correlation

```
n.lm <- lm(rate ~ color + compl, data = dat_tarin)
summary(n.lm)
```

```
##
## Call:
## lm(formula = rate ~ color + compl, data = dat_tarin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89389 -0.59519  0.07316  0.69006  2.43365
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.60673    0.23528  19.580  < 2e-16 ***
## color        0.21033    0.03876   5.426 1.11e-07 ***
## compl       -0.29332    0.04710  -6.228 1.41e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9745 on 335 degrees of freedom
## Multiple R-squared:  0.125,  Adjusted R-squared:  0.1198
## F-statistic: 23.94 on 2 and 335 DF,  p-value: 1.921e-10
```

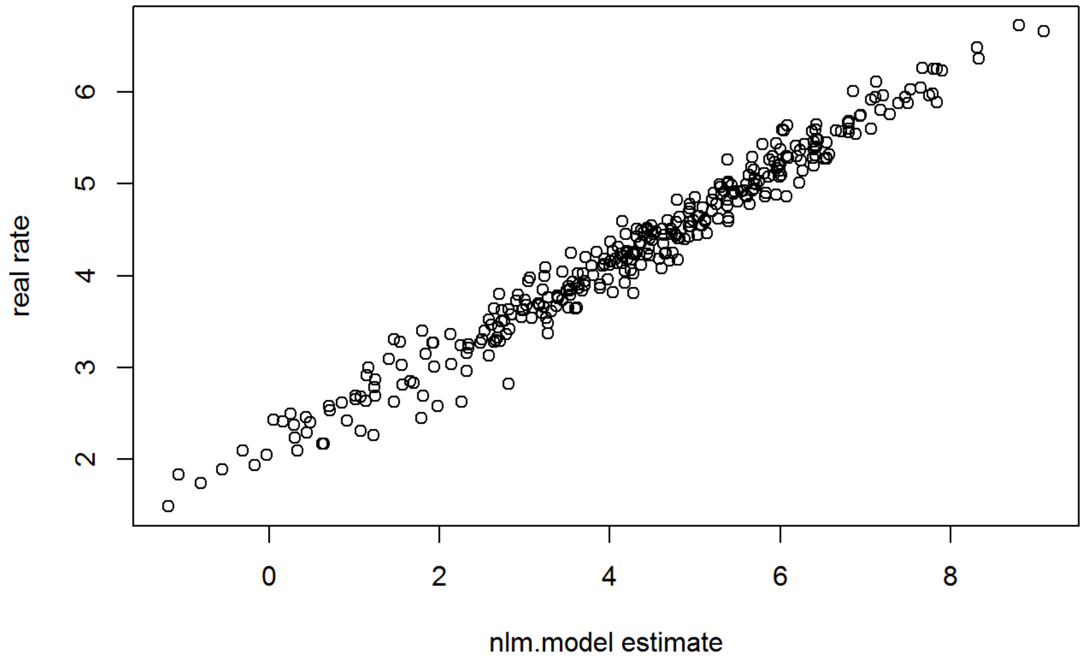
Method

SIEMENS

Linear regression (*baseline comparison*)

- Build fixed normal linear regression to find correlation

```
n.lm <- lm(rate ~ color + compl, data = dat_tarin)
summary(n.lm)
```

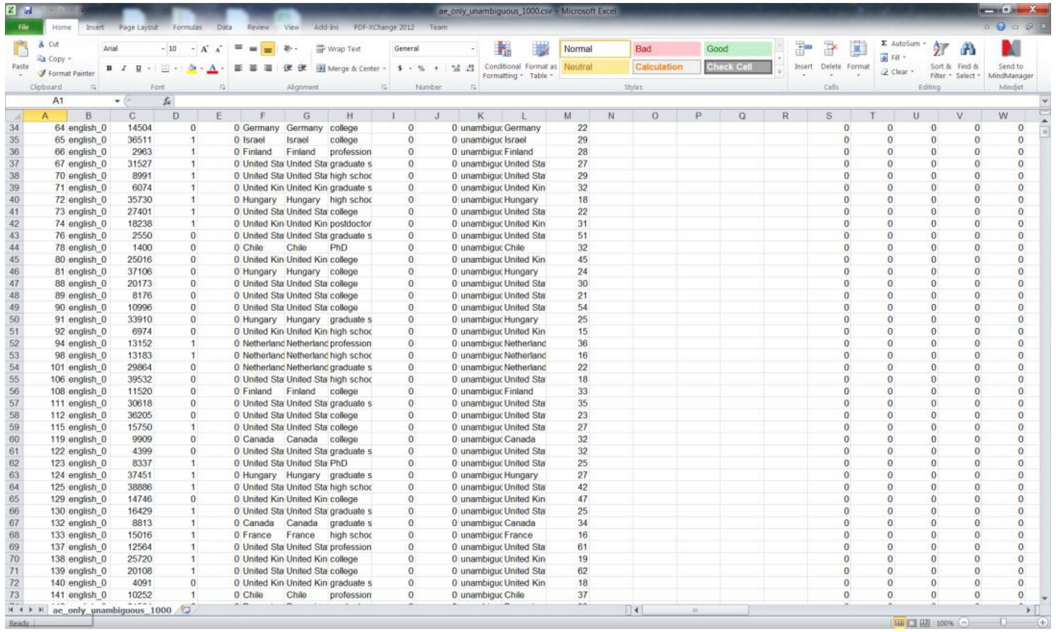


Experiment

SIEMENS

Dataset

- 398 webpages (languages of English and Foreign)
- 300 training (75.4%), 38 validation (9.5%), 60 testing (15%)
- Users rating from 40000 people, averagely each webpage received 1000 ratings



