Table of Contents

[مقدمه 3](#_Toc484385529)

[مروری روش‌ها موجود و کار‌های قبلی 9](#_Toc484385530)

[مقدمه 9](#_Toc484385531)

[روش‌های مبتنی بر ویژگی‌های پنهان 10](#_Toc484385532)

[تقسیم‌بندی داده‌های پایگاه دانش 12](#_Toc484385533)

[مراحل آموزش مدل 14](#_Toc484385534)

[روش‌های موجود 17](#_Toc484385535)

[۲- روش‌های دوخطی – رسکال 18](#_Toc484385536)

[۳- روش‌های ادراک چندلایه‌ای 23](#_Toc484385537)

[۳- روش‌های فاصله‌ی پنهان 28](#_Toc484385538)

* مقدمه
  + ضرورت و هدف
  + تعریف مساله
  + روش‌های انجام پژوهش
  + ساختار پایان‌نامه
* پیش‌بینی لینک در شبکه‌های ناهمگون
  + شبکه‌های ناهمگون
  + پیش‌بینی لینک
  + چالش‌ها
* مروری بر مطالعات انجام شده
  + مقدمه
  + نحوه‌ی آموزش
  + روش‌های مبتنی بر ویژگی‌های پنهان
  + نتایج
* عملکرد روش‌های موجود
  + مقیاس‌پذیری
  + نتایج روش‌ها
  + نقاط قوت و ضعف
* قوانین انجمنی
  + مقدمه
  + معیار اطمینان
  + الگوریتم
  + قوانین هدف
* استفاده از قوانین انجمنی
  + نحوه‌ی بهبود در مدل‌ها
* نتیجه گیری و کار‌های آینده
  + نتیجه‌گیری
  + کار‌های آینده
* مراجع
* پیوست‌ها

# مقدمه

تعریف مساله و هدف و ضرورت

در سال‌های اخیر شاهد رشد بسیار زیادی در شبکه‌های اجتماعی بوده‌ایم و مطالعات زیادی روی این شبکه‌ها انجام گرفته است. داده‌های شبکه‌های اجتماعی یکی از ابزار محبوب برای مدل کردن رابطه و رفتار افراد و جامعه یا گروهی که در آن عضو هستن بشمار می‌رود. این داده‌ها معمولا به صورت گرافی نمایش داده می‌شود که در آن گره‌ها افراد و لبه‌ها روابط بین این افراد می‌باشد. پیش‌بینی وقوع لینک‌ها، یک مساله اساسی و بنیادین در شبکه‌های اجتماعی می‌باشد. در موضوع پیش‌بینی لینک، نمایی از یک شبکه به ما داده می‌شود و ما مایل هستیم که بدانیم در آینده نزدیک، احتمالا چه تراکنش‌هایی میان اعضای فعلی شبکه روی می‌دهد و یا اینکه کدام یک از تراکنش‌های موجود را از دست می‌دهیم.

اکثر مطالعات انجام شده در این زمینه روی شبکه‌های تک‌رابطه‌ای بوده است. به این معنا که روابط بین موجودیت‌ها از یک نوع است و این روابط به صورت دوتایی‌های مرتب مورد استفاده قرار می‌گیرند. برای مثال اگر در یک شبکه اجتماعی رابطه را دوستی بین افراد در نظر بگیریم یال‌های گراف شبکه به صورت «الف، ب» خواهد بود به این معنی که شخص الف با شخص ب رابطه دوستی دارد.

در سال‌های اخیر پایگاه‌های دانشی ایجاد شده‌اند که داده‌ها در آن به صورت چند رابطه‌ای ذخیره شده‌اند و اطلاعات بیشتری از یک رابطه دوتایی بلی یا خیر به ما می‌دهند. ما در این تحقیق از داده‌های چند رابطه‌ای استفاده می‌کنیم. منظور از داده‌های چند رابطه‌ای گراف جهت‌داری است متشکل از موجودیت‌ها و روابط بین آن‌ها که بصورت «مبداh، رابطهr، مقصدt» نمایش داده می‌شود، به این معنی که یک رابطه r بین موجودیت‌های h وt وجود دارد. برای مثال سه‌تایی «تهران، واقع در، ایران» این اطلاع را به ما می‌دهد که استان تهران داخل کشور ایران قرار دارد. در این نوع پایگاه دانش هم انواع مختلف موجودیت وجود دارد و هم انواع مختلف رابطه بین موجودیت‌ها. پایگاه‌های دانشی مانند Google Knowledge Graph، FreeBase و WordNet وجود دارند که شامل تعداد زیادی نمونه چندرابطه‌ای می‌باشند و تعداد زیادی موجودیت و روابط بین آن‌ها را می‌توان در آن‌ها یافت و از آن برای یادگیری مدل استفاده کرد. شناخت پایگاه‌دانش و دریافت اطلاعات از آن نیز یکی از مسايل پیشبینی‌لینک روی داده‌های چند رابطه‌ای است.

پیشبینی لینک مقیاس‌پذیر روی داده‌های چند رابطه‌ای عمدتا با استفاده از ویژگی‌های پنهان صورت می‌گیرد. مشکل این روش‌ها تعداد بسیار زیاد پارامترها در فاز یادگیری مدل است به صورتی که انجام عملیات یادگیری مدل را غیرممکن می‌کند. کاهش تعداد پارامترهای مساله که در برخی روش‌ها پیشنهاد شده نیز برخی از الگوی‌های موجود در پایگاه دانش را در نظر نمی‌گیرند و باعث کاهش دقت مدل خواهند شد. در این تحقیق قصد داریم که بدون افزایش تعداد پارامتر‌های روش‌های الگوی پنهان شده و با استفاده از قوانین انجمنی موجود در پایگاه‌دانش عمل‌کرد این روش‌ها را بهبود بخشیم.

در این تحقیق قصد داریم که با کار روی داده‌های چند رابطه‌ای به سؤالاتی که از پایگاه‌دانش می‌شود پاسخ دهیم، به این صورت که یک مدل از روی داده‌های پایگاه دانش ساخته و آموزش داده می‌شود و پس از آن سؤالات به گونه‌ای که مقصد یا مبدأ آن مجهول باشد از مدل ایجاد شده پرسیده می‌شود. برای نمونه در مثال بالا اگر قسمت کشور مجهول باشد از پایگاه دانش پرسیده می‌شود «تهران، واقع در، ؟» و انتظار می‌رود که مدل آموزش دیده پاسخ سؤال را با دقت نسبتاً خوبی بیابد.

نتایج این مساله کاربرد‌های زیادی خواهد داشت به عنوان مثال در ادامه تعدادی از این کاربردها را مطرح خواهیم کرد:

* پیشبینی لینک‌های احتمالی در شبکه‌های اجتماعی، برای مثال در شبکه‌های اجتماعی بین کاربران و مطالب ثبت شده، نظرات و … بررسی شود که روابطی مانند دوستی، پسندیدن و نپسندیدن، روابط فامیلی و… وجود دارد یا خیر
* استفاده یه عنوان سیستم‌های توصیه‌گر، برای مثال کاربران و کالاها یا اشیاء موجودیت‌ها هستند و خریدن، امتیاز دادن، بررسی کردن و … رابطه‌ها هستند که می‌توان از بررسی این روابط و موجودیت‌ها اطلاعاتی نظیر کاربر xکالای y را خواهد خرید یا خیر یا اینکه کالای x به تعداد بالا فروش خواهد رفت یا خیر بدست آورد.
* کامل‌تر کردن پایگاه‌های داده، از نتایج این تحقیق می‌توان استفاده کرد و با اضافه کردن رابطه‌های انجمنی بدست آمده به پایگاه‌داده آن را کامل‌تر کرد.

روش‌ها:

داده‌هایی که قصد کار روی آن‌ها را داریم معمولاً ابعاد بالایی خواهند داشت. برای مثال پایگاه دانش Freebase شامل یک میلیون نوع موجودیت و ۲۵هزار نوع رابطه و ۱۷ میلیون نمونه برای آموزش و آزمون مدل است. از آنجایی که کار روی داده‌هایی با این تعداد ابعاد بسیار سخت و تقریباً غیر ممکن است باید از روش‌های کاهش ابعاد استفاده کنیم. در این تحقیق سعی می‌کنیم که این داده‌ها را به فضای برداری آورده و مساله را به یک مساله کمینه‌سازی تبدیل کنیم. این داده‌ها را به صورت یک تنسور[[1]](#footnote-1) فرض می‌کنیم که برای هر رابطه یک فضای دوبعدی برداری برای موجودیت‌های مختلف در نظر می‌گیریم، همچنین برای هر رابطه یک بردار بین موجودیت‌های دخیل در آن رابطه خواهیم داشت. سپس تلاش می‌کنیم که از این بردار ها در یک مساله کمینه سازی استفاده کنیم. برای مثال اگه سه‌تایی «head, relation, tail» را داشته باشیم که هرکدام به یک بردار نگاشت شده‌اند انتظار داریم مجموع بردار head و relation به بردار tail برسد پس باید مقدار زیر را کمینه کنیم:

head + relation – tail => 0

برای نمایش این قانون از تخمین مرتبه پایین استفاده می‌کنیم به صورت:

fr(h,t) = ||h + r – t||22

در صورتی که تابع f برای سه‌تایی «h, r, t» مقداری نزدیک به صفر داشته باشد این رابطه برقرار است و درغیر این صورت این رابطه برقرار نیست.

