Combining NMF and Regression:

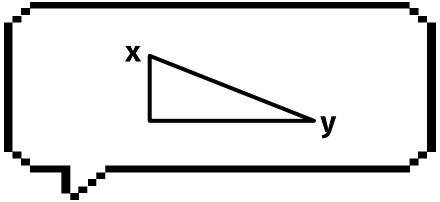
SSNMF

Our Team

Anand Somayajula

Feng Liu

Jenny Ding

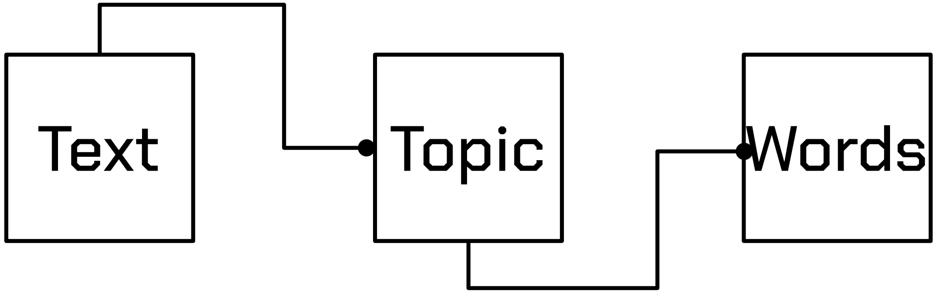


01

Background

A simple introduction to topic modeling and traditional NMF & Regression model

Topic Modeling




```
For topic 1 the words with the highest value are:
film               1.619832
viewers            0.374068
audience           0.335383
performance        0.321071
powerful           0.315721
best               0.282919
ill                0.282483
entertaining       0.282339
year               0.281354
directed           0.265265
Name: 0, dtype: float64
```

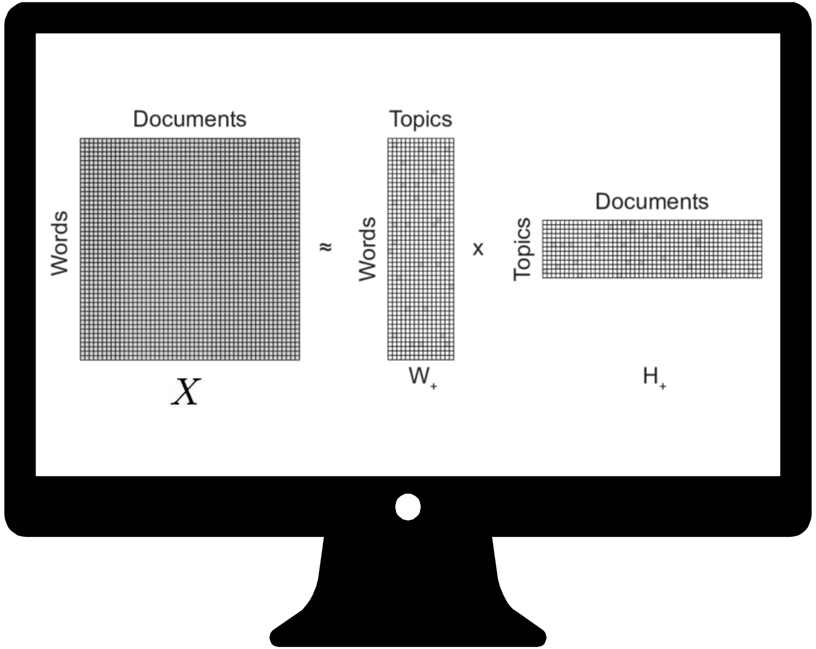
Non-negative Matrix Factorization

Let $W \in \mathbb{R}_{\geq 0}^{n \times k}$ (basis matrix) and $H \in \mathbb{R}_{\geq 0}^{k \times m}$ (coefficient matrix) be an NMF decomposition such that:

$$X \approx WH$$

To find W and H, we will try to minimize:

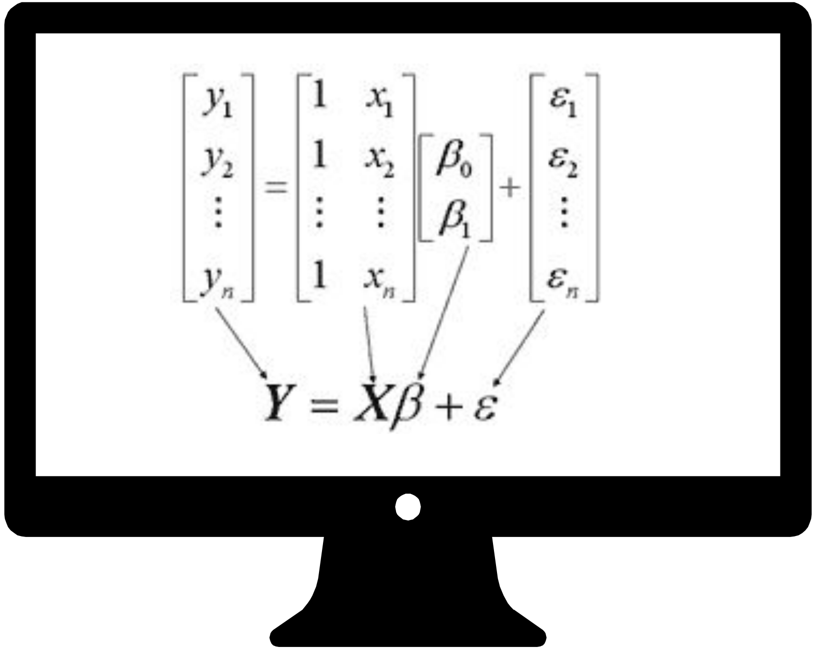
$$||X - WH||_F^2$$



Linear Regression

Model the relationship between two variables by fitting a linear equation to observed data.