Experiment

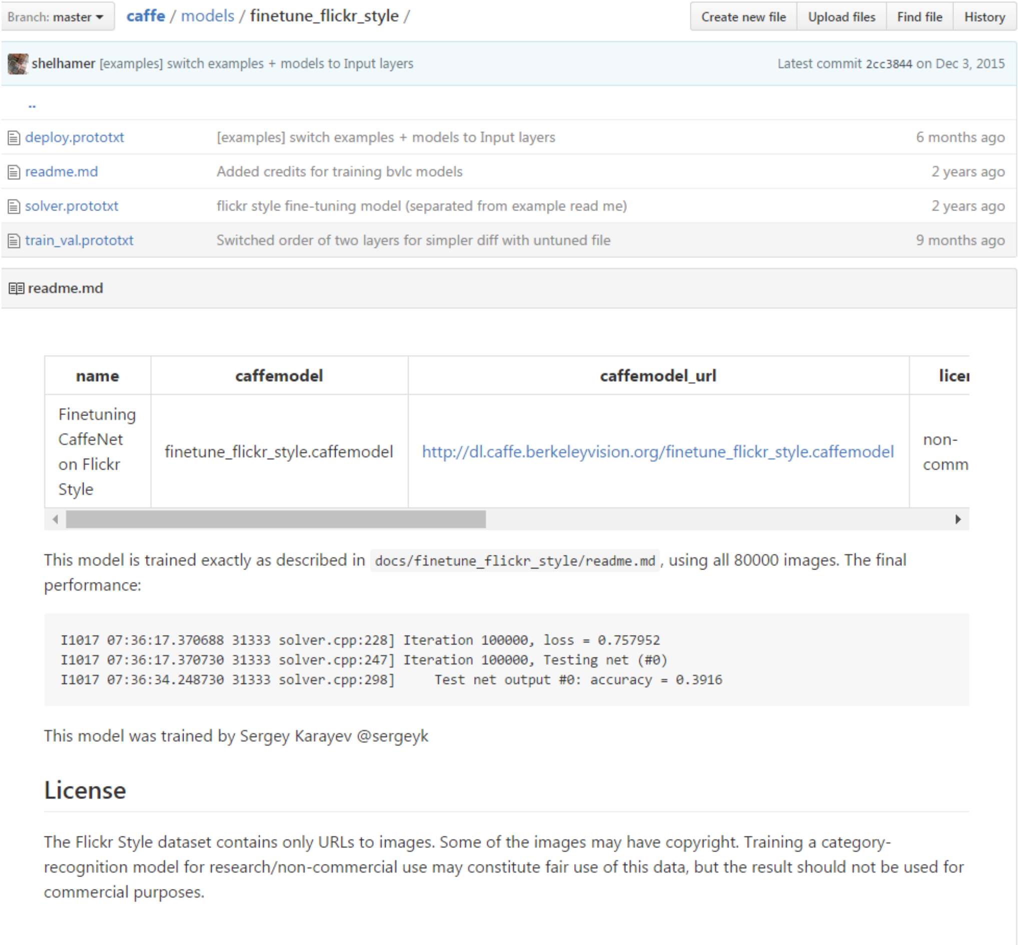
SIEMENS

Network Training

- Fine-tune ***CaffeNet*** pre-trained on Flickr dataset for image style recognition
- Initialize the first two convolutional layers

Sampling Strategy

- Use average rating from users (overfitting)
- Use all ratings from users (too much noise)
- Remove rating outliers (bias)
- Randomly sample user ratings



Branch: master ▾ caffe / models / finetune_flickr_style /

Create new file | Upload files | Find file | History

shelhamer [examples] switch examples + models to Input layers — Latest commit 2cc3844 on Dec 3, 2015

..

deploy.prototxt	[examples] switch examples + models to Input layers	6 months ago
readme.md	Added credits for training bvlc models	2 years ago
solver.prototxt	flickr style fine-tuning model (separated from example read me)	2 years ago
train_val.prototxt	Switched order of two layers for simpler diff with untuned file	9 months ago

readme.md

name	caffemodel	caffemodel_url	licei
Finetuning CaffeNet on Flickr Style	finetune_flickr_style.caffemodel	http://dl.caffe.berkeleyvision.org/finetune_flickr_style.caffemodel	non-comm

This model is trained exactly as described in `docs/finetune_flickr_style/readme.md`, using all 80000 images. The final performance:

```
I1017 07:36:17.370688 31333 solver.cpp:228] Iteration 100000, loss = 0.757952
I1017 07:36:17.370730 31333 solver.cpp:247] Iteration 100000, Testing net (#0)
I1017 07:36:34.248730 31333 solver.cpp:298]     Test net output #0: accuracy = 0.3916
```

This model was trained by Sergey Karayev @sergeyk

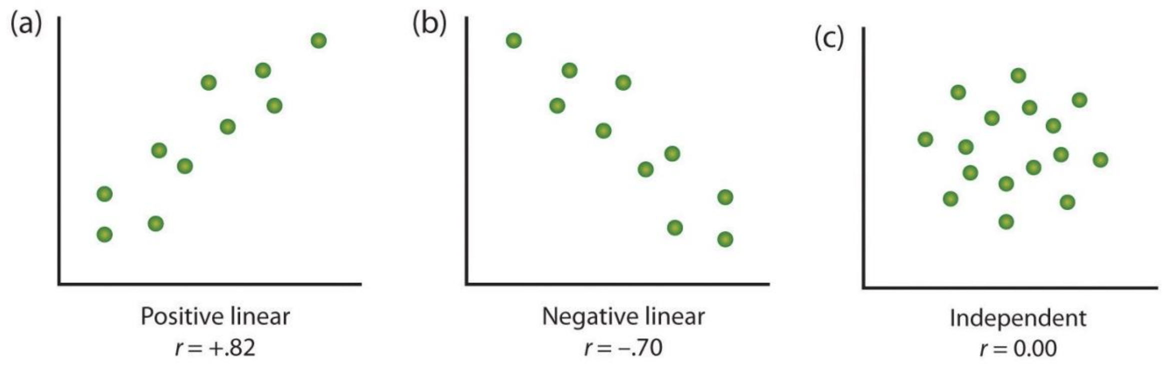
License

The Flickr Style dataset contains only URLs to images. Some of the images may have copyright. Training a category-recognition model for research/non-commercial use may constitute fair use of this data, but the result should not be used for commercial purposes.