در تحقیقاتی که قبلاً روی این مدل داده‌ها انجام شده است فقط از حقایقی که به صورت مستقیم داده شده استفاده شده است. برای مثال اگر در حقایق موجود در داده‌ها داشته باشیم «الف، مادر، ب» و «الف، همسر، ج» فقط از این دو حقیقت استفاده می‌شود و از این حقیقت که «ج، پدر، ب» استفاده نمی‌شود. در این تحقیق تصمیم داریم از قوانین انجمنی بین داده‌ها نیز استفاده کرده و حقایق ضمنی که بین داده‌ها هست را نیز در آموزش مدل استفاده کنیم.

یادگیری ماشین مدت زیادی است که در علوم کامپیوتر جایگاه خود را پیدا کرده و به عنوان ابزار قدرمتندی برای کمک به انسان در زمینه‌های مختلف محسوب می‌شود و ماشین را بیش از پیش در خدمت انسان در آورده است. در روش‌های اولیه یادگیری ماشین، عمدتا از داده‌ها و متن‌های خام در زمینه‌ی یادگیری استفاده میشد.

اخیرا از طرف برخی شرکت‌ها و موسسات بزرگ، همچون گوگل، ای‌بی‌ام، مایکروسافت و … پایگاه‌های دانشی معرفی شده است که انجام راه‌کار‌های مختلف یادگیری ماشین را ساده‌تر و کاربردی‌تر کرده است. در این پایگاه‌های دانش اطلاعات مورد نیاز برای عملیات‌های مختلف یادگیری به صورت منظم و نیمه‌منظم موجود است و دغدغه‌ی نرمال کردن و رفع خطا و استخراج حقایق رو به مقدار زیادی کم کرده است. این پایگاه‌های دانش عمدتا به صورت یک شبکه از موجودیت‌ها و روابط بین آن‌ها که می‌توان آن را به صورت یک گراف داده نمایش داده به این صورت که گره‌ها موجودیت‌ها و یال‌های بین گره‌ها نشانگر روابط بین آن‌ها باشند، که این یال‌ها می‌توانند از یک نوع باشند یعنی گراف نشانگر یه شبکه تک-رابطه‌ای باشد (همگون) یا هر یال با یک برچسب، نشانگر نوع رابطه باشد و شبکه نشانگر یک گراف چند رابطه‌ای (ناهمگون) باشد.

یکی از راهکارهای یادگیری ماشین، پیش‌بینی لینک در شبکه‌های همگون و ناهمگون است به این معنی که از اطلاعات موجود در گراف دانش استفاده کرده و وجود یا عدم وجود یک یال را پیش‌بینی کرد. این راهکار در زمینه‌های مختلف یادگیری ماشین مورد استفاده قرار می‌گیرد و کاربرد گسترده‌ای در زندگی انسان پیدا کرده است. برای مثال از این راه‌کار در سیستم‌های توصیه‌گر در فروشگاه‌های اینترنتی، سیستم‌های تشخیص پزشکی، جواب‌گویی به سوال و … استفاده می‌شود.

روش‌های مختلفی برای حل مساله‌ی پیش‌بینی لینک در پایگاه‌های دانش ناهمگون ارائه شده است که از رویکرد‌های مختلفی سعی به حل این مساله می‌کنند از جمله روش‌های آماری، روش‌های ویژگی‌های پنهان، روش‌های ویژگی‌های گراف و …، تمرکز ما در این مقاله بر روی روش‌های پیش‌بینی لینک مبتنی بر ویژگی‌های پنهان است.

در روش‌های مبتنی بر ویژگی‌های پنهان، با استفاده از ویژگی‌هایی که در موجودیت‌ها و روابط بین آن‌ها وجود دارد سعی می‌شود میزان ارتباط بین موجودیت‌ها را تشخیص دهیم و به این صورت وجود یک لینک را تایید یا رد کنیم. برای مثال اگر دو شخص با هم همکار هستند، به احتمال زیادی ویژگی‌های مشترکی دارند، مثلا هر دو اهل یک شهر هستند، هر دو در یک رشته‌ی دانشگاهی تحصیل کرده‌اند، خصوصیات اخلاقی یکسانی دارند و …. در رابطه‌ی همکار بودن به هیچ یک از این ويژگی‌ها به طور مستقیم اشاره نشده است و این ویژگی‌ها به طور ضمنی در این موجودیت‌ها قرار دارند که با استفاده از آن می‌توانیم وجود یا عدم وجود رابطه‌ی همکار بودن را حدس بزنیم. پس هر موجودیت می‌تواند تعداد زیادی ویژگی پنهان داشته باشد که رابطه‌ها به این ویژگی‌ها وزن می‌دهند، مثلا در رابطه‌ی همکاری احتمال اینکه رشته‌ی تحصیلی دو شخص در همکار شدن آن‌ها تاثیر گذار باشد بیشتر از ویژگی رنگ پوست دو شخص است، پس وزن ویژگی رشته‌ي تحصیلی در این رابطه بیشتر از ویژگی رنگ پوست است.

داده‌های چندرابطه‌ای به یک گراف جهت‌دار گفته می‌شود که در آن گره‌ها نشانگر موجودیت‌ها و یال‌ها نشانگر رابطه‌ی بین موجودیت‌ها است، که هر رابطه به صورت یک سه‌تایی (t، r، h)[[2]](#footnote-2) نمایش داده می‌شود به این معنی که موجودیت h رابطه‌ی به نام r دارد با موجودیت t. [TransE] مدل‌های بر اساس داده‌های چند رابطه‌ای نقش اساسی‌ای در بسیاری از زمینه‌های یادگیری ماشین بازی می‌کنند. برای مثال:

* در آنالیز شبکه‌های اجتماعی که در آن اعضای شبکه‌ی اجتماعی گره‌های گراف (موجودیت‌ها) و روابط دوستی بین اعضا یال‌ها (رابطه‌ها) را تشکیل می‌دهد.
* در سیستم‌های توصیه‌گر مانند یک فروشگاه آنلاین که در آن کاربران و کالا‌ها موجودیت‌ها و لینک‌هایی مانند خرید کردن، نظر دادن، امتیاز دادن، جستجو کردن رابطه‌ها را در گراف داده تشکیل می‌دهند.

# مروری روش‌ها موجود و کار‌های قبلی

مقدمه

در این بخش به معرفی روش‌های موجود که به حل مساله‌ی پیش‌بینی لینک در شبکه‌های ناهمگون می‌پردازند، خواهیم پرداخت. روش‌هایی با راه‌کار‌های مختلفی در حوزه‌های مختلف تلاش به حل این مساله کرده‌اند.این روش‌ها می‌توانند به سه دسته‌ی مختلف تقسیم می‌شوند: ۱- روش‌های مبتنی بر ویژگی‌های گراف ۲- روش‌های مدل تصادفی مارکوف ۳- روش‌های مبتنی بر ویژگی‌های پنهان. در ادامه توضیح مختصری در مورد هر دسته از روش‌ها می‌دهیم.

* در روش‌های مبتنی بر ویژگی‌های گراف از روی ویژگی‌های ساختاری گراف داده‌ها استفاده می‌شود مانند دسته‌بندی گره‌ها، دسته‌بندی نوع یال‌ها، تعداد گره‌های مشابه و ...
* روش‌های مدل تصادفی مارکف که در آن دید بالایی از گراف داده نداشته و سعی می‌کنیم مساله را به صورت محلی حل کنیم به این صورت که روابط هر موجودیت را با موجودیت‌های اطراف آن بررسی می‌کنیم و جواب‌های محلی را بدست ‌می‌آوریم.
* روش‌های مبتنی بر ویژگی‌های پنهان که در این روش‌ها هر موجودیت و نوع رابطه بین آنها به صورت برداری از ویژگی‌های پنهان تعریف می‌شود که ویژگی‌های پنهان نام دارد. برای مثال ویژگی‌هایی که یک موجودیت می‌تواند داشته باشد، محل به وجود آمدن آن، سن آن، جاندار یا بی‌جان بودن آن و ... است.

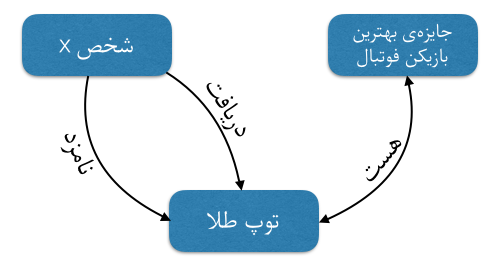
همان‌طور که در بخش XX گفتیم تمر‌کز ما در این تحقیق روی روش‌های مبتنی بر ویژگی‌های پنهان است، در ادامه این روش‌ها را به صورت کامل توضیح داده و تعدادی از آن‌ها را به طور مختصر معرفی کرده و نحوه‌ی کار آن‌ها را توضیح می‌دهیم و نتایج بدست آمده از آن‌ها را توضیح می‌دهیم.