$$\hat{Y}_i = \theta_0 + w_{i1}\theta_1 + ... + w_{ik}\theta_k$$



Objective

Combine NMF and Regression Algorithms into an SSNMF (Semi-Structured Non-negative Matrix Factorization)

“This is a very basic stand. I bought 2, one for my desk at work and one for my end table by the bed. It's inexpensive and does the only thing I need it to do, hold my phone up. When it's charging I turn it landscape. Bought one for my husband and he is happy with it as well."

02

Method

404 NOT FOUND

Our derivation for SSNMF

NMF, Then Regression

First [NMF]: $X \approx WH$

Objective Function:

$$||X - WH||_F^2$$

Second [Regression]:find θ such that $\theta \in \mathbb{R}^{k+1}$

Objective Function:

$$\theta = argmin_\theta||\widetilde{W}\theta - Y||^2$$

SSNMF

Combining two steps together with a weighting coefficient λ and do the gradient descent to optimize the new objective function.

Objective Function:

$$F = ||X - WH||_F^2 + \lambda||\widetilde{W}\theta - Y||_F^2$$

Objective Function

$$F = ||X - WH||_F^2 + \lambda||\widetilde{W}\theta - Y||_F^2$$

X : documents as rows and words as columns(positive matrix)

Y : measure scores

λ : weighting coefficient between [0, +∞)

$$\widetilde{w} = \left[\begin{array}{cc} 1 & \\ 1 & w \\ 1 & \end{array}\right]$$

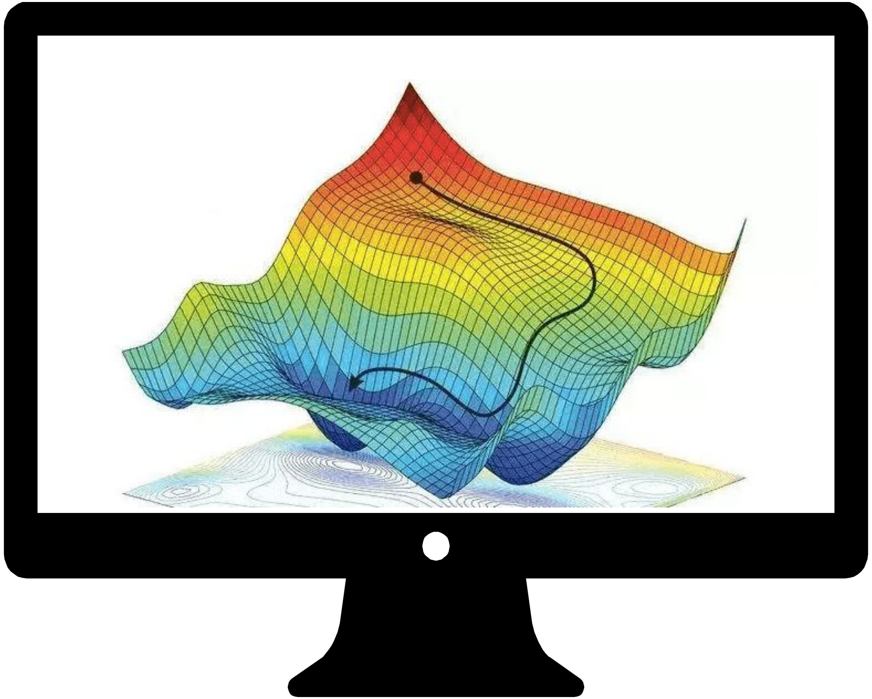
$$\hat{Y}_i = \theta_0 + w_{i1}\theta_1 + ... + w_{ik}\theta_k$$

$$W \in \mathbb{R}_{\geq 0}^{n \times k}$$

$$H \in \mathbb{R}_{\geq 0}^{k \times m}$$

$$\theta \in \mathbb{R}^{k+1}$$

Method 1: Gradient Descent



$$\boxed{F = ||X - WH||_F^2 + \lambda||\widetilde{W}\theta - Y||_F^2}$$

Method 1: Gradient Descent

H

Grad Descent:

$$\nabla_H = -2(W^T X) + 2(W^T W)H$$

Updates:

$$H^{t+1} = H^t - \eta_H (W^T W H - W^T X)$$

Step size:

$$\eta_H = \frac{H}{W W^T H}$$

W

Grad Descent:

$$\nabla_W = -XH^T + WHH^T - \lambda Y \hat{\theta}^T + \theta_0 \lambda 1_k \hat{\theta}^T + \lambda W \hat{\theta} \hat{\theta}^T$$