Results

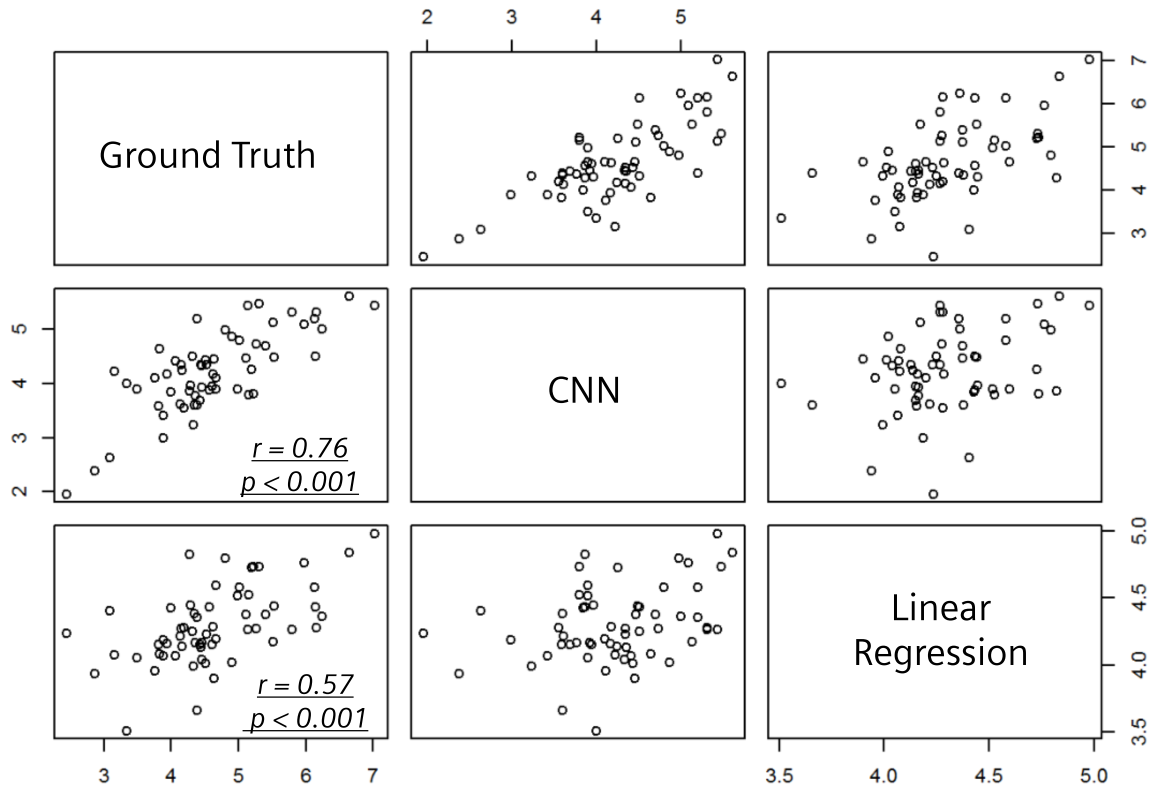
Quantitative Evaluation metric

- Pearson correlation r: measurement of the linear relationship between our prediction results and users' ratings