## روش‌های مبتنی بر ویژگی‌های پنهان

روش‌های مبتنی بر ویژگی‌های پنهان از جدید ترین راه‌کار‌هایی است که در حوزه‌ی پیش‌بینی لینک روی پایگاه‌های دانش موجود معرفی شده است.همانطور که در بخش قبل گفتیم این روش از ويژگی‌هایی که در نگاه اول از پایگاه‌دانش برداشت نمی‌شود استفاده می‌کند که ویژگی‌های پنهان نام دارند، در همه‌ی روش‌های مبتنی بر ویژگی پنهان این ویژگی‌ها را به صورت برداری تعریف می‌کنیم که هر مولفه از این بردار نشان دهنده‌ی یک ویژگی می‌باشد.

برای مثال در رابطه‌ی دوستی فاکتور‌هایی تاثیر گذار هستن و اگر در پایگاه دانش همه‌ی رابطه‌های دوستی موجود را بررسی کنیم به یک بردار از فاکتور‌ها می‌رسیم که بردار رابطه‌ی دوستی را تشکیل می‌دهد. مثلا فاکتور‌هایی مانند شهر محل زندگی، سن، دانشگاه، رشته‌ی دانشگاهی، جنسیت، مذهب و ... در شکل گیری رابطه‌ی دوستی می‌تواند موثر باشد، اما اینکه هر کدام از این روابط چقدر در ایجاد رابطه‌ی دوستی تاثیر دارند و اهمیت هر کدام چقدر است و این میزان اهمیت را چگونه در تشخیص این رابطه تاثیر دهیم به مدل یادگیری بستگی دارد که در ادامه در معرفی هر یک از روش‌ها به صورت کامل توضیح داده خواهد شد.

برای مثال برای اینکه بررسی کنیم شخص X بازیکن فوتبال خوبی هست یا خیر از دیگر روابط موجود استفاده می‌کنیم و میزان ارتباط این شخص را با معیار‌های بازیکن خوب فوتبال بودن بررسی می‌کنیم. در مثال شکل – شخص x هم برای توپ طلا نامزد شده و هم توپ طلا را دریافت کرده و از اطلاعات دیگر پایگاه دانش می‌دانیم که توپ طلا به بهترین بازیکن فوتبال هر سال داده می‌شود. پس شخصی که این جایزه را دریافت کرده بازیکن فوتبال خوبی است.



روش‌های زیادی برای مدل‌های مبتنی بر ویژگی‌های پنهان معرفی شده است، روش‌هایی که در این تحقیق مورد مطالعه و بررسی قرار گرفته این به صورت زیر دسته‌بندی می‌شوند.

* روش‌های خطی
* روش‌های دو خطی
  + RESCAL
* روش‌های ادراک چندلایه‌ای
  + Neural Tensor Network (NTN)
* روش‌های فاصله‌ی پنهان
  + Structured Embedding
  + Translating Embedding (TransE)
  + Translating on Hyperplane (TransH)

در ادامه ابتدا پایگاه‌دانش Freebase که مطالعات روی آن انجام می‌گیرد و نحوه‌ی تقسیم ‌بندی آن را توضیح خواهیم داد. سپس روش کلی آموزش مدل‌های مبتنی بر ویژگی‌های پنهان را شرح می‌دهیم و در ادامه روش‌هایی که در بالا نام برده شدند را توضیح داده و نقاط قوت و ضعف آن‌ها را بررسی می‌کنیم و نتایج بدست آمده از هریک از این روش‌ها را نیز بررسی خواهیم کرد و در فصل بعد تلاش بر بهبود این روش‌ها خواهیم کرد.

## تقسیم‌بندی داده‌های پایگاه دانش

برای آموزش دادن مدل‌های نام برده شده در بخش قبل از پایگاه دانش Freebase که قسمتی از گراف دانش تولید شده توسط گوگل است استفاده می‌کنیم. در این پایگاه دانش ۸۰ میلیون موجودیت، ۲۰ هزار نوع رابطه مختلف و ۱.۲ میلیارد حقیقت وجود دارد. حقیقت‌هایی که در این پایگاه دانش وجود دارد به صورت سه‌تایی مرتب

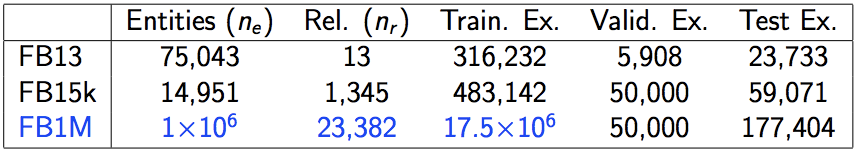
(subject, predicate, object)

مشخص شده‌اند به این صورت که موجودیت subject رابطه‌ی predicate دارد با موجودیت object. به این نوع ذخیره‌ی داده اصطلاحا SPO گفته می‌شود. برای مثال یک نمونه حقیقت موجود در این پایگاه دانش به صورت:

(Barack Obama, place\_of\_birth, Hawai)

است که بیانگر حقیقت «باراک اوباما متولد هاوایی است» می‌باشد.

این پایگاه‌دانش شامل تعداد زیادی حقیقت است که عملیات آموزش روی آن هزینه‌ی زیادی از نظر زمان و منابع خواهد داشت. برای سادگی و تسریع کار از یک نمونه‌ی نرمال کوچک این پایگاه‌دانش به نام Freebase15k استفاه می‌کنیم که در آن ‍۱۴۹۵۱ موجودیت، ۱۳۴۵ رابطه‌ی مختلف و ۵۹۲۲۱۳ حقیقت وجود دارد.

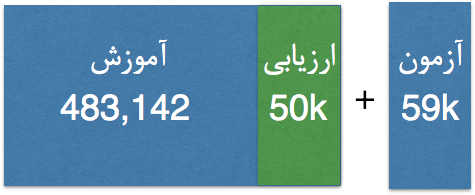


روش‌های مبتنی بر ویژگی‌های پنهان به صورت تکراری انجام می‌شوند و نیاز است در هر تکرار بررسی کنیم که به آستانه‌ی مناسب برای قطع تکرار الگوریتم رسیده‌ایم یا خیر. همچنین پس از انجام عملیات آموزش نیاز است که مدل آموزش دیده را آزمایش کنیم و میزان دقت آن را بدست آوریم.

برای انجام عملیات آموزش، بررسی کیفیت آموزش در هر مرحله و بررسی کیفیت کلی آموزش به سه دسته مختلف از داده نیاز داریم.

* داده‌های آموزش: قسمت عمده‌ی داده‌ها برای عملیات آموزش استفاده می‌شود، که الگوریتم اصلی هر روش روی آن اعمال می‌شود و سعی می‌کنیم پارامتر‌هایی که همان ویژگی‌های پنهان هستند را تنظیم کنیم.
* داده‌های ارزیابی: قسمتی از داده که داده‌های ارزیابی نام دارند برای بررسی میزان بهبود یا تخریب مدل در هر تکرار استفاده می‌شوند، این داده‌ها کاملا از داده‌های آموزش جدا هستند و در مرحله‌ی آموزش اصلا به مدل نشان داده نمی‌شود و مدل تحت تاثیر این داده‌ها قرار نمی‌گیرد. در انتهای هر مرحله به کمک این داده‌ها بررسی می‌شود که تغییراتی که روی پارامتر‌های این مدل انجام شده است باعث بهتر یا بدتر شدن این مدل شده است، در صورتی که بهبودی مشاهده شود تغییرات انجام شده در این مرحله نگه داشته می‌شود و به سراغ مراحل بعدی می‌رویم اما اگه نتایج بدتر شده باشد پارامتر‌ها را به مقادیر قبلی برگردانده و مرحله‌ی بعدی را شروع می‌کنیم.
* داده‌های آزمون: قسمت دیگری از داده که داده‌های آزمون نام دارند برای بررسی کیفیت عملکرد کلی مدل به کار می‌روند. این داده‌ها نیز کاملا از داده‌های آمورش تفکیک شده‌اند و در زمان آموزش روی مدل تاثیری نمی‌گذارند و کلا در هیج یک از مراحل آموزش استفاده نمی‌شوند و تنها پس از آموزش مدل استفاده می‌شوند تا کیفیت مدل آموزش دیده شده را بررسی کنند.

در این تحقیق پایگاه دانش Freebase15k به صورت شکل – تقسیم و استفاده شده است، به این صورت که برای قسمت آموزش 483,142 حقیقت استفاده کرده‌ایم، برای قسمت ارزیابی 50k حقیقت استفاده کرده‌ایم و برای قسمت آزمون 59k حقیقت استفاده کرده‌ایم.