Updates:

$$W^{t+1} = W^t + \eta_W \cdot (XH^T) - \eta_W \cdot [WHH^T - \lambda Y \hat{\theta}^T + \theta_0 \lambda 1_k \hat{\theta}^T + \lambda W \hat{\theta} \hat{\theta}^T]$$

Step size:

$$\eta_W = \frac{W}{WHH^T + (\lambda W \hat{\theta} \hat{\theta}^T - \lambda Y \hat{\theta}^T + \theta_0 \lambda 1_k \hat{\theta}^T)_+}$$

$$\text{where } (n)_+ = \begin{cases} u, u > 0 \\ 0, u \leq (0) \end{cases}$$

θ

Grad Descent:

$$\nabla_\theta = -2(\widetilde{W}^T \widetilde{W} \theta) + 2(\widetilde{W}^T Y)$$

Updates:

$$\theta^{t+1} = \theta^t - \eta_\theta (\widetilde{W}^T Y - \widetilde{W}^T \widetilde{W} \theta)$$

Step size:

$$\eta_\theta = \alpha_0 \Gamma^n \ (for\ 0 < \Gamma < 1)$$

Method 2: NLS

$$X \approx W H$$

$$H = \begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_m \\ | & | & & | \end{bmatrix}$$

$$X \approx \begin{bmatrix} | & | & | & & | \\ Wh_1 & Wh_2 & Wh_3 & \cdots & Wh_m \\ | & | & | & & | \end{bmatrix}$$

$$\|X - WH\|_F^2 = \sum_{j=1}^{m} \|x_j - Wh_j\|^2$$

Method 2: NLS

H

Optimize columns over H

$$H_{:j} = argmin_h ||Wh - X_{:j}||^2$$

$h \in R^k$

j:[1,m]

W

Optimize rows over W

$$W_{i:} = argmin_w ||\widetilde{X}_{i:}^T - \widetilde{H}^T w^T||^2$$

$w \in R^k$

i:[1,n]

θ

Optimize over θ

$$\theta = argmin_z ||Y - \widetilde{W} z||^2$$

$z \in R^{k+1}$

03

Proof of Concept

Test our algorithm on small sample matrix

We generate an randomized matrix to see if our algorithm works.

We manually set W, H and theta and generates the true X and true Y by:

$$X \approx WH \qquad\qquad Y = \theta_0 + W\theta[1:]$$

Size of testing data:

$$X \in \mathbb{R}^{5\times 4}, W \in \mathbb{R}^{5\times 2}, H \in \mathbb{R}^{2\times 4}, \theta \in \mathbb{R}^{3\times 1}$$

Actual value of X and Y vs Algorithm Calculations

```
X_true:
 [[ 30.3833  71.2034  47.8161  27.8209]
 [ 75.5949 163.3844 124.6938  86.3214]
 [ 57.2648 131.1791  91.3771  56.1867]
 [ 51.5397 103.3955  88.3396  68.7844]
 [ 58.952  129.5854  96.3386  64.6205]]
```

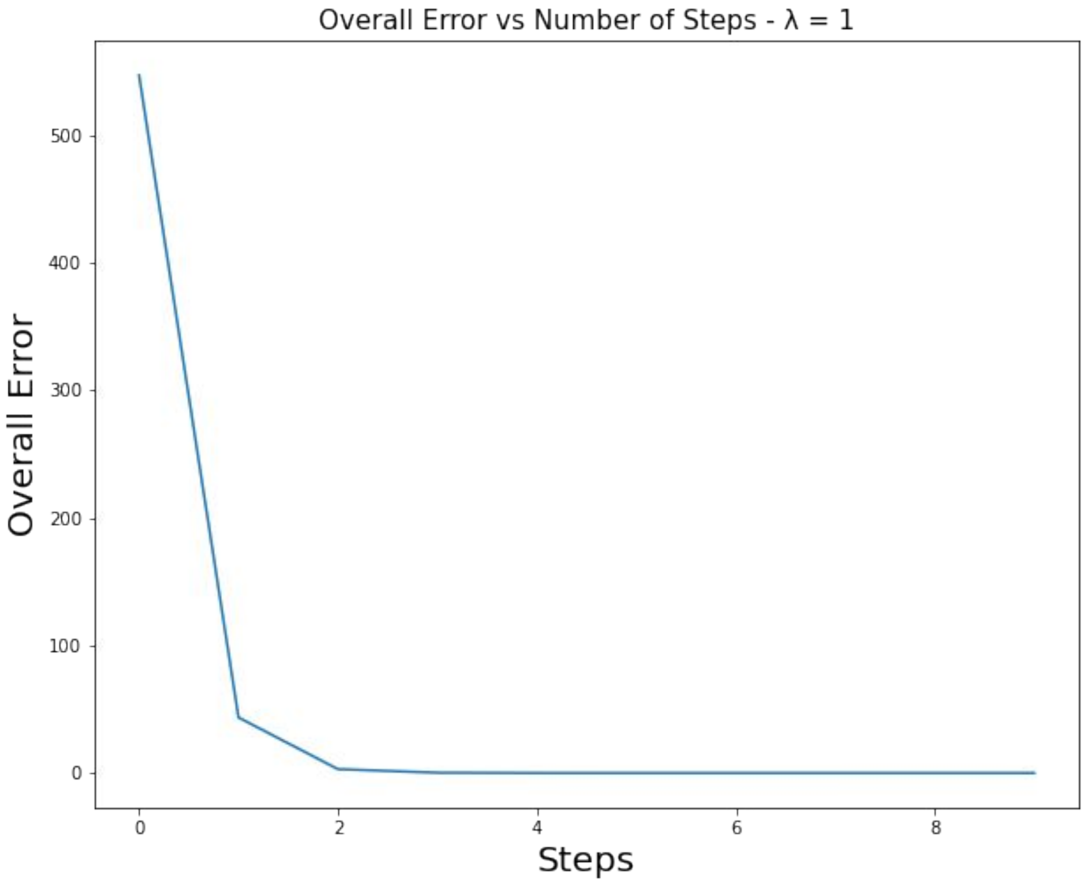
```
Y_true:
 [[-27.5813]
 [-62.4906]
 [-52.7382]
 [-34.5973]
 [-49.7541]]
```

