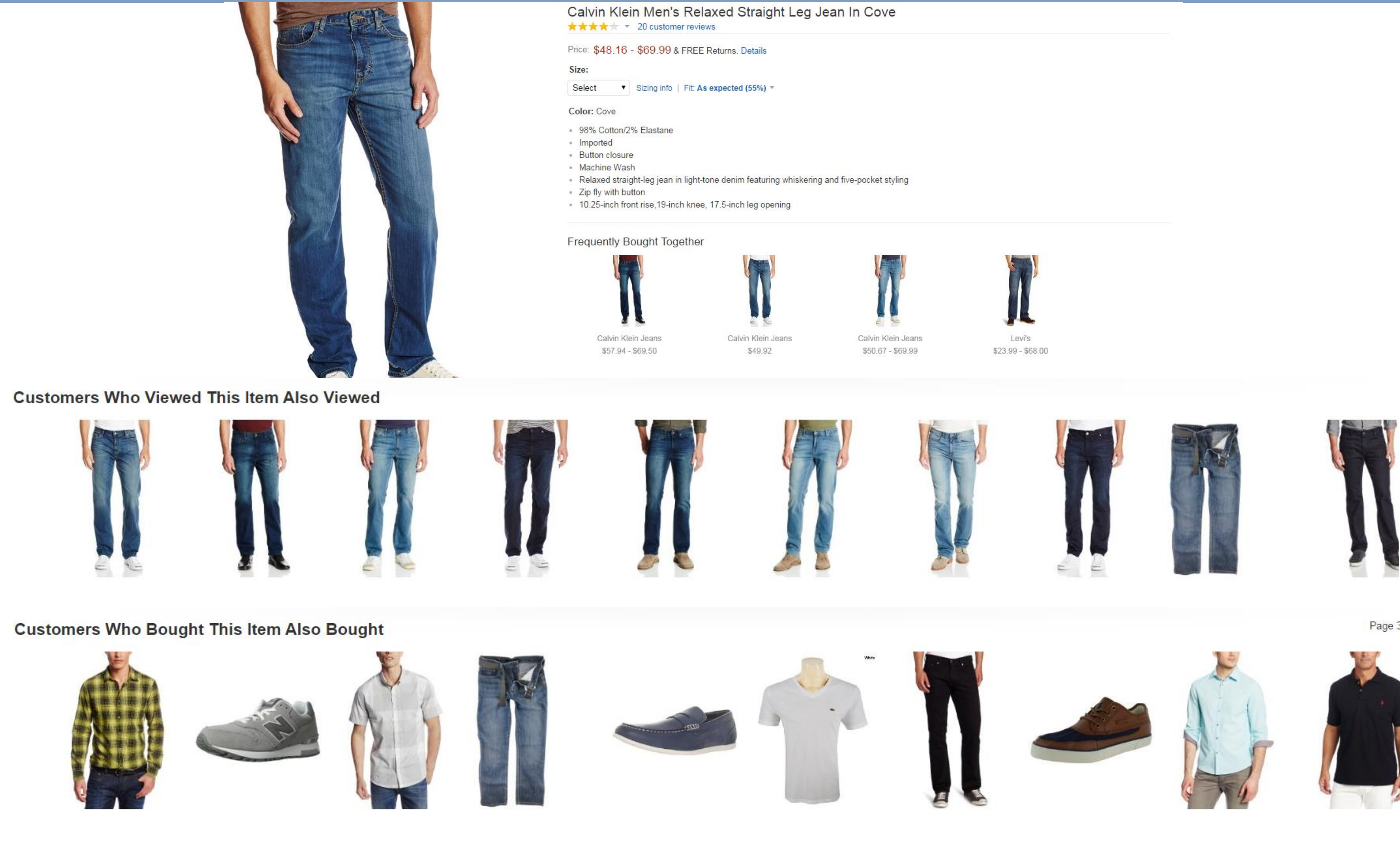
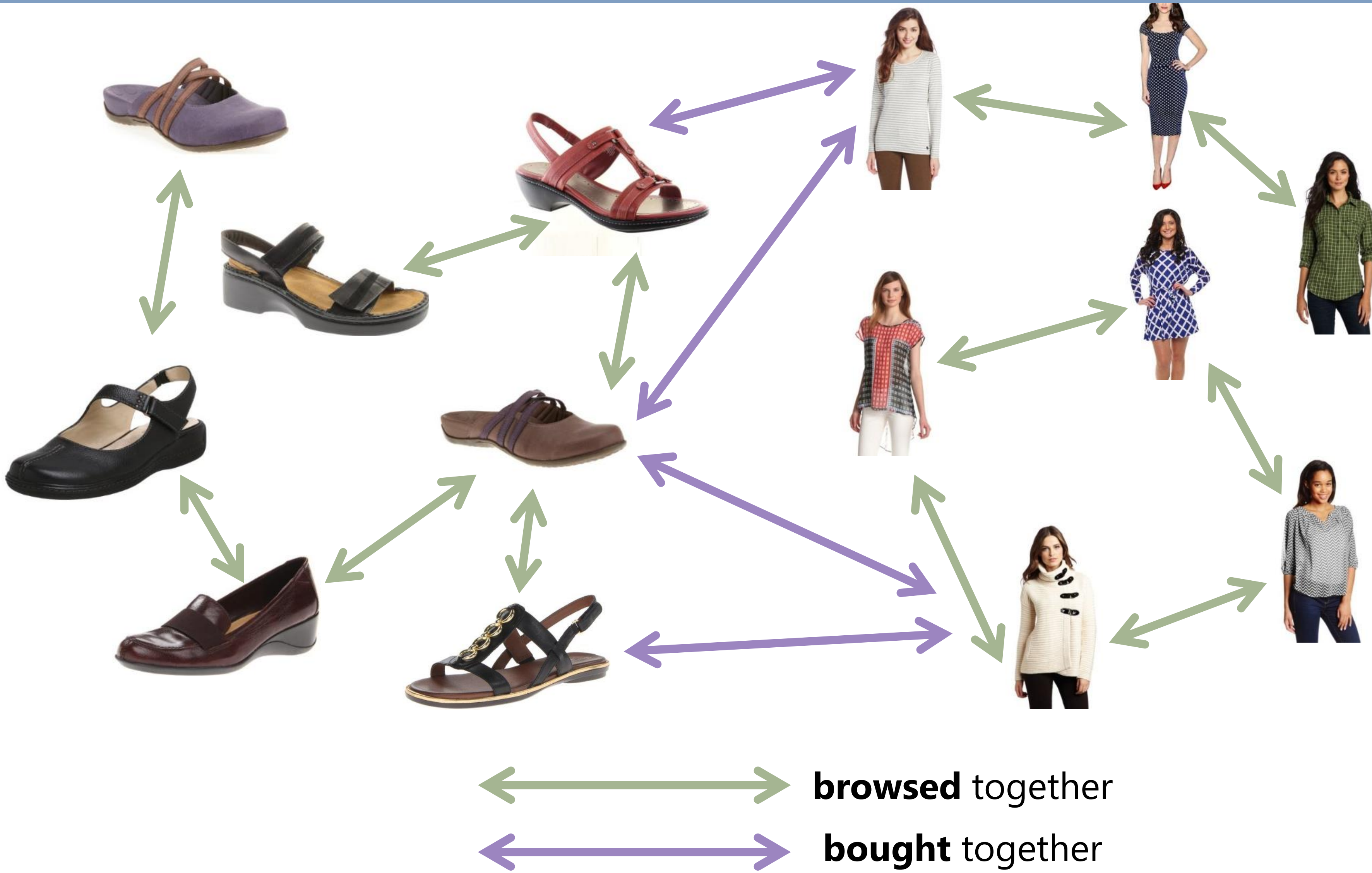
Image-based recommendations on styles and substitutes