## مراحل آموزش مدل

همانطور که در بخش قبل گفته شد پایگاه دانش را به سه قسمت آموزش، ارزیابی و آزمون تقسیم می‌کنیم. نحوه‌ی آموزش کلی همه‌ی روش‌های مبتنی بر ویژگی‌های پنهان در سه مرحله انجام می‌شود: ۱- مرحله‌ی آموزش ۱-مرحله‌ی ارزیابی ۳- مرحله‌ی آزمون. مراحل آموزش و ارزیابی به صورت تکراری و معمولا با تکرار بالا انجام می‌شوند و در هر تکرار بررسی می‌شود که بهبودی اتفاق افتاده است یا خیر، اگه بهبودی داشتیم نتایج این مرحله تکرار را نگه داشته و مرحله‌ی تکرار بعدی را شروع می‌کنیم و اگر بهبودی اتفاق نیافتاده بود تنایج این مرحله را تاثیر نمی‌دهیم و مرحله‌ی تکرار بهدی را شروع می‌کنیم.

در ادامه این سه مرحله را توضیح می‌دهیم.

مرحله‌ی آموزش:

همانطور که قبل‌تر توضیح داده شد آموزش به صورت تکراری انجام می‌شود و در هر تکرار الگوریتم آموزش روش مورد نظر روی قسمتی یا همه‌ی داده‌های مجموعه‌ی آموزش انجام می‌شود. در این قسمت سعی می‌شود که پارامتر‌های همه‌ی ویژگی‌ها جهت‌دهی شده و آموزش ببینند تا کمترین خطا در پاسخ به سوالاتی که از مدل پرسیده می‌شود را داشته باشند.

مرحله‌ی ارزیابی:

پس از هر تکرار مدل آموزش دیده شده را روی داده‌های ارزیابی اجرا می‌کنیم و معیار‌هایی که برای بررسی کیفیت مدل در نظر گرفته‌ایم را بدست می‌آوریم و از روی آن میزان بهبود مدل در این تکرار را بررسی می‌کنم.

برای مثال فرض کنید که حقیقت زیر در مجموعه داده‌های ارزیابی وجود دارد و در مرحله‌ی آموزش مدل این حقیقت مشاهده نشده است:

(WALL-E, has\_genre, Fantasy)

این حقیقت به این معناست که «ژانر فیلم WALL-E فانتزی است». در مرحله‌ی ارزیابی موجودیت اول یا آخر این حقیقت را حذف می‌کنیم و قسمت حذف شده را از مدل سوال می‌پرسیم و انتظار داریم که قسمت حذف شده را حدس بزند. سوالی که از این مدل پرسیده می‌شود به این صورت است:

(WALL-E, has\_genre, ?)

به این معنی که «ژانر فیلم وال-ای چیست؟».

در مدل‌های مبتنی بر ویژگی‌های پنهان پاسخ به اینگونه سوالات به صورت مجموعه‌ی مرتب شده‌ی همه موجودیت‌هاست. یعنی این مدل میزان نزدیکی همه‌ی موجودیت‌ها با پاسخ این سوال را بررسی می‌کند و به ترتیب نزدیک به دور مرتب می‌کند و به عنوان پاسخ به ما می‌دهد. برای مثال پاسخ به سوال بالا به صورت زیر می‌باشد:



همانطور که می‌بینیم مدلی که این سوال از آن پرسیده شده پاسخ درست را در ششمین حدس به ما داده است.

در مرحله‌ی ارزیابی نیاز به معیار‌هایی داریم که بررسی کنیم که مدل با توجه به این معیار‌ها بهبود داشته یا خیر. در این تحقیق از دو معیار زیر استفاده شده است:

* رتبه‌ی میانگین (Mean rank): میانگین رتبه‌ی جواب‌های درستی که مدل داده است.
* حدس زیر ۱۰: درصد سوالاتی که پاسخ درست مدل به آن زیر رتبه‌ی ۱۰ بوده است که در ادامه به آن hit@10 می‌گوییم.

همانطور که قبل‌تر توضیح داده شد ۵۰هزار حقیقت در دسته‌ی ارزیابی وجود دارد، ما پس از هر تکرار الگوریتم دو معیار بالا را به دست آورده و میزان بهبود مدل را اندازه می‌گیریم. پس از پرسیدن این ۵۰ هزار سوال میانگین رتبه‌ای که جواب‌های درست داشته معیار اول را به ما می‌دهد و درصد سوالاتی که جواب درست آن زیر رتبه‌ی ۱۰ بوده است معیار دوم را به ما می‌دهد.

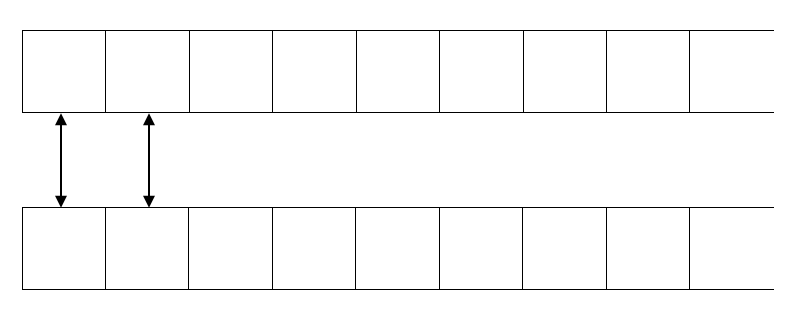
مرحله‌ی آزمون:

پس از انجام کامل مراحل آموزش و ارزیابی و متوقف شدند الگوریتم از داده‌ای آزمون که در بخش – توضیح دادیم و معیار‌های کیفیت که در بخش – توضیح دادیم استفاده می‌کنیم و کیفیت مدل را بررسی می‌کنیم. در این بخش هم مانند بخش ارزیابی داده‌ها به مدل در حال آموزش نشان داده نشده و برای مدل جدید هستند. در این مرحله هم یک قسمت از هر حقیقت موجود در داده‌های آزمون را حذف کرده و آن را از مدل سوال می‌پرسیم، دقیقا به مانند مرحله‌ی ارزیابی. پس از پرسیدن سوالات دو معیار رتبه‌ی میانگین و hit@10 را بدست می‌آوریم که این دو معیار نشان دهنده‌ی میزان کیفیت و دقت روش است.

# روش‌های موجود

۱- روش‌های خطی

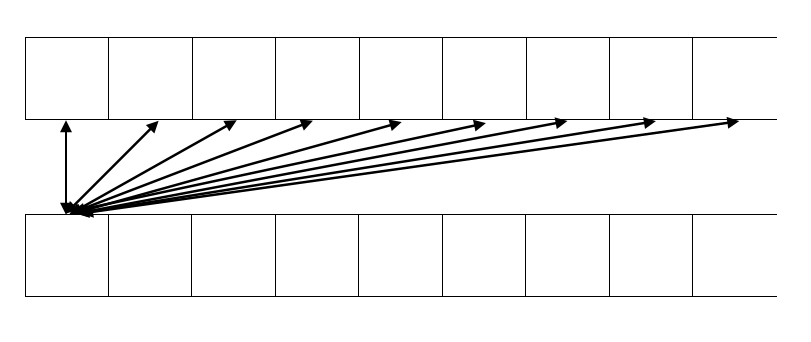
همانطور که در بخش – گفتیم در روش‌های مبتنی بر ویژگی‌های پنهان موجودیت‌ها و رابطه‌ها به صورت بردار‌هایی در یک فضای n بعدی تبدیل می‌شوند که به کمک معیارهای مختلف شباهت، رابطه‌ی بین دو موجودیت را بدست می‌آوریم. در روش‌های خطی در زمان آموزش و بررسی میزان شباهت بردارها را به صورت خطی با یکدیگر مقایسه می‌کنیم، به این صورت که هر اندیس از بردار موجودیت اول را با اندیس متناظر آن در بردار رابطه یا موجودیت دیگر بررسی می‌کنیم، شکل +.



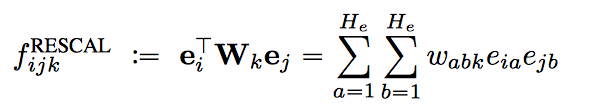
با توجه به نتایج ضعیفی که روش‌های این دسته در آموزش مدل و پیش‌بینی لینک بدست آورده اند به این روش‌ها نمی‌پردازیم و به همین معرفی کفایت می‌کنیم.

# ۲- روش‌های دوخطی – رسکال

روش RESCAL [] یک روش ویژگی‌های پنهان رابطه‌ای[[3]](#footnote-3) است که حقیقت‌های پایگاه دانش را به صورت تراکنش‌های بین جفت ویژگی‌های پنهان در نظر می‌گیرد. یعنی بر خلاف روش‌های خطی هر ویژگی پنهان از هر بردار را با همه‌ی ویژگی‌های پنهان دیگر بردار بررسی می‌کند، به صورت شکل +. به همین دلیل این روش را روش دوخطی نیز می‌نامیم.