Results

Scatter plots of prediction results



Results

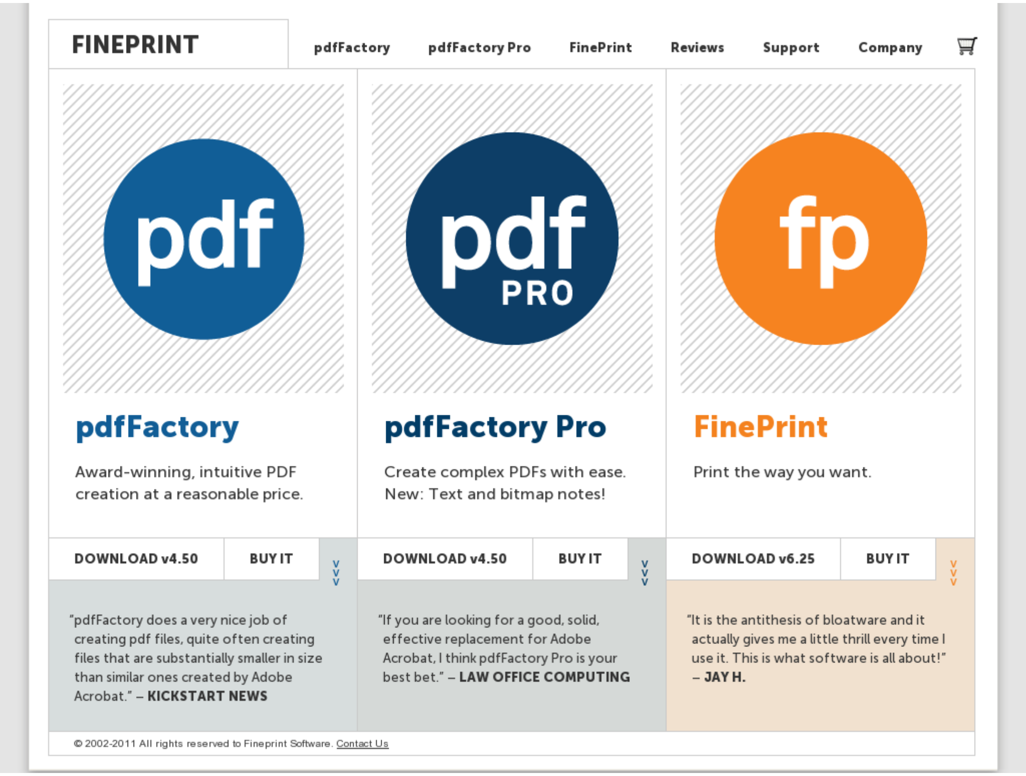
Examples of webpage rating prediction

- Users' rating: 4.44
- CNN prediction: 4.35
- Linear Regression: 4.13

Results

Examples of webpage rating prediction

- Users' rating: 5.14
- CNN prediction: 4.43
- Linear Regression: 4.26

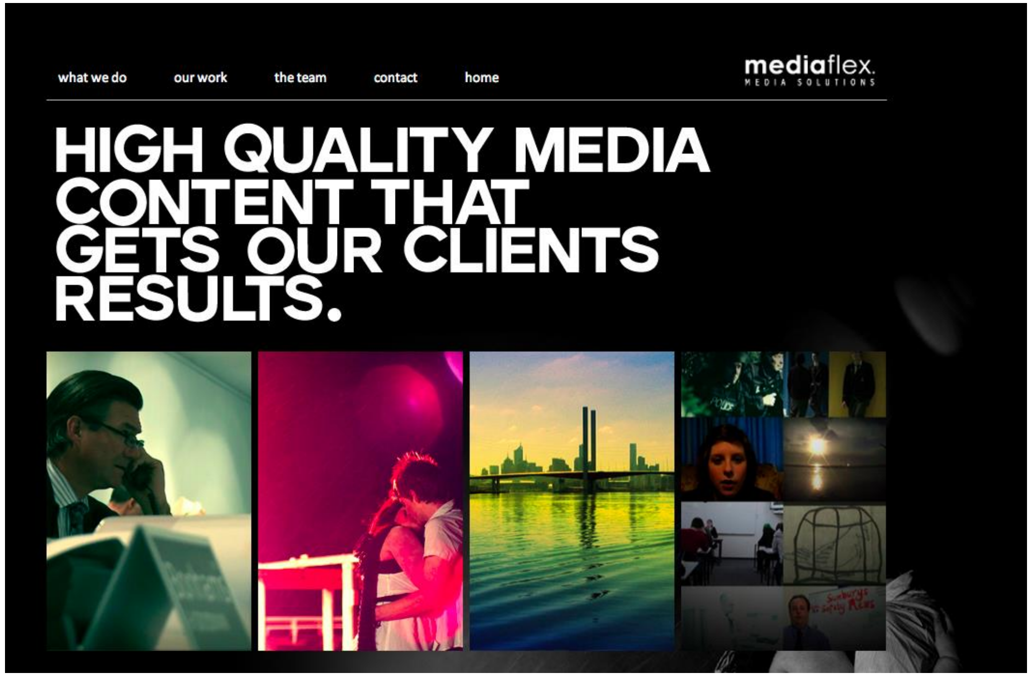


Results

SIEMENS

Examples of webpage rating prediction

- Users' rating: 7.03
- CNN prediction: 5.43
- Linear Regression: 4.98



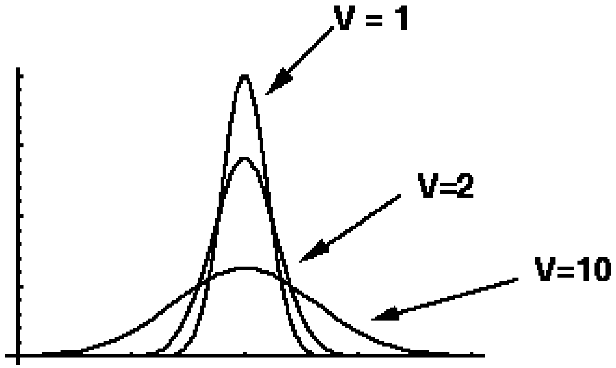
Future

SIEMENS

- Analyze the value of knowledge transfer from network for image style recognition

The application of skills, knowledge, and/or attitudes that were learned in one situation to another **learning** situation (Perkins, 1992)

- Study the rating variance among different user populations
- Equip network with the capability to predict the variance




WORLD OF
WARCRAFT™

World of Warcraft
About

SIEMENS

World of Warcraft (WoW) is a **massively multiplayer online role-playing game** (MMORPG) released in 2004 by Blizzard

- **Third- or first-person view**,
- Exploring the landscape
- Fighting various monsters
- Completing quests,
- **Interacting** with non-player characters (NPCs) or **other players**