Julian McAuley, UCSD
& Chris Targett, Qinfeng ('Javen') Shi, Anton van den Hengel, University of Adelaide

Relationships between products



Relationships between products



Modeling networks of **images**

Understanding product networks with **images**

Prediction: Can we estimate whether two products are likely to be purchased/browsed together?

Understanding: Can we understand which products have compatible visual "styles", and use this to recommend baskets of products to people?

Relationships between products – why?

1. To understand the notions of **substitute** and **complement goods**

Relationships between products – why?

2. To **recommend**
baskets of related items

Query:

Suggested outfit:

Query:

Suggested outfit:

Data

Amazon product network:

- thousands of **categories**
- 9 million **products**
- 21 million **users**
- 140 million **reviews**
- 300 million **relationships**

Data

Four types of relationship:

1) People who **viewed** X also **viewed** Y
2) People who **viewed** X eventually **bought** Y

3) People who **bought** X also **bought** Y
4) People **bought** X and Y **together**

Substitutes (1 and 2), and **Complements** (3 and 4)

Why might images be useful

- Visual explanations might be useful for some categories
- The image is the most important feature for many categories
- Cold-start problems

Problem setting

Binary prediction task:
Given a pair of products, **x and y**, predict whether they were purchased together, or whether they were chosen randomly

$$p(x \text{ and } y \text{ are related}) \sim -d(x, y)$$

Problem setting

But we are not **given** a distance function:
We need to **learn** the concept of similarity from data:

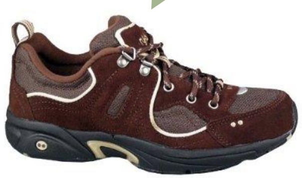
$$p_\theta(x \text{ and } y \text{ are related}) \sim -d_\theta(x, y)$$

Train θ by maximum likelihood:

$$\theta = \arg\max_{\theta'} \prod_{\text{edges } (x,y)} p_\theta(x \text{ and } y \text{ are related}) \prod_{\text{non-edges } (x,y)} (1 - p_\theta(x \text{ and } y \text{ are related}))$$

Problem setting

$$p(x \text{ and } y \text{ are related}) \sim -d(x, y)$$

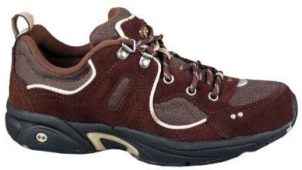


[0.723845, 0.153926, 0.757238, 0.983643, ...]

[0.456353, 0.898354, 0.123342, 0.234253, ...]

image features

Problem setting



$$\underbrace{[0.723845,\ 0.153926,\ 0.757238,\ 0.983643,\ \ldots\ ]}_{\text{4096-dimensional image features}}$$

We used **Caffe**, a convolutional neural net trained on **ImageNet**

http://caffe.berkeleyvision.org/

What are we actually learning?

How did Amazon generate their ground-truth data?

Given a product:

Let U_i be the set of users who viewed it

for every product in the corpus...

U_1 U_2 U_3 $\cdots$

What are we actually learning?

How did Amazon generate their ground-truth data?

Given a product:

Let U_i be the set of users who viewed it

Rank products according to: $\frac{|U_i \cap U_j|}{|U_i \cup U_j|}$ ('Jaccard index')

.86 .84 .82 .79 ...

Linden, Smith, & York (2003)

Attempt 1: distance between features

Features of (image of) product *i*:

$$\mathbf{x}_i = [0.723845, 0.153926, 0.757238, 0.983643, \ldots]$$

Features of product *j*:

$$\mathbf{x}_j = [0.456353, 0.898354, 0.123342, 0.234253, \ldots]$$

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_k \theta_k (\mathbf{x}_{i,k} - \mathbf{x}_{j,k})^2$$

Attempt 1: distance between features

Features of (image of) product *i*:

$$\mathbf{x}_i = \text{[0.723845, 0.153926, 0.757238, 0.983643, ...]}$$

Features of product *j*:

$$\mathbf{x}_j = \text{[0.456353, 0.898354, 0.123342, 0.234253, ...]}$$

At best we'll discover visual **similarity,**
but visual relationships are more subtle

Attempt 2: Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j) M (\mathbf{x}_i - \mathbf{x}_j)^T$$

$$M = \begin{array}{cc} \begin{array}{cccc} & \text{texture} & & \end{array} & \\ \begin{pmatrix} 0.1 & 0.2 & \cdots & 0.1 \\ 0.2 & 0.1 & & 0.6 \\ \vdots & & \ddots & \vdots \\ 0.1 & \boxed{0.6} & \cdots & 0.1 \end{pmatrix} & \begin{array}{c} \\ \\ \\ \text{color} \end{array} \end{array}$$

Attempt 2: Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j) M (\mathbf{x}_i - \mathbf{x}_j)^T$$

- High-dimensional
- Prone to overfitting
- Too slow!