در این روش امتیاز هر سه‌تایی را از رابطه‌ی + بدست می‌آوریم که در آن He تعداد ابعاد بردار در نظر گرفته شده برای موجودیت‌ها است (تعداد ویژگی‌های پنهان هر موجودیت). Wk یک بردار وزن با ابعاد He\*He است که هر اندیس wabk نشانگر این است که ویژگی پنهان a و b در رابطه‌ی k چقد با هم کنش می‌کنند.



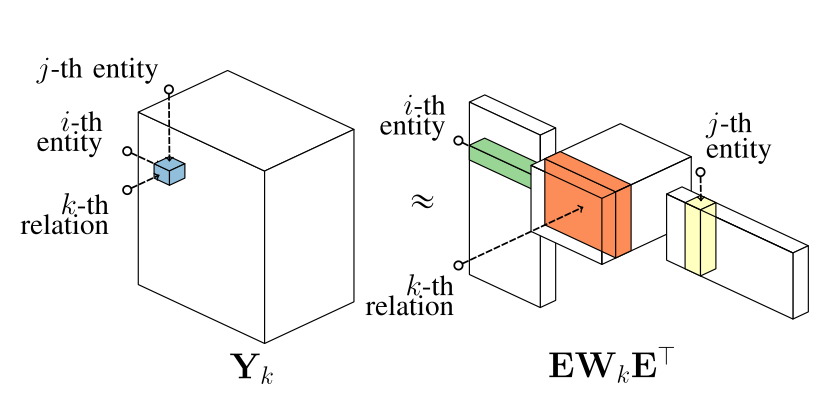
همانطور که در رابطه‌ی + مشاهده می‌شود در این روش هر رابطه به یک ماتریس تبدیل شده و هر موجودیت به یک بردار، اگر حقیقت (a,k,b) به این معنی که موجودیت a رابطه‌ی k را با b دارد را در نظر بگیریم و بخواهیم بررسی کنیم که این رابطه برقرار هست یا خیر، احتمال وجود این رابطه را از روی امتیازی که تابع امتیاز + به ما می‌دهد بدست می‌آوریم. این امتیاز به این صورت محاسبه می‌شود که بردار موجودیت a در ماتریس مربوط به k ضرب شده و پس از آن در بردار b ضرب می‌شود که نتیجه‌ی آن یک مقدار حقیقی است که امتیاز این حقیقت را به ما می‌دهد.

در ادامه به برخی از نکات مورد توجه این مدل به صورت موردی اشاره می‌کنیم.

آموزش رابطه‌ای[[4]](#footnote-4) از طریق بازنمایی مشترک[[5]](#footnote-5): در رابطه‌ی + هر موجودیت به صورت یک بردار بازنمایی شده است بدون توجه به اینکه در قسمت اول حقیقت می‌آید یا قسمت دوم آن. همچنین این موجودیت‌ها برای همه‌ی رابطه‌ها یک بازنمایی مشترک دارند و در هر رابطه نیاز به تعریف جدید ندارند. برای مثال موجودیت i در قسمت اول حقیقت xijk با رابطه‌ی k آمده است و همین موجودیت در رابطه‌ی xpiq به عنوان موجودیت دوم در رابطه‌ی q ظاهر شده است. هر دو تابع امتیاز fijk = eiWkej و fpiq = epWqei از یک بازنمایی برای موجودیت i (ei) استفاده می‌کنند. بنابراین همه‌ی پارامتر‌ها به صورت مشترک[[6]](#footnote-6) آموزش دیده می‌شوند و این بازنمایی مشترک باعث می‌شود که اطلاعات روی همه‌ی حقیقت‌ها به وسیله‌ی بازنمایی موجودیت‌ها و ماتریس وزن‌دار رابطه‌ها پخش شوند و بتوانیم وابستگی‌های جهانی[[7]](#footnote-7) در داده‌ها را تشخیص دهیم.[1]

ارتباط معنایی بردارها[[8]](#footnote-8): خاصیت بازنمایی مشترک در این روش کمک می‌کند که میزان شباهت موجودیت‌ها در فضای رابطه‌ای[[9]](#footnote-9) نیز بدست بیاید. برای مثال موجودیت‌هایی که با رابطه‌های مشابه به موجودیت‌های مشابه متصل هستند به یکدیگر شبیه هستند. به عنوان نمونه اگر بازنمایی ei و ep شبیه به هم باشد، تابع امتیاز fijk و fpjk باید مقادیر نزدیک به هم داشته باشند پس موجودیت‌ها با تعداد زیادی رابطه‌ی مشترک بازنمایی یکسانی خواهند داشت. این خصوصیت می‌تواند در بازنمایی[[10]](#footnote-10) و خوشه‌بندی‌های با مقیاس بالای[[11]](#footnote-11) موجودیت‌ها روی داده‌های رابطه‌ای مورد استفاده قرار گیرد.[63,64,1]

ارتباط با عامل‌بندی تنسور[[12]](#footnote-12): این روش شباهت زیادی به روش‌های استفاده شده در سیستم‌های توصیه‌گر[66] و عامل‌بندی تنسور سنتی دارد[۶۷]. ضرب ماتریس که در معادله‌ی + معرفی کردیم می‌تواند به صورت Fk = EWkET نوشته شود که در آن F∈RNe\*Ne ماتریسی است که همه‌ی امتیازات مربوط به رابطه‌ی k را در خود جا داده است و سطر iام از ماتریس E∈RNe\*He بازنمایی موجودیت ei است که برداری از ویژگی‌های پنهان این موجودیت است. در شکل + این تبدیل ماتریس به تنسور نمایش داده شده است.[1]



برازش مدل: اگر بخواهیم از یک مدل آماری استفاده کنیم، پارامتر‌های این روش می‌توانند به صورت یک مدل کمینه‌سازی برمبنای گرادیان[[13]](#footnote-13) تخمین زده شوند مانند گرادیان نزولی تصادفی (SGD)[[14]](#footnote-14) [68]. این روش می‌توانیم به صورت یک روش بر مبنای امتیاز حل کرد که می‌تواند از پارامتر‌های مدل را به صورت بسیار بهینه‌ای تخمین بزند: با توجه به ساختار تنسور که در بالا توضیح دادیم و همچنین با توجه به تنک بودن داده‌های موجود، نشان می‌دهیم که روش رسکال می‌تواند به کمک توالی‌[[15]](#footnote-15)ای از بروزرسانی‌های بسته‌ی کارآمد[[16]](#footnote-16) محاسبه شود.[63,64] به صورت تحلیلی می‌توان نشان داد که در این راهکار با هر بروزرسانی در E و Wkبه صورت خطی با تعداد موجودیت‌ها Ne، تعداد رابطه‌ها Nr و تعداد حقیقت‌های مشاهده شده توسط مدل رشد می‌کند.[64]

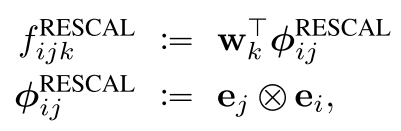
پیش‌بینی مجزا: در رابطه‌ی + احتمال وجود یک رابطه از روی یک ضرب ماتریسی ساده از مرتبه‌ی O(He2) بدست می‌آید. بنابراین، زمانی که پارامتر‌های مدل تخمین زده شدند، پیچیدگی محاسبات برای پیش‌بینی امتیاز یک حقیقت فقط به تعداد ویژگی‌های پنهان وابسته است و مستقل از اندازه‌ی کل گراف است. با این حال به لطف بازنمایی مشترک که قبل‌تر توضیح داده شده، این مدل می‌تواند در زمان تخمین پارامتر‌ها، وابستگی‌های جهانی بین موجودیت‌ها و رابطه‌ها را فهمیده و در فرایند آموزش تاثیر دهد.[1]

نتایج یادگیری رابطه‌ای: رسکال در زمینه‌های مختلف مدل‌های یادگیری مدرن[[17]](#footnote-17) مورد استفاده قرار گرفته است. مثلا در [63] نشان داده شده است که رسکال موفق شده در پیش‌بینی رابطه نتایج نزدیک یا بهتر روی چندین مجموعه داده‌ی معیار نسبت به روش‌های Markov Logic Networks [70] و the Infinite (Hidden) Relational model [71, 72] و Bayesian Clustered Tensor Factorization [73] بدست آورد. همچنین رسکال برای پیش‌بینی لینک روی کل پایگاه‌دانش مانند YAGO و DBpediaمورد استفاده قرار گرفته است. [64, 74]فارق از پیش‌بینی لینک، رسکال در روش‌های یادگیری رابطه‌ای تصادفی (SRL)[[18]](#footnote-18) مانند بازنمایی موجودیت‌ها[[19]](#footnote-19) و خوشه‌بندی بر مبنای لینک[[20]](#footnote-20) عملکرد موفقی داشته است. برای مثال این روش عملکرد موفقی در دسته‌بندی نویسندگان، ناشران و سالن‌های انتشار روی مجموعه داده‌های ناشران داشته است[63,65]. علاوه بر این، ارتباط معنایی موجودیت‌ها که در این روش محاسبه شد در ایجاد طبقه‌بندی[[21]](#footnote-21) به کمک خوشه‌بندی سلسله مراتبی[[22]](#footnote-22) روی داده‌های دسته‌بندی نشده[[23]](#footnote-23) به کار برده شده است [75].