World of Warcraft

Zone

World of Warcraft

Race and Career

World of Warcraft

Guild

About

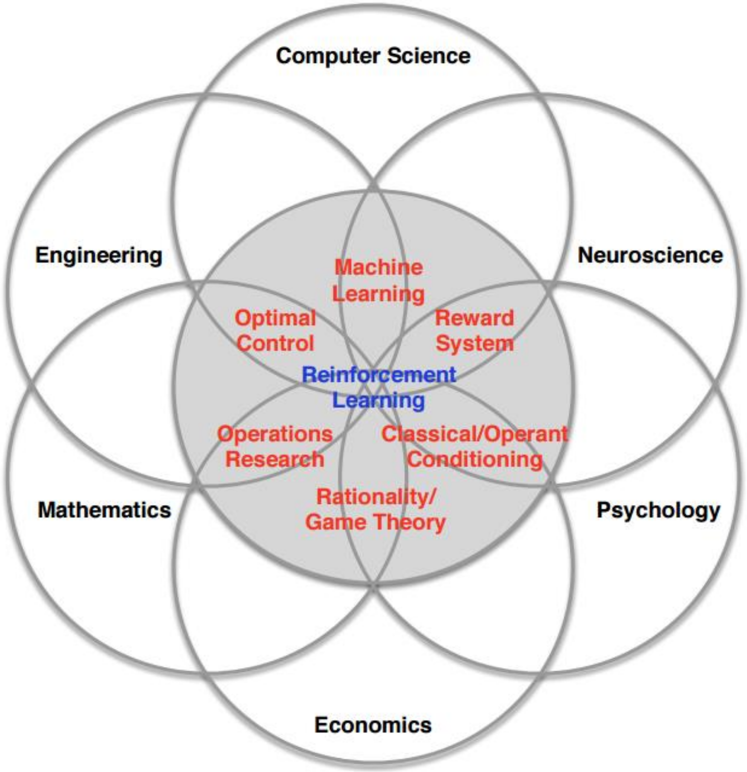
SIEMENS

An agent play with an environment

- To maximize reward
- E.g. maximize user experience, win a Go game

Relation with deep learning

- RL to formulate problem
- DL to solve problem



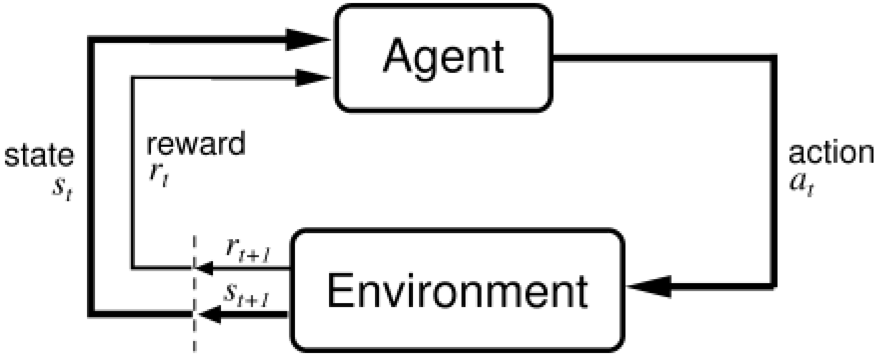
About

SIEMENS

Recent advances in RL

- Mastering game of Go (Deepmind, Nature 2016)
- Playing Atari games (Deepmind, Nature 2015)

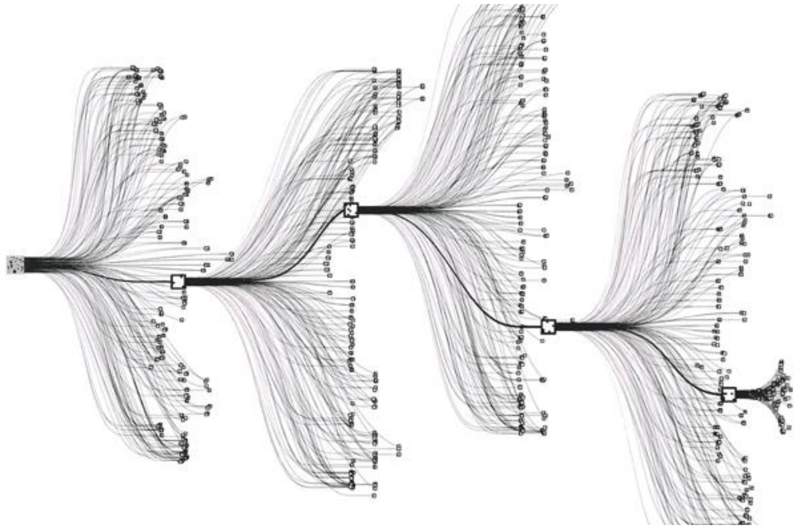
RL for User Behavior Analysis and Design



About

Mastering the Game of Go

- A feat previously thought to be at least a decade way



About

SIEMENS

Mastering RL for UBA and Design

- **We treat user as agent, and try to**
  - Model how users perform actions, i.e. predict the next user action/state
  - Model the “experience” of users, quantitatively, i.e. have a tell of “what is a good experience”
- **To do this, we elaborate several**
  - Deep Q-Networks
  - Inverse Reinforcement Learning

Dataset

Dataset provided by World of Warcraft Avatar History (WoWAH)

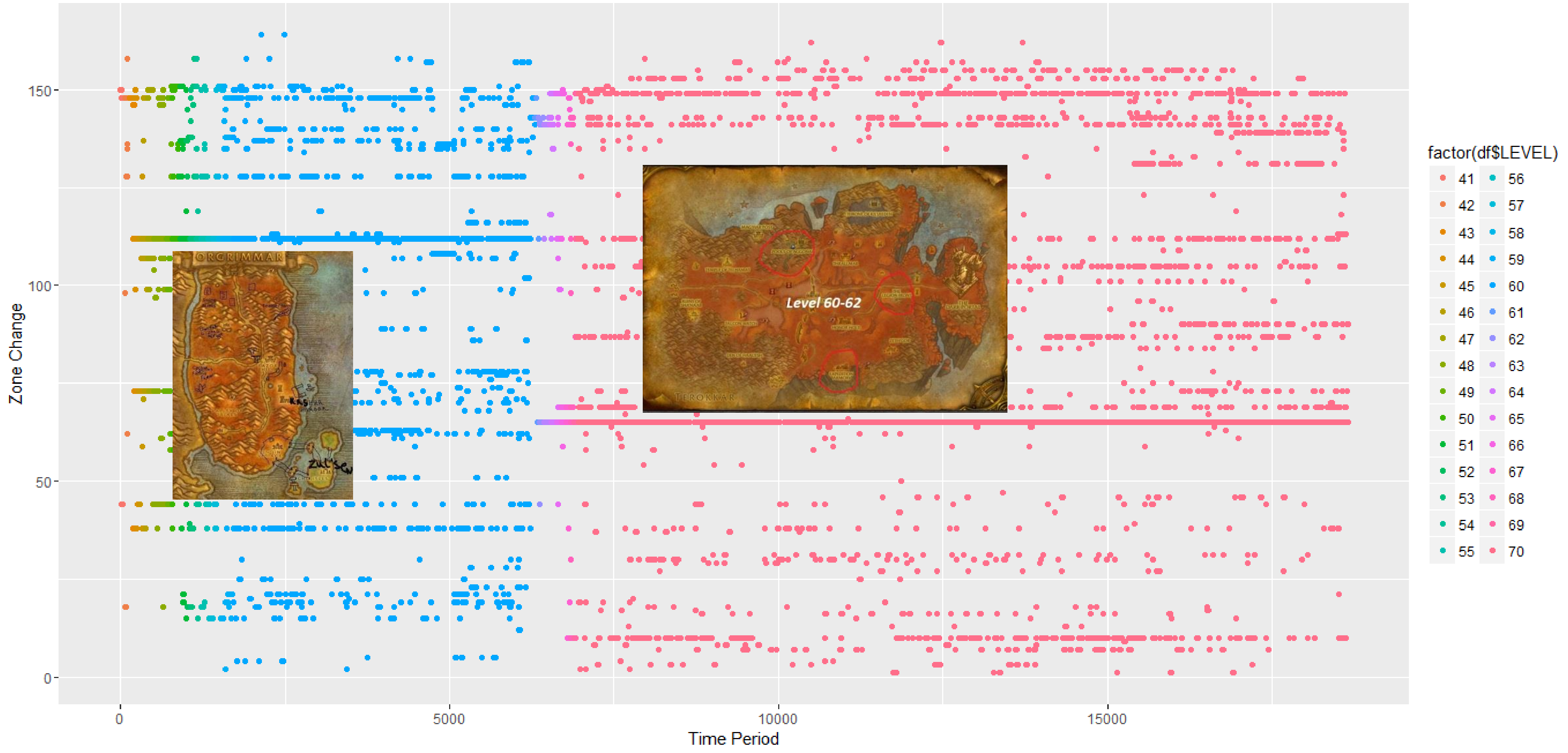
- 70,000 users
- Every 10 mins during 2005-2008
- The current zone (location)
- Other contextual information