Attempt 3: Low-rank Mahalanobis

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j) M (\mathbf{x}_i - \mathbf{x}_j)^T$$

Replace *M* by an approximation of low rank

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j) U U^T (\mathbf{x}_i - \mathbf{x}_j)^T$$

$$\boxed{d_v(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j) U \Delta_v U^T (\mathbf{x}_i - \mathbf{x}_j)^T}$$

user

user-personalized transform

(see paper)

Attempt 3: Low-rank Mahalanobis

let $\mathbf{s}_i = \mathbf{x}_i U$

$(1 \times K)$ $(1 \times F)$ $(F \times K)$

then $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{s}_i - \mathbf{s}_j\|_2^2$

We call this the 'style space' embedding of **x**

Training

$$U = \arg\max_{U'} \prod_{\text{edges } (x,y)} p_U(x \text{ and } y \text{ are related}) \prod_{\text{non-edges } (x,y)} (1 - p_U(x \text{ and } y \text{ are related}))$$

Results

Books

rank (K)	buy after viewing	also viewed	also bought	bought together	average
1	66.3%	66.1%	66.7%	60.7%	**65.0%**
10	72.4%	71.6%	72.1%	68.8%	**71.2%**
100	73.5%	72.4%	73.6%	69.0%	**72.1%**

Electronics

rank (K)	buy after viewing	also viewed	also bought	bought together	average
1	68.4%	74.7%	64.5%	72.3%	**67.5%**
10	83.4%	80.4%	77.6%	78.0%	**79.9%**
100	85.7%	84.0%	82.3%	82.4%	**83.6%**

Results

Clothing

rank (K)	also viewed	also bought	bought together	average
1	78.7%	75.4%	78.9%	**77.7%**
10	88.2%	86.8%	90.7%	**88.6%**
100	90.0%	90.8%	93.8%	**91.5%**

Shoes

rank (K)	also viewed	also bought	bought together	average
1	78.4%	78.9%	89.5%	**82.3%**
10	94.1%	95.3%	96.1%	**95.2%**
100	96.6%	97.6%	97.9%	**97.4%**

Visualizing 'style space'

We've projected images into a low dimensional space encoding their style, what are the "extreme" points?

Visualizing 'style space'

Visualizing ‘style space’

Visualizing 'style space'

Which styles are at **opposite** ends of the spectrum?

Generating recommendations

How can we use the system to generate recommendations?

Query:

Suggested outfit:

Generating recommendations

How can we use the system to generate recommendations?

Query:

Suggested outfit:

Generating recommendations

Outfits in the wild

Least coordinated

Most coordinated

Outfits in the wild

Old outfits

−53.2 −45.1 −28.2 −3.74 −2.69 4.80 5.32 6.17 7.47 8.54 9.22 9.46 12.25 12.81 18.71 22.72 34.46

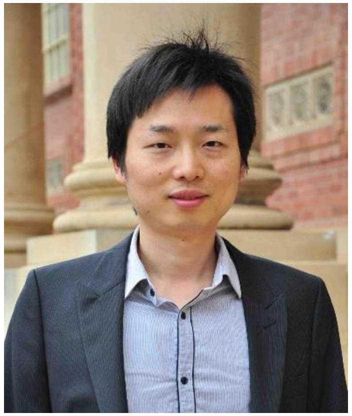
New outfits

Change in log-likelihood

Questions?

Co-authors:

Christopher Targett



Qinfeng "Javen" Shi

Anton van den Hengel

(The University of Adelaide)