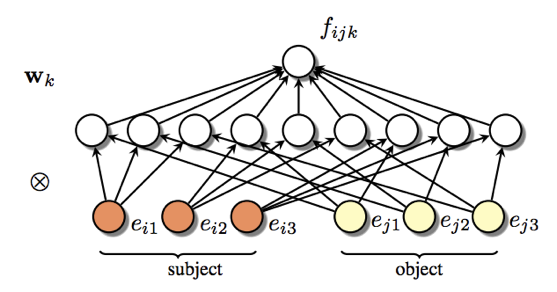
RESCAL is a tensor factorization approach to relational learning which is designed to account for the inherent structure of dyadic relational data. In doing so, our approach is able to perform collective learning via the latent components of the factorization. The results on various datasets as well as the runtime performance are very competitive and show that tensors in general and RESCAL specifically are promising new approaches to relational learning. Cur- rently we intend to investigate different extensions to our approach. In order to obtain highly scalable solutions, we are looking into distributed versions of RESCAL as well as a stochastic gradient descent approach to the optimization problem. Furthermore, to improve both the predictive per- formance and the runtime behaviour of RESCAL, we also plan to exploit constraints like typed relations while com- puting the factorization.

# ۳- روش‌های ادراک چندلایه‌ای[[24]](#footnote-24)

می‌توانیم رسکال را به صورت مدلی که برای هر حقیقت یک بازنمایی تولید می‌کند و از روی این بازنمایی‌ها وجود یا عدم وجود این حقیقت‌ها را پیش‌بینی می‌کند تفسیر کنیم. به طور خاص می‌توانیم رسکال را به صورت رابطه‌های + و + بازنویسی کنیم.



در روابط + ضرب داخلی ei و ej که در محاسبه‌ی مجموع امتیاز‌ها شرکت می‌کردند را از مجموع بیرون کشیده و به صورت ضرب خاجی نوشتیم. بنابراین رسکال بازنمایی جفت موجودیت i و j را به صورت ضرب تنسور ویژگی‌های پنهان این دو موجودیت بدست آورد رابطه‌ی + و وجود حقیقت xijk را از روی بازنمایی جفت موجودیت‌ها (ϕij) و ماتریس وزن‌دار رابطه (wk) بدست آورد. این تقسیم‌بندی رابطه‌ی رسکال در شکل + به تصویر کشیده شده است. برای توضیح بیشتر در مورد ایجاد بازنمایی پنهان به وسیله‌ی ضرب تنسور به [88, 89, 90] مراجعه شود.

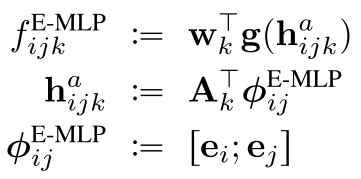


از آنجایی که ضرب تنسور تراکنش بین همه‌ی جفت موجودیت‌ها را مدل می‌کند، وقتی تعداد ویژگی‌های پنهان زیاد باشد، رسکال در این راهکار به تعداد زیادی پارامتر نیاز پیدا خواهد کرد. این موضوع می‌تواند باعث مشکل در مقیاس‌پذیری روش روی پایگاه‌های دانش بزرگ با تعداد زیاد رابطه شود.

در ادامه‌ی این بخش به روش‌های ادراک چندلایه‌ای که به شبکه‌های عصبی پیش‌خور[[25]](#footnote-25) نیز معروف هستند می‌پردازیم. این راهکار این امکان را به ما می‌دهد که مدل‌های جایگزینی برای ساختن بازنمایی حقیقت‌ها در نظر بگیریم و همچنین بتوانیم از توابع غیر خطی در پیش‌بینی وجود لینک‌ها استفاده کنیم.

مدل [[26]](#footnote-26)E-MLP:

در ابتدا مدل Entity-MLP را معرفی می‌کنیم. تابع امتیاز این مدل به صورت معادلات + و + و + محاسبه می‌گردد.



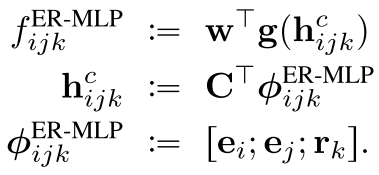
در روابط فوق g(u) = [g(u1),g(u2),. ..] یک تابع g است که بر روی تک تک المان‌های بردار u اعمال می‌شود.که می‌تواند یک تابع غیر خطی باشد، مانند g(u) = tanh(u) .

در این روابط ha یک لایه‌ی مخفی[[27]](#footnote-27) اضافه شده است که ماتریس وزن‌دار دیگر روی بازنمایی موجودیت‌ها اعمال می‌کند. در اصل ما در رابطه‌ی + دو موجودیت ei و ej را ترکیب کردیم و هیچ کنشی بین این دو موجودیت محاسبه و تاثیر داده نشده است، بنابراین نیاز به یک ماتریس وزن‌دار برای محاسبه‌ی تاثیر این دو موجودیت نیاز بود و ha وارد معادله شد.

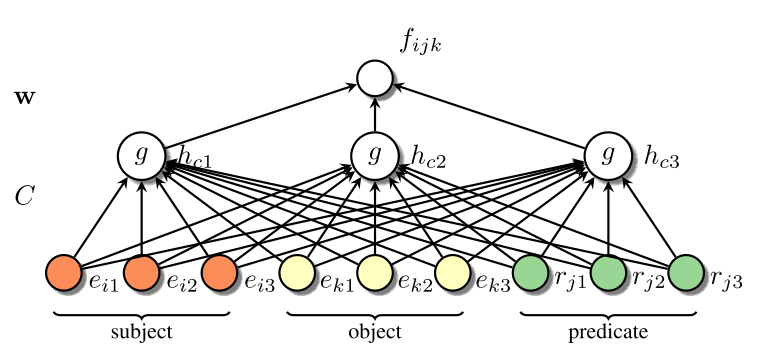
بزرگترین تفاوتی که این روش با روش‌های ضرب تنسور مانند رسکال دارد این است که در اینجا بجای محاسبه‌ی همه‌ی تراکنش‌های ممکن بین دو موجودیت، فقط تراکنش‌های موجود در Ak مورد بررسی قرار می‌گیرد. این راهکار به طور قابل ملاحضه‌ای تعداد پارامتری که باید آموزش دیده شوند را کاهش می‌دهد.

روش ER-MLP[[28]](#footnote-28):

یکی از اشکالات روش E-MLP این است که باید برای هر رابطه‌ی ممکن، یک بردار wk و یک ماتری Ak تعریف شود، که تعداد پارامتر‌ها را زیاد می‌کند. برای حل این مشکل روش ER-MLP معرفی شده است که در این روش رابطه را نیز به صورت برداری در کنار موجودیت‌ها در نظر می‌گیریم، و بجای ماتریس Ak می‌توانیم از یک بردار ثابت C استفاده کنیم. بنابراین روابط ER-MLP به صورت + و + و + خواهند بود.



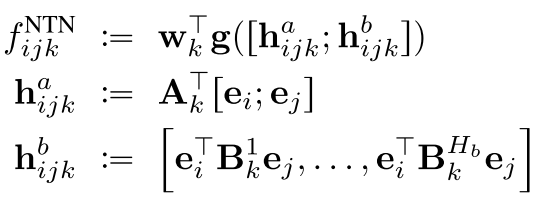
توجه کنید که در این روش از یک بردار وزن‌دار جهانی برای همه روابط استفاده شده است و بردار C تعریف شده مستقل از رابطه‌ی rk است که باعث کاهش تعداد پارامتر‌های مساله می‌گردد. نحوه‌ی کار روش ER-MLP در شکل + نمایش داده شده است.



در [91] نشان داده شده است که روش‌های MLP کلماتی که قرابت معنایی[[29]](#footnote-29) دارند را به درستی نزدیک به یکدیگر تشخصی می‌دهند در حالی که برای چنین کاربردی آموزش دیده نشده اند. همچنین در [28] این ویژگی نشان داده شده است، برای مثال به کمک MLP مساله‌ی نزدیک‌ترین همسایه[[30]](#footnote-30) برای بازنمایی پنهان چندین رابطه‌ی انتخاب شده از پایگاه‌دانش Freebase را حل کرده اند. در نتایج می‌توان مشاهده کرد که روابطی که ارتباط معنایی دارند نزدیک به یکدیگر قرار گرفته‌اند.

شبکه‌های عصبی تنسور[[31]](#footnote-31)

در [92] با ترکیب روش‌های ادراک چند لایه‌ای (MLP) و روش‌های دوخطی (Bilinear) روش جدیدی به نام شبکه‌های عصبی تنسور (NTN) معرفی شده است. روابط محاسبه‌ی تابع امتیاز این روش به صورت + و + و + است.