Table 2: Field Description

Field	Valid Values
Query Time	Between Jan. 2006 and Jan. 2009
Query Seq. #	An integer ≥ 1
Avatar ID	An integer ≥ 1
Guild	An integer within $[1, 513]$
Level	An integer within $[1, 80]$
Race	Blood Elf, Orc, Tauren, Troll, Undead
Class	Death Knight, Druid, Hunter, Mage, Paladin, Priest, Rogue, Shaman, Warlock, Warrior
Zone	One of the 229 zones in WoW world

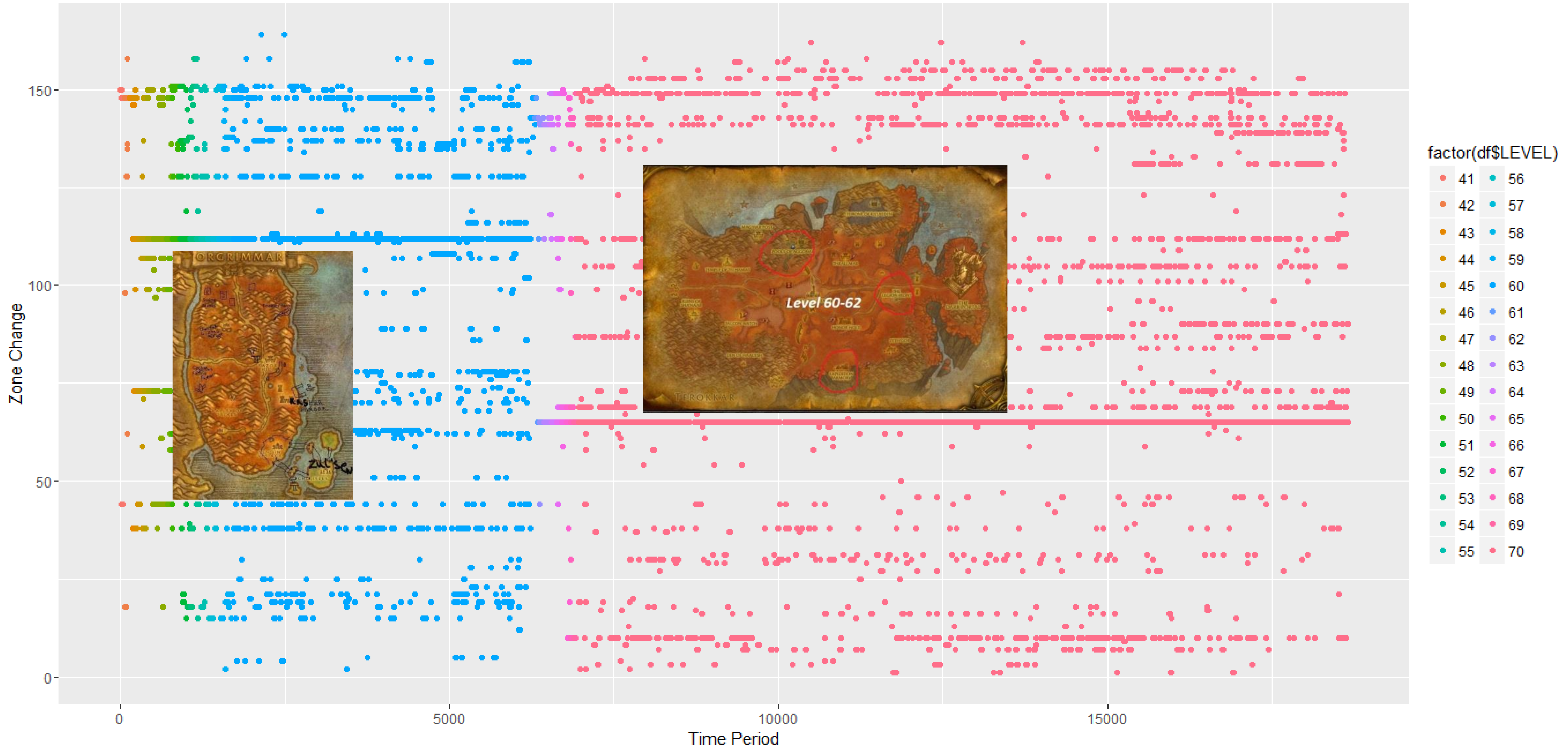
World of Warcraft

Dataset player ID 31

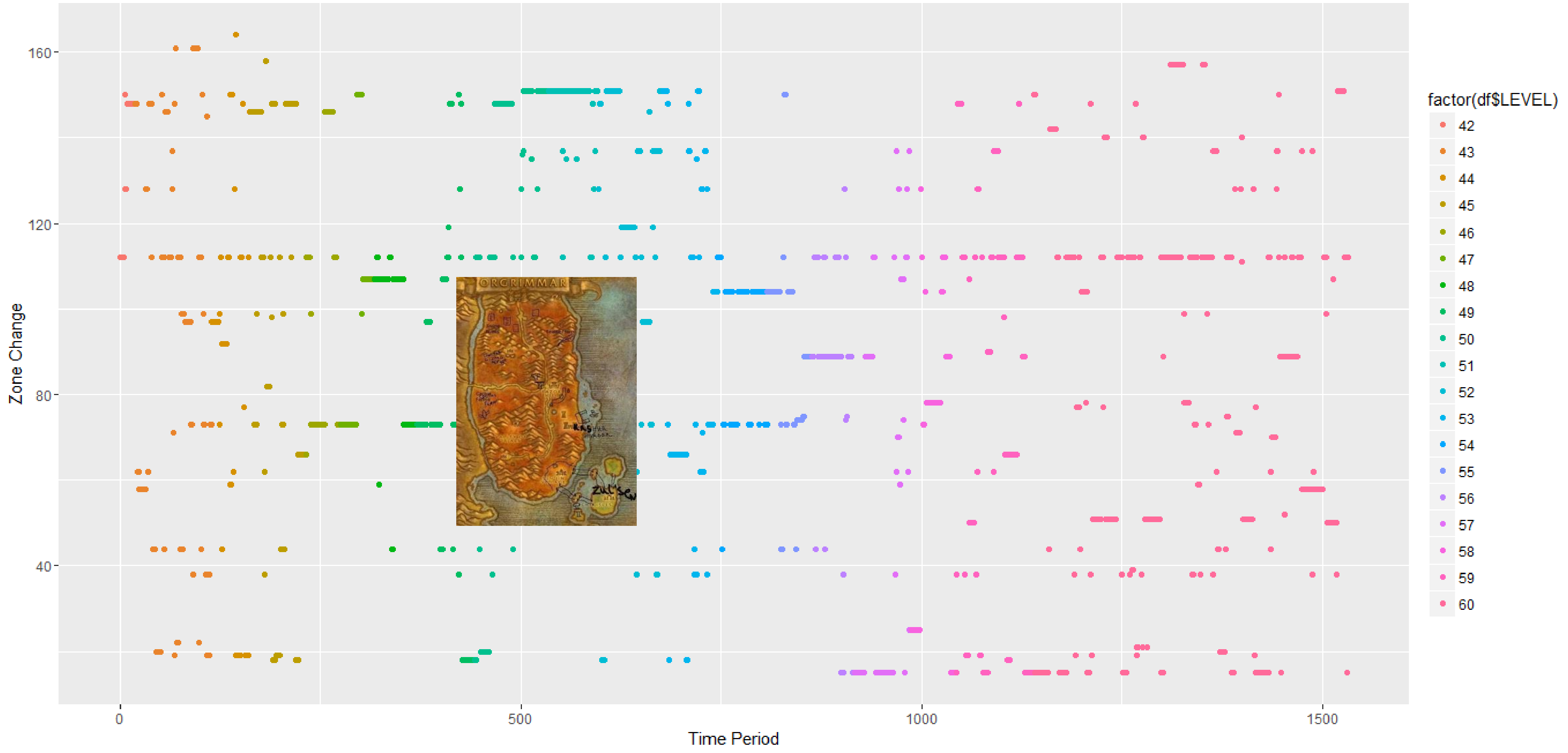