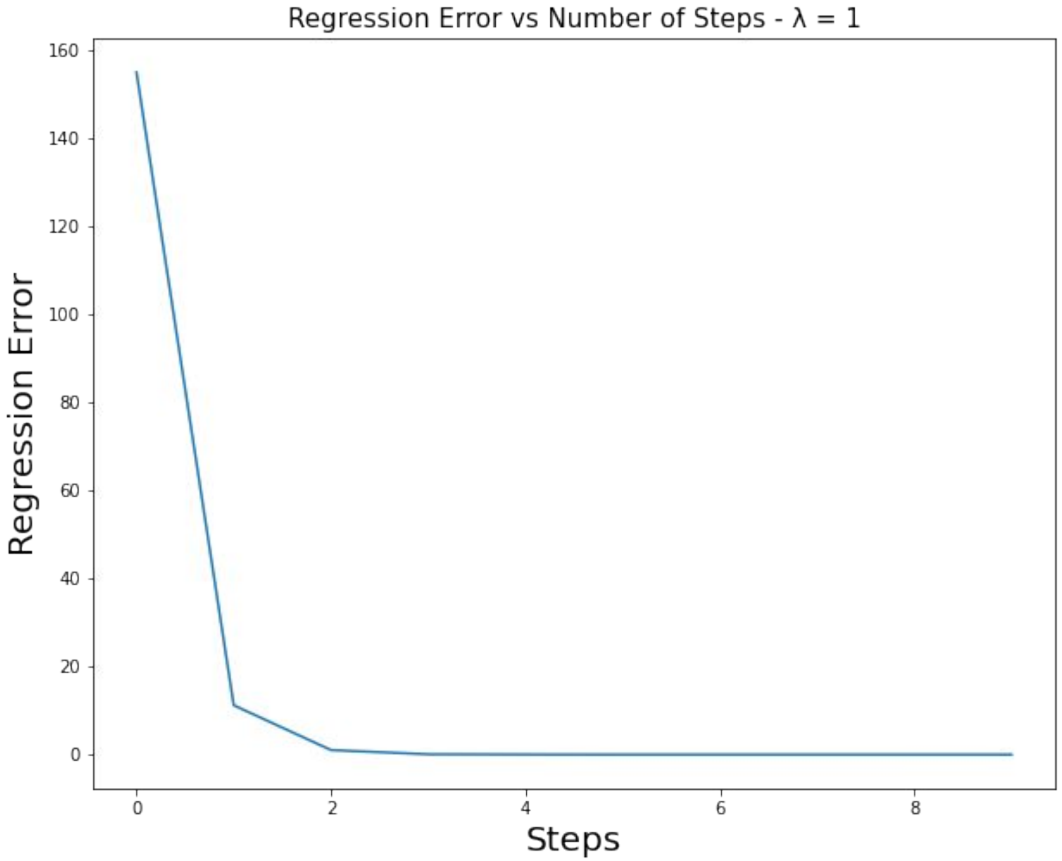
$$X \approx WH$$

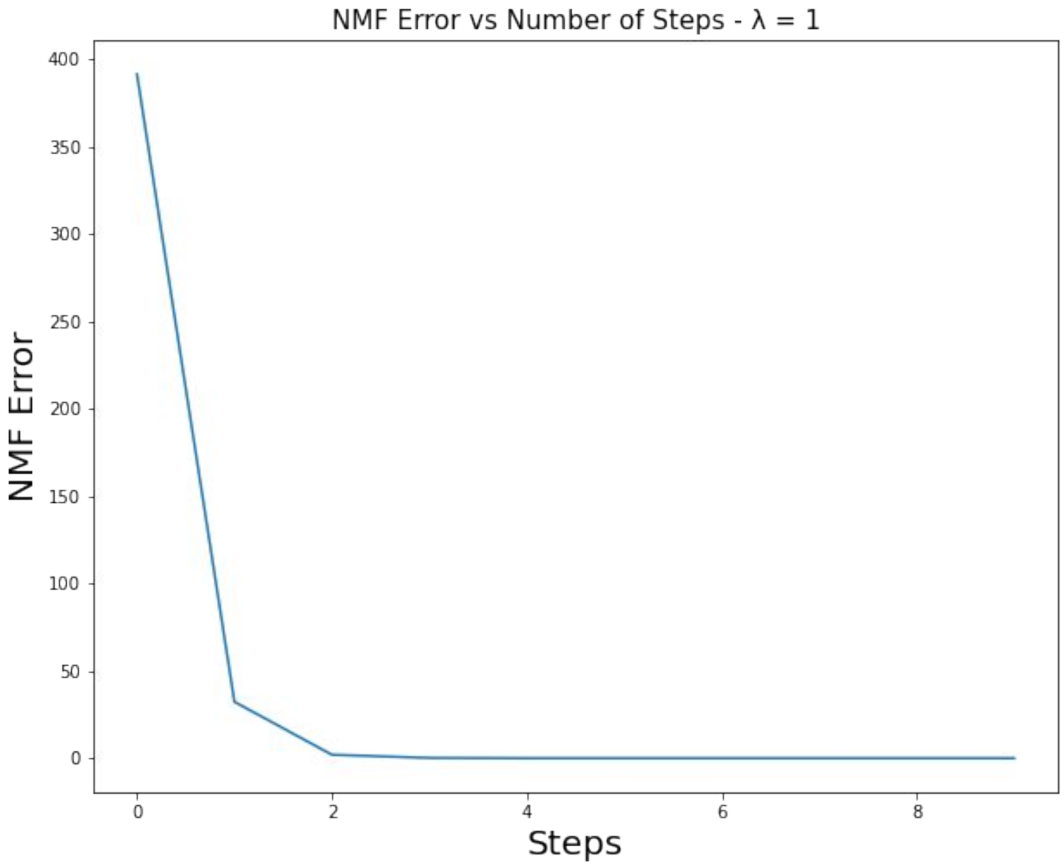
```
array([[ 30.3838,  71.2058,  47.8162,  27.8195],
       [ 75.5946, 163.3827, 124.6937,  86.3223],
       [ 57.2648, 131.1787,  91.3771,  56.187 ],
       [ 51.5399, 103.3969,  88.3397,  68.7836],
       [ 58.952 , 129.5853,  96.3386,  64.6206]])
```

$$Y \approx \widetilde{W}\theta$$

