Infusing Collaborative Recommendations with Distributed Representations

G. Zanotti[1] M. Horvath[2] L. N. Barbosa[2]
V. T. K. G. Immedisetty[1] J. Gemmell[1]

[1]Center for Web Intelligence
School of Computing, DePaul University

[2]CAPES Foundation
Ministry of Education of Brazil

DLRS '16

Outline

Outline

Computing similarities & predicting ratings

- Collaborative filtering (CF) relies on good similarity functions.
- **Challenge:** how to represent users/items and compute meaningful similarities?
- User-based CF: represent users by a vector of ratings given to items.
  - To compute similarity between users, use cosine distance.
  - Predict a user's rating for an item by weighted average of most similar users' ratings of the item.
- **Question:** can we improve this model by adding information to the representation?

Related work

- User-based and item-based collaborative filtering.
- Content-based recommenders.
- Social annotation systems include the use of tags to improve recommendations.
- Latent variable models (SVD, PCA, etc.).
- Neural networks have been used to fuse content-based and ratings-based information.

Our contributions

- To improve predicted ratings, improve the representation by incorporating tag and content-based data.
- We replace the simple "vector of ratings" vector space with a *distributed representation* from a more informative vector space.
  - Lowers the dimensionality of the ratings vector.
  - Provides a simple way to incorporate additional information.
- Simple cosine distance in a meaningful vector space = better ratings.
  - Makes it easy to improve existing nearest neighbor-based recommenders.
- **Bonus:** creates representations for tags, users, items, and content-based information.

Outline

Distributed representations

- Distributional hypothesis: *words which are similar in meaning occur in similar contexts* (Rubenstein & Goodenough, 1965):

Sam liked Sean Connery in Dr. No.

Dr. No is the first James Bond film.

Rhonda watched Dr. No last month.

- Under this hypothesis, using one vector element per item (i.e. one-hot encoding) to represent a word is inefficient.
- Distributed representations use multiple vector elements to represent concepts, and multiple concepts may share the same element.
  - Results in a more robust representation.
  - Lowers dimensionality.