World of Warcraft

Dataset player ID 31



World of Warcraft
Dataset player ID 25



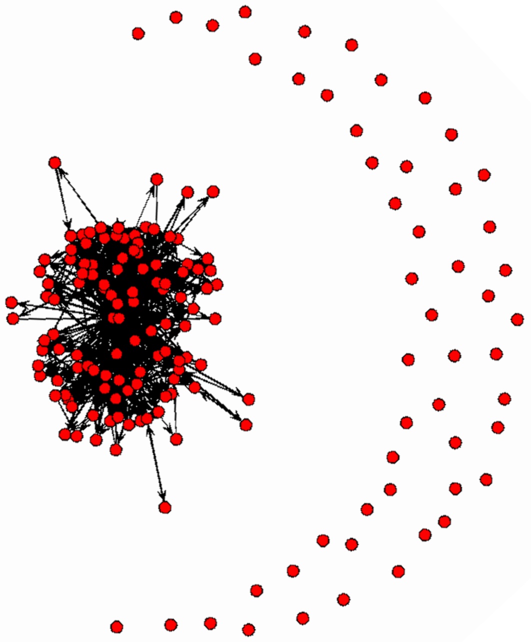
Social Network

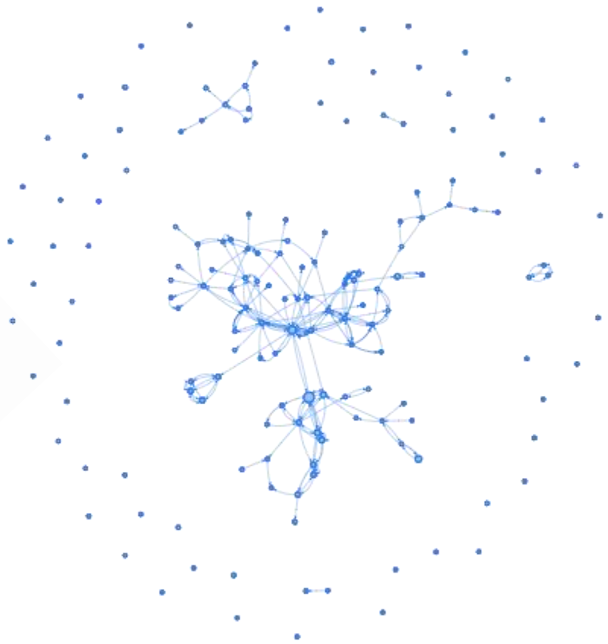
User Behavior Network Analysis

```
> wownet
 Network attributes:
  vertices = 164
  directed = TRUE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
  total edges= 5182
    missing edges= 0
    non-missing edges= 5182

 Vertex attribute names:
    vertex.names

 Edge attribute names not shown
```