```
array([[-27.5837],
       [-62.489 ],
       [-52.7379],
       [-34.5986],
       [-49.754 ]])
```

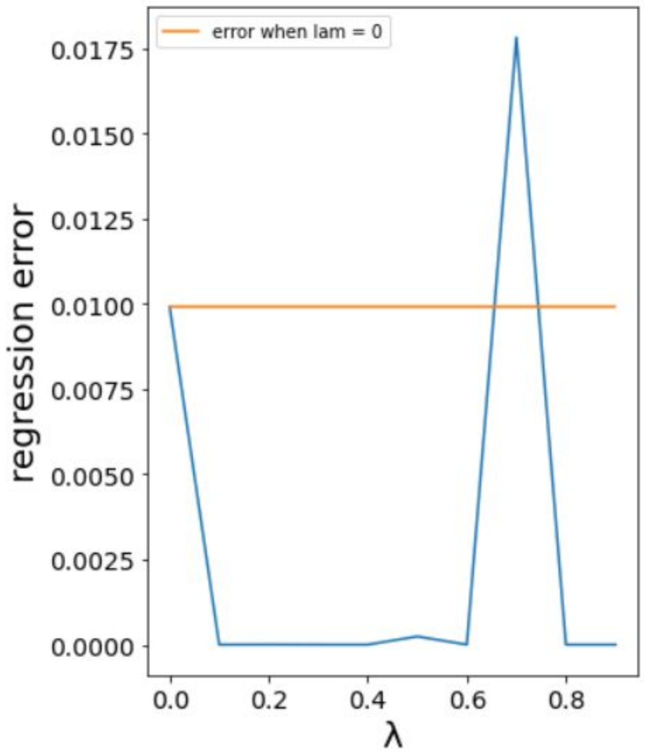

Overall Error vs Number of Steps - λ = 1
Overall Error
Steps
0
100
200
300
400
500
0
2
4
6
8

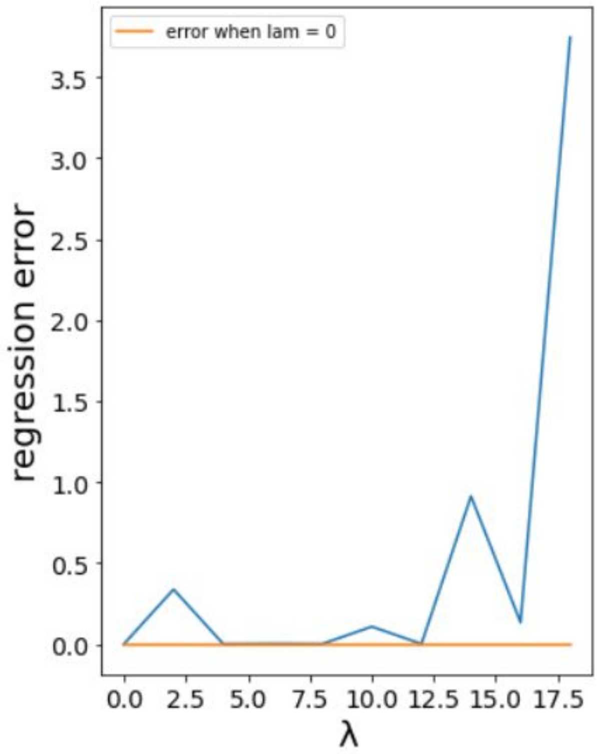

Regression Error vs Number of Steps - λ = 1
Regression Error
Steps
0
20
40
60
80
100
120
140
160
0
2
4
6
8