در اینجا Bk یک تنسور است، که در آن میزان رابطه‌ی بین جفت موجودیت‌ها نگاشت شده است. Hijkb را یک لایه‌ی مخفی دوخطی[[32]](#footnote-32) می‌نامیم، زیرا هم به صورت یک لایه‌ی مخفی از مدل‌های MLP هست و هم به کمک یک تابع وزن‌دار میزان کنش جفت موجودیت‌ها را مشخص می‌کند مانند آنچه در روش رسکال توضیح دادیم.

با توجه به معادله‌های معرفی شده در +و+و+ مشخص است که این روش مشکلات مقیاس پذیری که در هر دو روش E-MLP و RESCAL وجود داشت را دارد. همچنین در [95] و [28] نشان داده شده است که این روش به بیش‌برازش[[33]](#footnote-33) میل می‌کند (حداقل روی مجموعه داده‌هایی که در این مقالات استفاده شده است)[1].

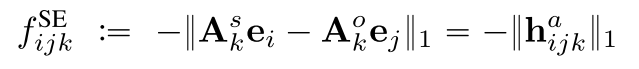
# ۳- روش‌های فاصله‌ی پنهان[[34]](#footnote-34)

کلاس دیگری از مدل‌ها، مدل‌های فاصله‌ی پنهان هستند (همچنین در تحلیل شبکه‌های اجتماعی به مدل‌های فضای پنهان معروفند) که احتمال وجود رابطه بین بین موجودیت‌ها را از فاصله‌ی بین بازنمایی پنهان آن‌ها در فضا می‌سنجد: موجودیت‌ها با یکدیگر رابطه دارند اگر بازنمایی پنهان آن‌ها با یک معیار فاصله نزدیک به هم باشد [1].

در [96] مدلی برای داده‌های تک-رابطه‌ای[[35]](#footnote-35) برای اولین بار راهکاری در این زمینه معرفی شده است. این راهکار احتمال وجود لینک در شبکه‌های اجتماعی را با تابع امتیاز f(ei,ej)= -d(ei, ej) محاسبه می‌کند، که در آن d(.,.) یک تابع اندازه‌گیری فاصله‌ی دلخواه مانند فاصله‌ی اقلیدسی[[36]](#footnote-36) است.

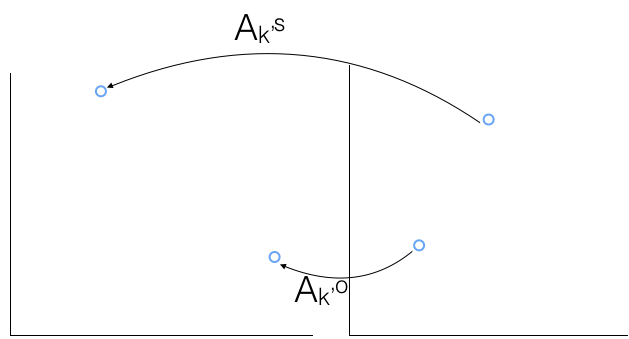
**مدل Structured embedding**

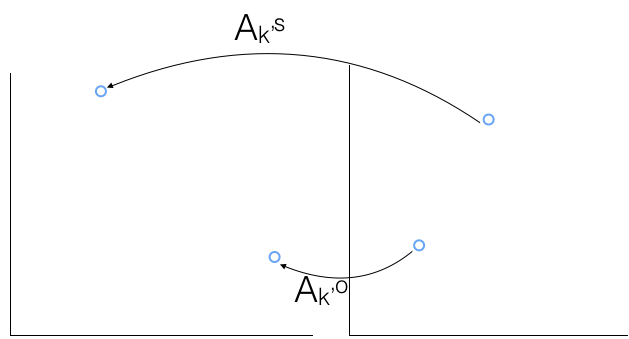
در [93] مدلی به نام structured embedding (SE) معرفی شده که در آن ایده‌ای که در بند قبل توضیح داده شد برای داده‌های چند-رابطه‌ای[[37]](#footnote-37) گسترش داده شده است. در این روش تابع امتیاز برای حقیقت‌های xijk به صورت + مدل شده است.



در رابطه‌ی + Ak از ماتریس‌های مربوط به موجودیت‌های اول و دوم [Aks, Ako] تشکیل شده است. ماتریس‌های Aks و Ako بازنمایی ویژگی‌های پنهان موجودیت‌ها را به فضای مخصوص به رابطه‌ی k منتقل می‌کند. این انتقال به صورتی آموزش دیده می‌شود که جفت رابطه‌هایی که برقرار هستند، نسبت به جفت رابطه‌هایی که برقرار نیستند به یکدیگر نزدیکتر باشند.

در شکل + مشاهده می‌شود که در صورتی که بین دو موجودیت i و j رابطه‌ی k برقرار باشد، پس از نگاشت این دو موجودیت توسط ماتریس‌های نگاشت Ako و Aksاین دو موجودیت به یکدیگر نزدیکتر شده اند. همینطور در شکل + مشاهده می‌شود که در صورتی که رابطه‌ی k’ بین دو موجودیت i و j برقرار نباشد، پس از نگاشت این دو موجودیت توسط ماتریس‌های نگاشت Ak’s و Ak’o دو موجودیت در فاصله‌ی بیشتری از یکدیگر قرار می‌گیرند.





یکی از نکات منفی که در این روش به چشم می‌خورد یادگیری ماتریس‌های وزن‌دار جدا برای موجودیت‌های اول و دوم است، که باعث افزایش تعداد پارامتر مورد نیاز برای آموزش است.

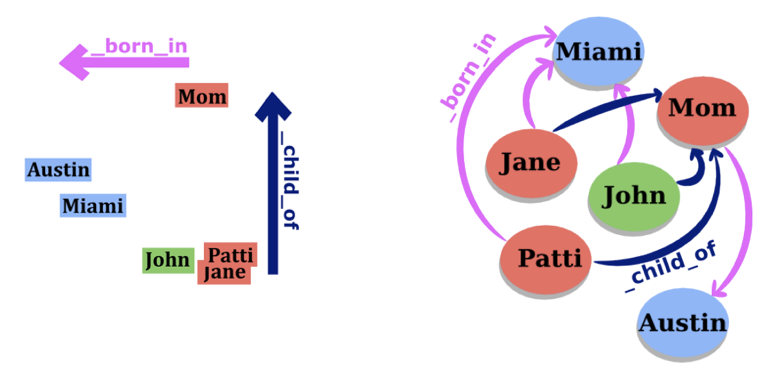
**مدل [[38]](#footnote-38)TransE**

در [94] مدلی برای حل مشکل تعداد پارامتر زیاد در روش SE مطرح شده است که بجای اینکه از ماتریس‌های Aks و Ako برای تاثیر دادن رابطه‌ها استفاده شود، رابطه را یک بردار همانند بردار موجودیت‌ها در نظر گرفته شده و به عنوان یک آفست در کنار موجودیت اول در محاسبات از آن استفاده می‌کند. [1] به طور خاص امتیاز حقیقت xijk از رابطه‌ی + بدست می‌آید.

../../../../../Desktop/Screen%20Shot%201396-03-13%20at%2001.

در این روش رابطه‌ها به صورت یک بردار انتقال استفاده می‌شوند، به این صورت که فقط روی موجودیت اول اعمال می‌شوند و در صورت وجود رابطه بین این دو موجودیت، موجودیت اول را به موجودیت دوم نزدیک می‌کنند. در شکل + یک گراف را مشاهده می‌کنیم که در آن روابط فرزندی و مکان تولد بین ۶ موجودیت نمایش داده شده است. در شکل + یک مثال از اعمال روش TransE روی این گراف را مشاهده می‌کنیم. مشاهده می‌شود که موجودیت‌ها بردارهایی ثابت در نظر گرفته شده‌اند و بردار‌های رابطه به صورت یک بردار انتقال به تصویر کشیده شده است که موجودیت‌ها را به هدف‌های مورد نظر نزدیک می‌کنند.

مثلا اگر john را توسط بردار Child\_of منتقل کنیم به موجودیت Mom نزدیک می‌شود که حقیقت (john, Child\_of, Mom) را تشکیل می‌دهد. اما اگر همین موجودیت را توسط بردار رابطه‌ی born\_in منتقل کنیم، به موجودیت Miami نزدیک می‌شود که حقیقت (john, born\_in, Miami) را تشکیل می‌دهد.