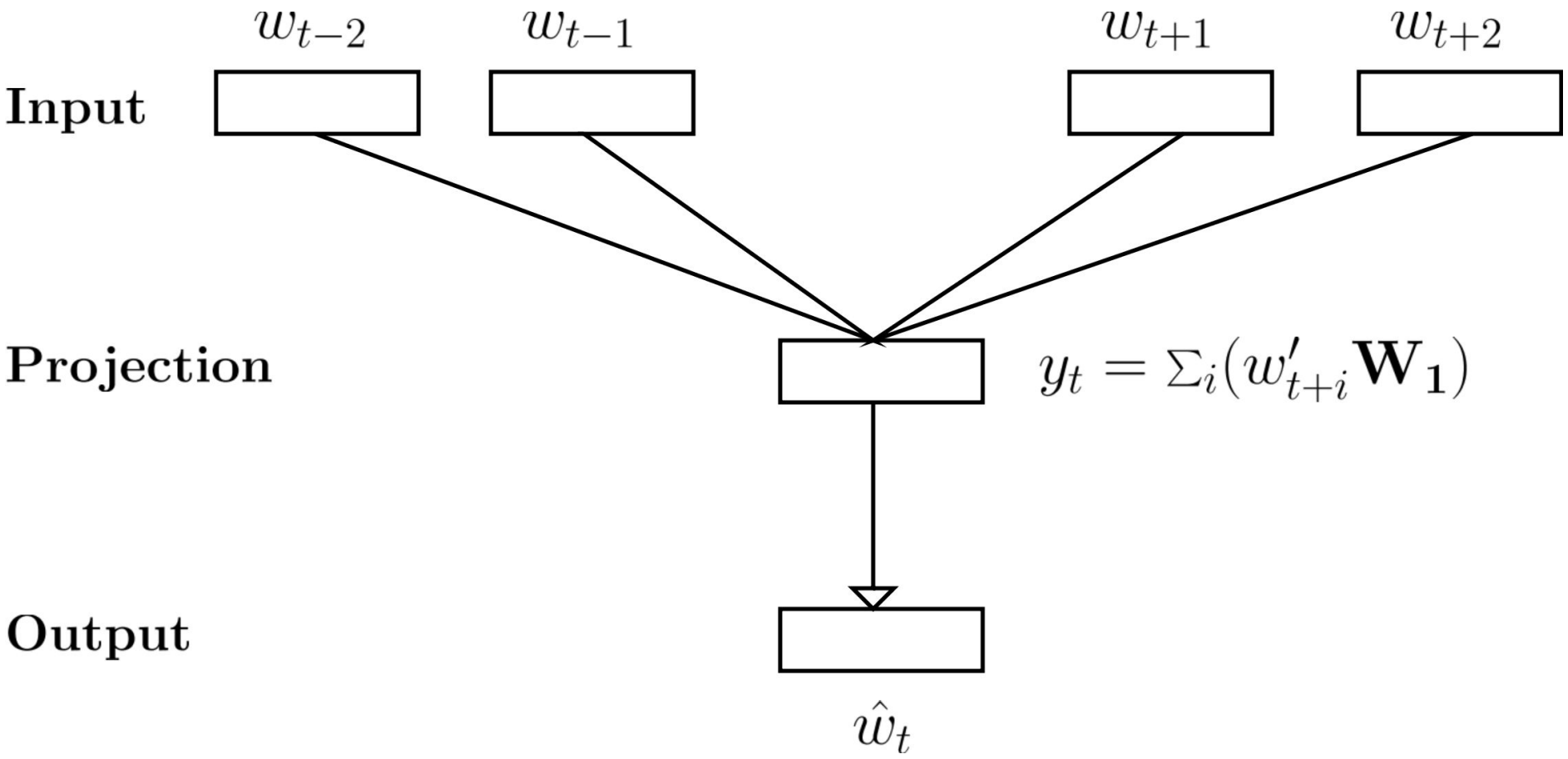
Meaningful vector spaces

- The **Continuous Bag of Words** and **Skip-gram** models (Mikolov, *et al.* 2013) learn distributed representations for words.
- In these vector spaces, magnitude and direction are meaningful under addition, e.g. $[\text{king}] - [\text{man}] + [\text{woman}] \approx [\text{queen}]$.
- These models replace the nonlinearities of traditional neural networks with special architectures and update methods, allowing for faster training on more data.