NMF Error vs Number of Steps - λ = 1
400
350
300
250
200
150
100
50
0
NMF Error
0
2
4
6
8
Steps

When SSNMF performs better than NMF





04

Model Application

SSNMF application on Amazon Reviews

Amazon Review

	overall	reviewText
0	4	it's fine. I just would like the stickers to b...
1	2	took me three returns to get one that didn't w...
2	2	While the product is fine the description and ...
3	5	It's beautiful and blends right in with my woo...
4	5	I love this stand! I had been looking around f...
...	...	...
495	5	This compact vacuum that can carry around it'...
496	5	Everyone in my family is jealous I have a vacu...
497	1	Possibly worst product I've purchased on Amazo...
498	5	Perfect little guy for my 2014 Nissan altima
499	2	The unit picks up loose dirt ok, but does a po...

Amazon Review

500 reviews of Automotive Products
Processed using TFIDF package. Resultant matrix X is of the form:

$$X = \begin{array}{cc} & \begin{array}{ccccc} word_1 & . & . & . & word_m \end{array} \\ \begin{array}{c} doc_1 \\ . \\ . \\ doc_n \end{array} & \begin{pmatrix} f_{11} & . & . & . & f_{1m} \\ . & . & . & . & . \\ . & . & . & . & . \\ f_{n1} & . & . & . & f_{nm} \end{pmatrix} \end{array}$$

Result from simple NMF and then regression

```
print(runner.theta)
```

```
[[ 2.61332157]
 [ 0.61294417]
 [ 0.59445488]
 [ 0.76496062]
 [-0.23248215]
 [ 0.37547377]
 [-0.34527982]
 [ 0.48460611]
 [-0.13392346]
 [ 0.53645556]
 [ 0.24891794]]
```

Positive

For topic 2 the words with the highest value are:

good	0.130397
product	0.030242
vacuum	0.022363
cleaner	0.019715
cleaning	0.019453
small	0.018684
workmanship	0.018300
practical	0.018016
having	0.017827
price	0.017792

Negative

For topic 6 the words with the highest value are:

suction	0.043123
just	0.019656
ok	0.017143
work	0.016195
little	0.015311
item	0.014352
doesn't	0.013533
poor	0.012423
strong	0.012295

SSNMF Result

```
print(runner.theta)
```

```
[[-1.38991622e+00]
 [-2.66936981e+02]
 [ 1.36664860e+00]
 [ 2.64758453e+04]
 [-3.41571797e-02]
 [-7.72471333e+02]
 [-1.14904817e+02]
 [ 6.14965099e+00]
 [-8.30492224e+01]
 [ 2.51450747e+00]
 [-5.06044143e+02]]
```

Positive:

For topic 9 the words with the highest value are:

powerful	0.237815
perfect	0.043648
handy	0.031360
vacuum	0.030418
wish	0.025918
recommend	0.020057
light	0.014846
compact	0.013682
gets	0.011006
boat	0.010337

Negative:

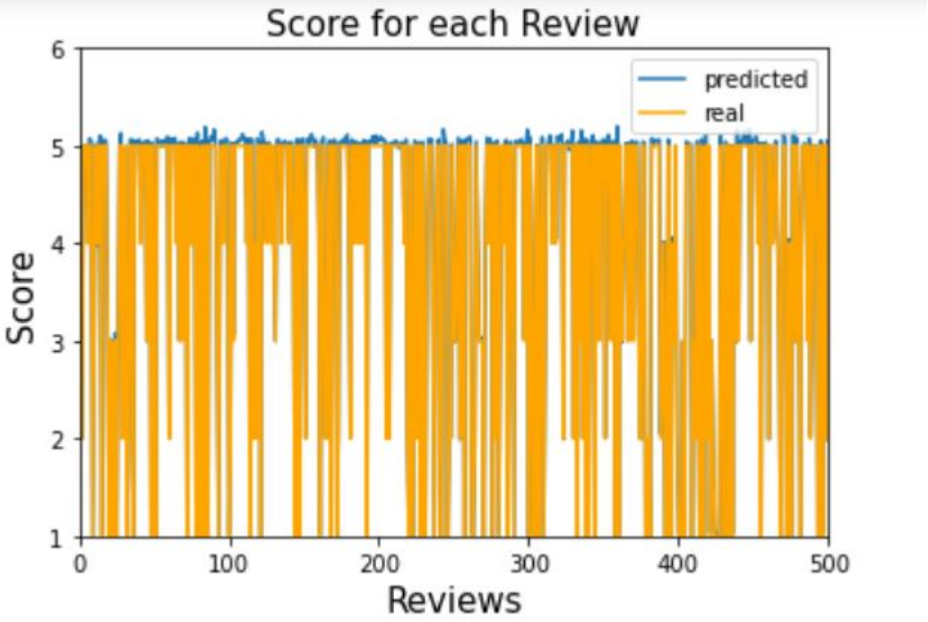
For topic 1 the words with the highest value are:

power	0.144965
suction	0.092796
weak	0.024590
terrible	0.014330
lot	0.013125
wish	0.013106
poor	0.012410
barely	0.011033
returned	0.011029
strong	0.010738