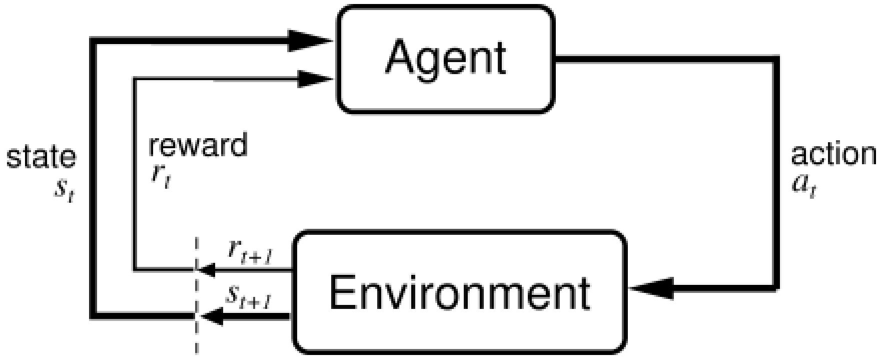




User Behavior Model

RL for user behavior modeling

- We treat user as the agent, who conduct an action a_t every 10 mins;
  - Each a_t corresponds to a zone ID; meaning the user transfer to that zone
  - Time elapse only when user's online
- We treat all recent contextual information as the state s_t

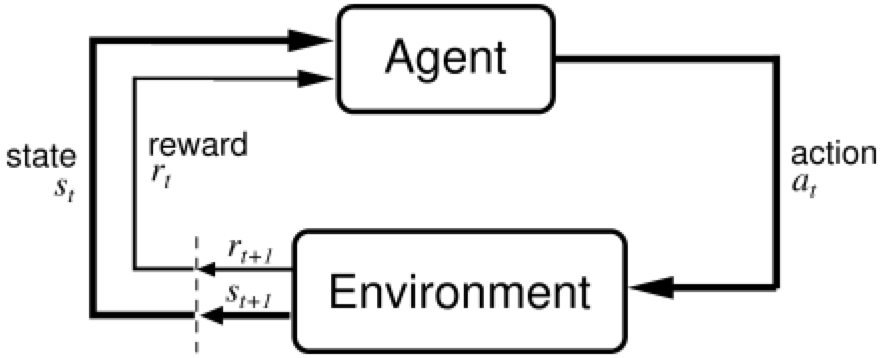


User Behavior Model

SIEMENS

RL for user behavior modeling

- We define $r(s)$ as the reward function, and assume that all users are subject with that
  - E.g. $r(st)$ could be $l(x)$ if the player levels up at time t, from level x
  - We try to model “how users behave” in order to maximize their reward

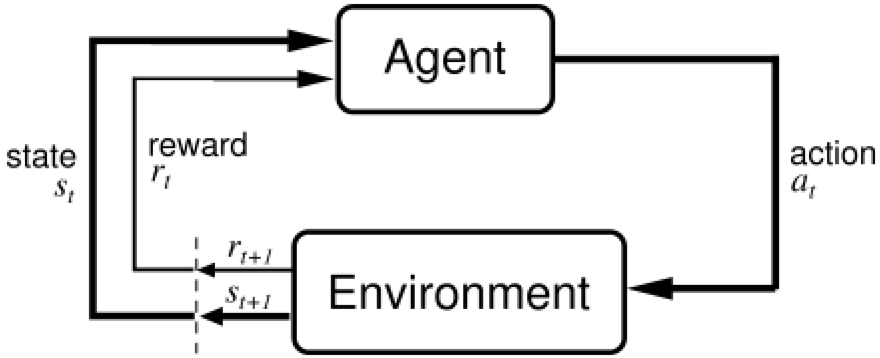


User Behavior Model

SIEMENS

The baseline work is to

- First filter out a bunch of expert players (e.g. use less time to conduct mass level up), and
- Use their trajectory as ground truth, to train a classifier $s \rightarrow a$
- This serves as the baseline to evaluate the quality of policies trained by other algorithms



Training Q-Network

We train a network to approximate the $Q(s,a)$ value, which is defined as

$$Q(s,a) = \sum_t \mathbf{E}[\gamma^t r_t | s_0 = s, a_0 = a]$$

We optimize the network parameter θ in order to make the Q function satisfying Bellman equation

$$Q(s,a) = \gamma r_t + \mathbf{E}_{s_{t+1}}[\max_{a'} Q(s_{t+1}, a')]$$

Training Q-Network

We achieve this by minimizing the square error s.t. bellman equation holds

$$L = (Q(s,a) - \gamma r_t + \mathbf{E}_{s_{t+1}}[\max_{a'} Q(s_{t+1}, a')])^2$$

To do this, we update θ by

$$\theta = \theta - \alpha \frac{\partial L}{\partial \theta}.$$

Training Q-Network

SIEMENS

$$L = (Q(s_t, a) - \gamma r_t + \mathbf{E}_{s_{t+1}}[\max_{a'} Q(s_{t+1}, a')])^2$$

- On every step of training, we choose at random a set of s_t, a, s_{t+1}, r_t, and evaluate the expection by using the s_{t+1} value only (b/c we have no access to the dynamic of the game
- We use rmsprop to adjust α

SIEMENS

Policy Induced by Q-Network

- One way to evaluate the performance of Q-network is to check if it comply with the expert players, i.e. if

$$ {}_{a}Q(s_t, a) = a_t $$

- Using this metric we've got 0.382 as accuracy so far, with the size $|A|$ of the action set being 165

Policy Induced by Q-Network

SIEMENS

- One way to evaluate the performance of Q-network is to check if it comply with the expert players, i.e. if

$$ {}_{a}Q(s_t, a) = a_t $$

- Using this metric we've got 0.382 as accuracy so far, with the size $|A|$ of the action set being 165

Recovering θ

SIEMENS

- Given a set of players we are interested in e.g. experts, entertainers, etc., we try to recover the θ value under which they conducted the behaviors

- Assume they are trying to maximize the reward $\theta^T \varphi(s)$, we find the optimal θ

$$\max_{|\theta|_1=1} \quad C$$

s.t.

$$Q(s, a*) - max_{a \in A - a*} Q(s, a) > C \quad \forall s, a, a*$$