Continuous Bag of Words

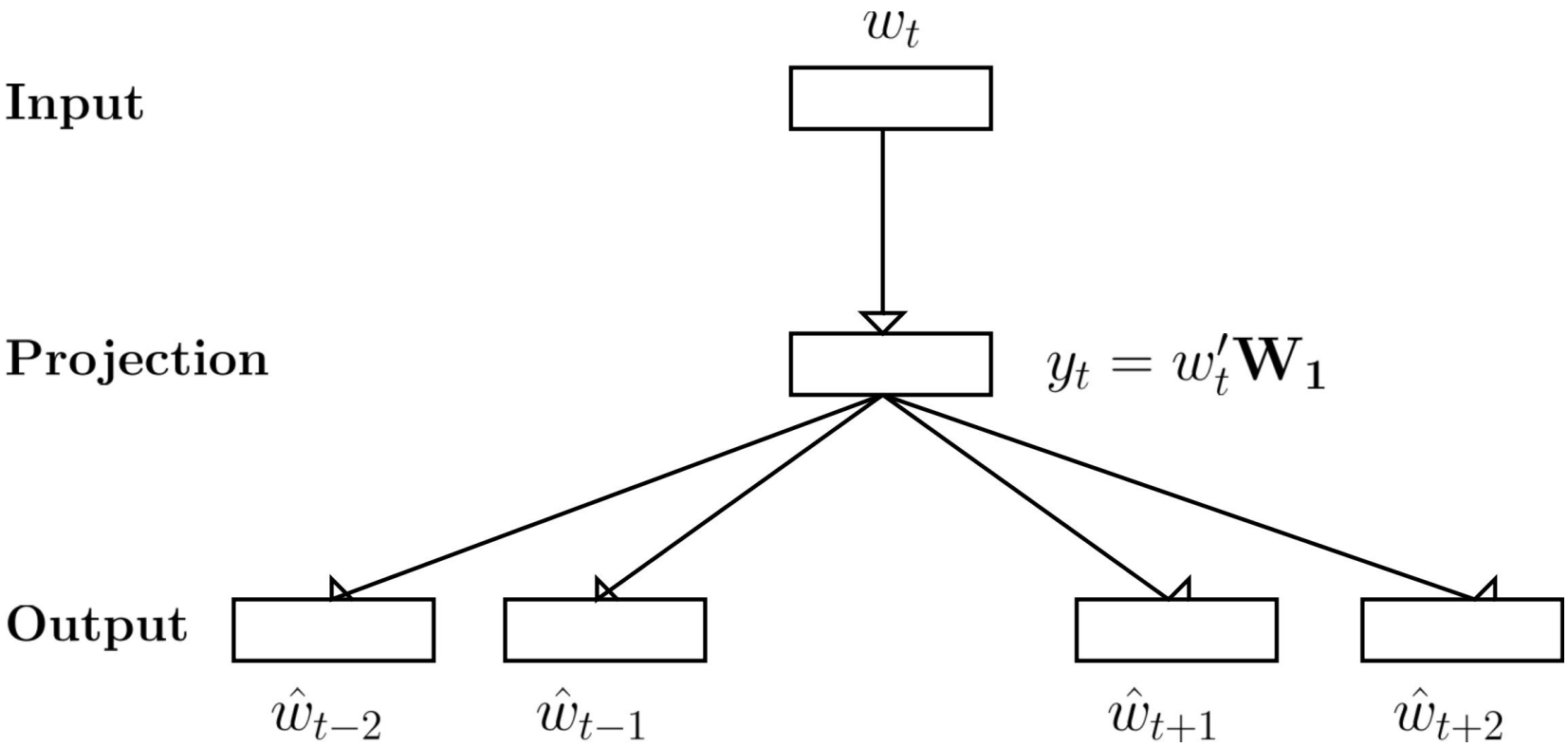
- The Continuous Bag of Words model (CBOW) is a neural network composed of an input, projection, and output layer.
- **Goal:** predict word given surrounding context words.
- Trains on a collection of sentences.
- Inputs are one-hot encoded, output is as well.
- Projection layer learns lower-dimensional distributed representation.

Continuous Bag of Words



Skip-gram

- The **Skip-gram** (SG) model has a similar architecture, but the goal is different: predict the context words given the input word.



Outline

Infusing CF with distributed representations

- If we treat items, users, tags, etc. as words, we can learn from their co-occurences.
- **Our translation:** If we want to find a movie similar to e.g. "Dr. No", we should examine the "words" around it: users, actors, tags, etc.
- We extend user-based and item-based collaborative filtering by replacing traditional user/item representations with those learned by CBOW and SG models, e.g.

$$\vec{u} = [w(f_1), w(f_2) \ldots w(f_{|F|})]$$

Hybrid recommender model

- When computing *user* similarity using CBOW or SG, we abbreviate our method **UBCB** and **UBSG**, respectively.
- When computing *item* similarity using CBOW or SG, we abbreviate our method **IBCB** and **IBSG**, respectively.
- We combine traditional IBCF and UBCF with our models using a hybrid recommender, the Linear Weighted Hybrid method (Gemmell 2012):

$$p_h(u,i) = \sum_{c \in C} \alpha_c p_c(u,i)$$

- C is the set of models, α_c is the weight assigned to a model c, and $p_c(u,i)$ is the model's prediction.
- α weights are set uniformly, learned iteratively, and restricted to be convex.

Outline

Experimental process

- Combined MovieLens 10M with Internet Movie Database (IMDb) metadata to create artificial sentences.
  - Each included a user ID, movie ID, set of tags applied by the user to the movie, genre, and leading actors and directors.
- Sentences were processed by the CBOW and SG models to create distributed representations.
- Each user profile was divided equally across five partitions, and predictions were created by cross-validation.