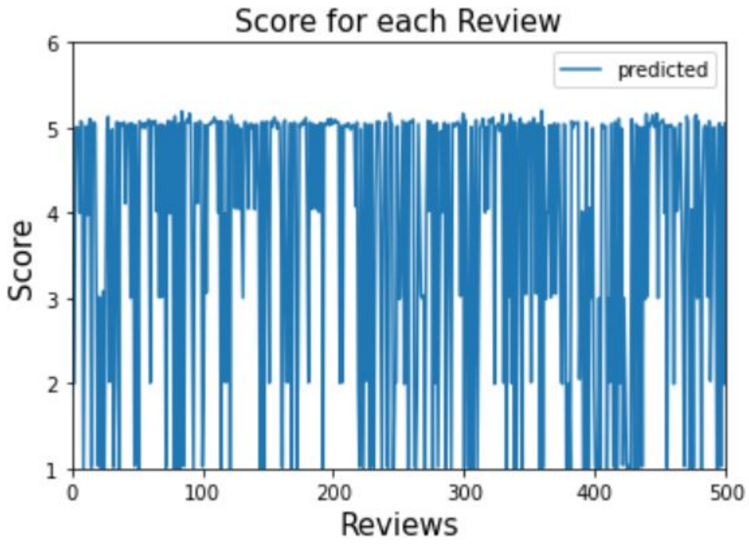
SSNMF result Graph

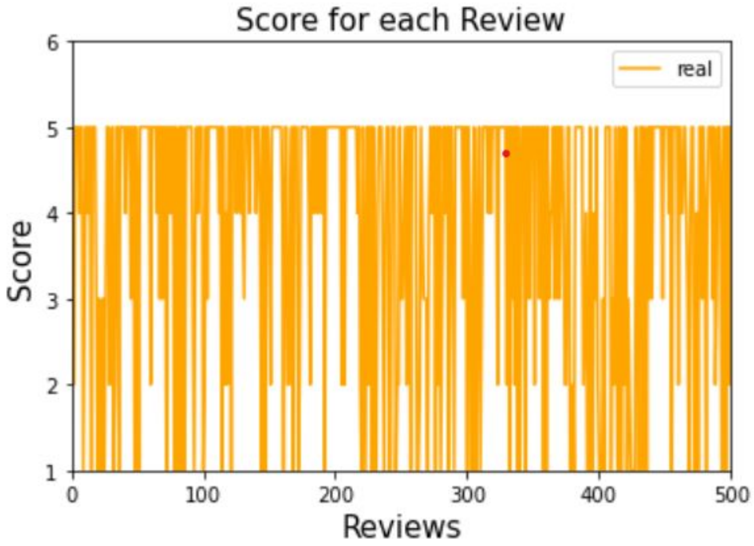
Lambda = 1
Sample: first 500 reviews
Iterations: 1000

Observation:
1. True Y and predict Y have similar shape
2. There's no constraint for prediction



SSNMF Results



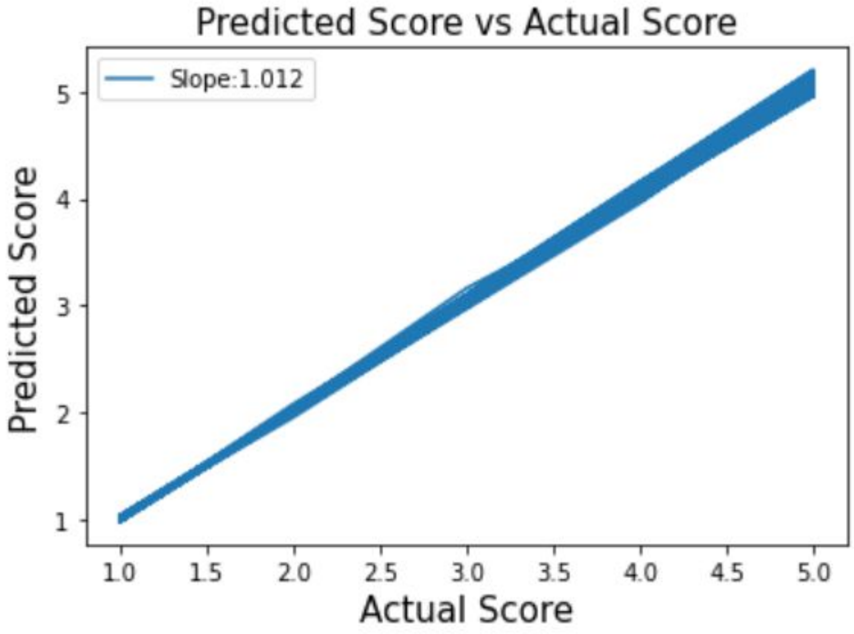


Predicted Score vs Actual Score

Sample: first 500 reviews
Iteration: 1000

Future Improvement:

1. Increase the number of iterations
2. Enlarge sample size

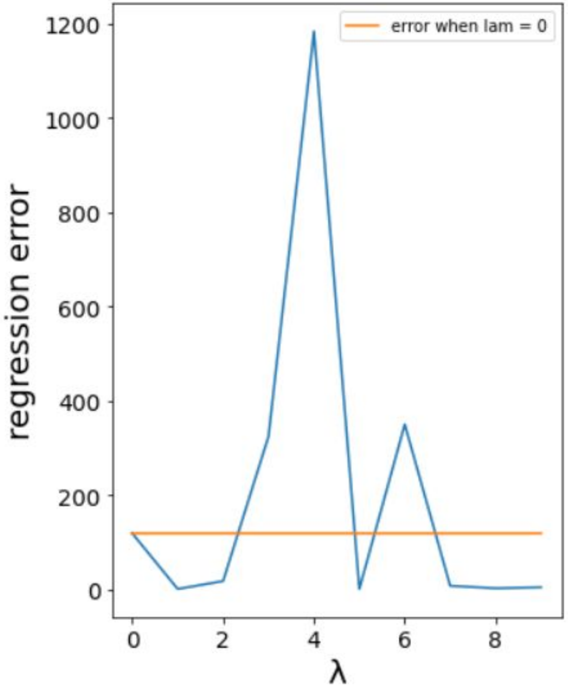


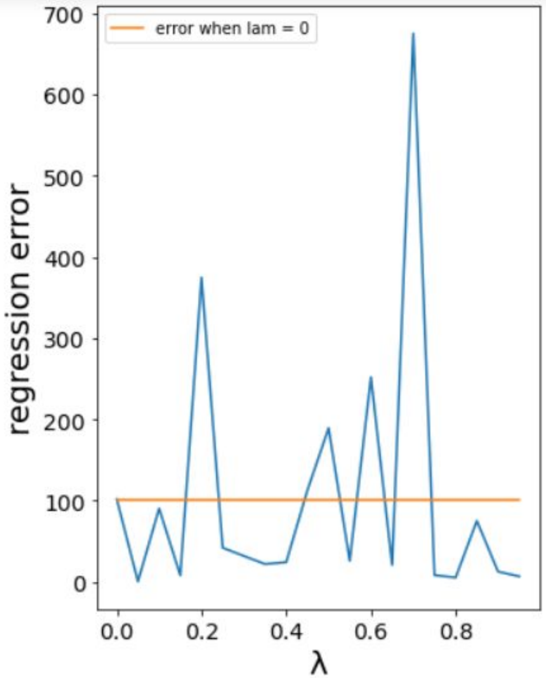
Regression error over lambda

100 Iterations

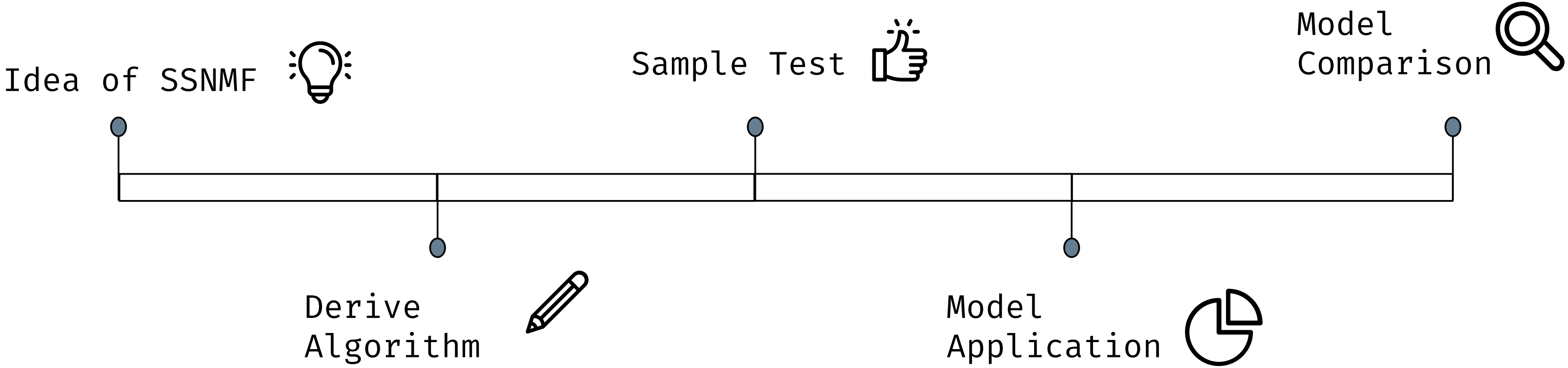
Similar idea as with test dataset

The real Amazon data needs more iterations





Conclusion


Idea of SSNMF
Derive Algorithm
Sample Test
Model Application
Model Comparison

Future Work

1. Finding an a general property of lambda on which SSNMF outperforms NMF method (on real dataset) .
2. Validation - Given a review, decompose it into topics and then make a prediction.
3. Identify the suitable number of topics for a dataset.
4. Use GPU for more iterations (real-world data takes a long time to run)

Acknowledgements

Special thanks to:

Professor Lindstrom who directed and supervised us throughout the research process

UCLA math department for offering this class