Learned semantic information I

Movies	Directors	Actors	Tags
Prisoner of Azkaban (0.92)	Alfonso Cuaròn (0.71)	Pam Ferris (0.98)	best in franchise (0.90)
Order of the Phoenix (0.82)	David Yates (0.71)	Daniel Radcliffe (0.82)	love this movie so much (0.86)
Chamber of Secrets (0.79)	Kazuya Murata (0.67)	Harry Melling (0.81)	ignores established character (0.85)
Half-Blood Prince (0.75)	Walter Murch (0.66)	Jason Boyd (0.81)	potter (0.83)
Goblet of Fire (0.75)	Rod Hardy (0.65)	Richard Griffiths (0.81)	emma thompson (0.82)

Table 1: The top five most similar movies, directors, actors and tags along with their similarity to the distributed representation of the movie "Harry Potter and the Philosopher's Stone".

- Resulting movies are all from the Harry Potter franchise, even though the movies don't occur together in training sentences.
- Alfonso Cuaròn and David Yates directed most of these movies.
- Actors include Daniel Radcliffe, the lead actor in the series, as well as supporting actors, Pam Ferris and Harry Melling, whose most popular roles occur in these movies.

Learned semantic information II

Movies	Directors	Actors	Tags
Saving Private Ryan (0.70)	Steven Spielberg (0.79)	Tom Hanks (0.79)	best war cinematography (0.70)
Catch Me If You Can (0.69)	Lee Unkrich (0.54)	Russ Meyer (0.59)	adult diaper commercial (0.70)
The Terminal (0.67)	Robert Zemeckis (0.52)	Rebecca Williams (0.59)	fun but unrealistic (0.70)
Forrest Gump (0.60)	John Lasseter (0.51)	Stephen Ambrose (0.58)	tom hanks (0.69)
Lincoln (0.57)	Steve Purcell (0.51)	Alexander Godunov (0.58)	fellowship (0.68)

Table 2: The top five most similar movies, directors, actors and tags along with their similarity to the resulting distributed representation of "Steven Spielberg" plus "Tom Hanks".

- Combining distributed representations to produce a new representation is also possible.
- First three movies were directed by Spielberg and starred Hanks.
- The directors other than Spielberg have all worked with Hanks.

Learned semantic information III

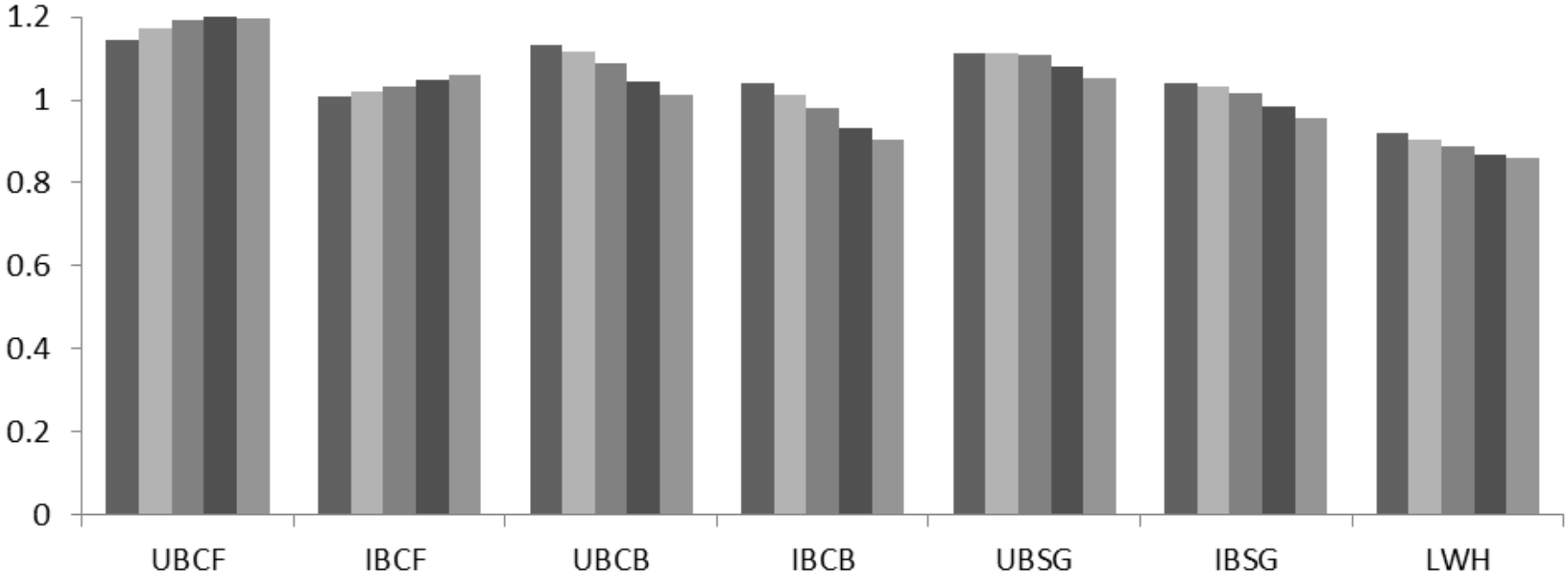
Movies	Directors	Actors	Tags
Octopussy (0.80)	John Glen (0.78)	Roger Moore (0.85)	desmond llewelyn (0.80)
For Your Eyes Only (0.78)	Peter Hunt (0.73)	Robert Davi (0.77)	setting circus (0.80)
A View to a Kill (0.77)	Michael Damian (0.72)	Carey Lowell (0.76)	maud adams (0.80)
The Spy Who Loved Me (0.77)	Jean-Claude Van Damme (0.68)	Tanya Roberts (0.76)	kabir bedi (0.80)
A Princess for Christmas (0.77)	Harvey Hart (0.66)	Michael Lonsdale (0.76)	kristina wayborn (0.80)

Table 3: The top five most similar movies, directors, actors and tags along with their similarity to the resulting distributed representation after computing "Dr. No" minus "Sean Connery" plus "Roger Moore".

- Here, "Sean Connery" is subtracted from the popular James Bond movie, "Dr. No". Then, "Roger Moore", the actor who replaced Sean Connery in the Bond franchise, was added.
- Top four resulting movies are Bond films starring Roger Moore.
- John Glen and Peter Hunt have both directed Bond films.
- Remaining actors have all appeared in Bond films.
- Tags include Desmond Llewelyn, best known for his role as Q in 17 of the James Bond films.

Evaluating predicted ratings

- Comparison baseline: traditional UBCF and IBCF models.
- Computed RMSE for neighborhoods of size 5, 10, 20, 50 and 100.
- Best recommender was the item-based recommender using the Continuous Bag of Words model (**IBCB**) with a neighborhood of 100, achieving an RMSE of 0.905.



Linear Weighted Hybrid weights

- The LWH process reduces the RMSE to 0.858.
- Examining the weights shows that the CBOW/SG-based models learn valuable information.

	UBCF	IBCF	UBCB	IBCB	UBSG	IBSG
α	0.032	0.058	0.144	**0.419**	0.122	0.225

Table: The weights assigned to each of the component recommenders indicating their contribution to the linear weighted hybrid with a neighborhood of size 100.

Summary

- We created a meaningful vector space of distributed representations for users, items, tags, genres, actors, and directors.
- Our method is extensible, adapts to improve existing methods, and can consume additional information easily.

- Future directions
  - Examine additional models for creating meaningful vector spaces.
  - Improve model evaluation and explore new architectures and vector update methods.

Thanks for listening.

CBOW Training

- Projection layer activations are created by matrix multiplication with the *same* projection matrix.
- Activations are summed or averaged to produce single vector of dimension D.
- Output layer (i.e. another matrix multiplication) produces output vectors of dimension V.
- Model is trained by optimizing the likelihood function formed by feeding the output layer activations σ into the softmax function:

$$\Pr(w_k|\mathbf{w}_c) = \frac{\exp(\sigma(k))}{\sum_{n=1}^{V} \exp(\sigma(n))}$$

- Error vector is backpropagated.

Skip-gram Training

- The softmax function acts on the output layer activations for each context word $w_{c,i}$ fed into the network.
- The softmax functions takes the following form, where the $w_{c,i}$ is the ith context word for the target word w_k, and $\sigma(c, i)$ is the activation corresponding to $w_{c,i}$:

$$\Pr(w_{c,i}|w_k) = \frac{\exp(\sigma(c, i))}{\sum_{m=1}^{V} \exp(\sigma(m))}$$

- Error vectors are summed and then sum vector is backpropagated.