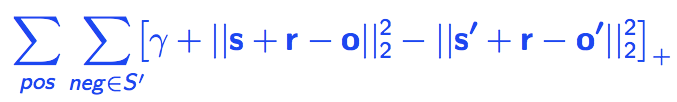
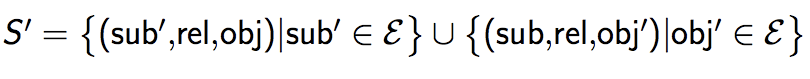
روش TransE این مساله را به صورت یک مساله‌ی کمینه‌سازی حل می‌کند به این صورت که سعی می‌کند فاصله‌ی مجموع بردار‌های موجودیت اول و رابطه (e1+r) را با موجودیت دوم (e2) کمینه کند. همانطور که قبل‌تر گفتیم تابع امتیاز در این روش‌ها یک تابع فاصله است، پس در اینجا هدف کاهش فاصله بین (e1+r) و e2 است که بصورت + نوشته می‌شود.

d(e1, r, e2) = ||e1 + r – e2||

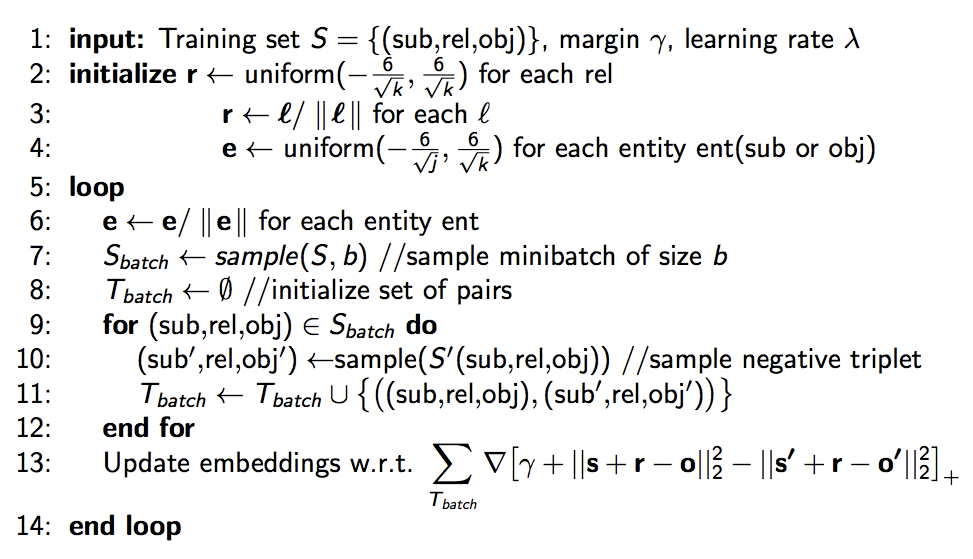
برای رسیدن به نتیجه‌ی بهینه نوآوری دیگری که در این روش معرفی شده است این است که تابع کمینه‌سازی را به گونه‌ای تغییر داده است که فاصله‌ی نمونه‌های اشتباه را نیز زیاد کرده است.

در زمانی آموزش پارامتر‌ها علاوه بر در نظر گرفتن این نکته که باید تابع فاصله‌ی نمونه‌های مثبت موجود در پایگاه دانش کمینه شود، سعی شده تا فاصله‌ی نمونه‌های منفی را نیز افزایش دهد. از انجایی که پایگاه‌های دانش مورد استفاده از فرض جهان‌باز پیروی می‌کنند وقتی حقیقت (e1, r, e2) در پایگاه دانش موجود نیست نمی‌توانیم نتیجه بگیریم که این سه‌گانه غلط است و آن را نمونه‌ی منفی تلقی کنیم.

در این روش برای ساختن نمونه‌های منفی، نمونه‌های مثبت مورد استفاده قرار گرفته‌اند به این صورت که یک‌بار موجودیت اول حذف شده و یک موجودیت تصادفی جای آن قرار داده شده است و یک‌بار موجودیت دوم حذف شده و یک موجودیت تصادفی جای آن قرار گرفته است، به این صورت به ازای هر نمونه‌ی مثبت دو نمونه‌ی منفی نیز تولید شده است (رابطه‌ی +). در رابطه‌ی + مشاهده می‌شود که علاوه بر کاهش فاصله‌ی موجودیت‌های مثبت، یک جریمه هم برای نمونه‌های منفی در نظر گرفته شده است و همچنین یک حاشیه γ بین نمونه‌های مثبت و منفی اضافه شده است.

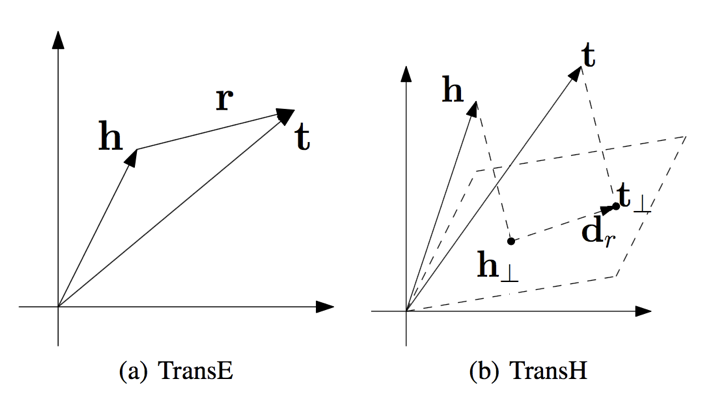


در این روش نیز مانند روش رسکال از SGD استفاده شده است که هم امکان آموزش به کمک نمونه‌برداری دسته‌ای[[39]](#footnote-39) را فراهم می‌کند و هم از مشکل گیر کردن در نقاط بهینه‌ی[[40]](#footnote-40) محلی جلوگیری می‌کند. الگوریتم کامل این روش در + آمده است.



**مدل [[41]](#footnote-41)TransH**

فهم روابط ۱-به-۱ و ...



نتایج روش‌های موجود

در این بخش نتایج بدست آمده از روش‌های معرفی شده در بخش قبل را مورد بحث و بررسی قرار می‌دهیم. روش‌هایی که در این پایان‌نامه مورد بررسی و بهبود قرار گرفته‌اند ۱- روش RESCAL از روش‌های دوخطی ۲- روش NTN از روش‌های ادراک چندلایه‌ای ۳- روش SE ۴- روش TransE و ۵- روش TransH از روش‌های فاصله‌ی پنهان هستند.

در ادامه آزمایشاتی روی این ۵ روش انجام شده است و این روش‌ها از نظر مقیاس‌پذیری و میزان کیفیت خروجی که به کمک معیار‌های رتبه‌ی میانگین و hit@10 محاسبه می‌شود مورد بررسی قرار گرفته‌اند. همه‌ی این آزمایشات روی پایگاه‌دانش Freebase15k انجام شده است که شامل حدود ۱۵هزار موجودیت، ۱۳۰۰ رابطه و در مجموع ۶۰۰هزار حقیقت است. تقسیم‌بندی داده‌های آموزش، ارزیابی و آزمون به شکلی که در + توضیح داده شد انجام شده است.

روش رسکال:

تعداد پارامتر‌ها: در این روش برای هر موجودیت یک بردار dبعدی و برای هر رابطه یک ماتریس وزن‌دار d\*d بعدی در نظر گرفتیم. پس تعداد پارامتر‌هایی که در این روش نیاز است که آموزش دهیم از مرتبه‌ی O(ned + nrd2) است. در آزمایشات تعداد ابعادی که بهترین نتیجه را بدست می‌دهد d=250 است. اگر ابعاد مساله را ۲۵۰ در نظر بگیر باید به تعداد ۸۸میلیون پارامتر آموزش داده شود (ned + nrd2 = 15k\*250 + 1.3k\*250\*250 = 88m).

کیفیت خروجی: پس از اعمال مدل آموزش دیده‌ی رسکال روی مجموعه داده‌ی آزمون ۵۰هزار حقیقتی، میزان ۴۲.۱٪ پاسخ‌های داده شده توسط این مدل زیر رتبه‌ی ۱۰ بوده (hit@10) و میانگین رتبه‌ی کل پاسخ‌هایی که داده شده (mean rank) ۶۸۳ است.

روش NTN

تعداد پارامتر‌ها: در این روش نیز برای هر موجودیت یک بردار dبعدی در نظر گرفته شده و برای هر رابطه یک ماتریس Bk و دو بردار Ak و wk در نظر گرفته می‌شود که در مجموع برای هر رابطه d3 بعد در نظر گرفته می‌شود و تعداد کل پارامتر‌هایی که باید آموزش دیده شوند از مرتبه‌ی O(ned + nrd3) است. تعداد پارامتری که در آزمایشات برای این روش نتیجه‌ی مناسبی بدست می‌دهد d=50 است. برای این تعداد ویژگی پنهان تعداد کل پارامتری که باید آموزش داده شوند تقریبا ۱۶۵میلیون پارامتر می‌شود که نسبت به روش رسکال با اینکه تعداد ابعاد بردارها بسیار کمتر است، تقریبا دو برابر است(ned + nrd2 = 15k\*50 + 1.3k\*50\*50\*50 163,250k).

کیفیت خروجی: مقدار hit@10 در این روش ۲۷٪ و میانگین رتبه‌ی همه‌ی پاسخ‌های این روش ۱۶۴ است. همانطور که در معرفی این روش گفته شد این روش به سرعت به بیش‌برازش میل می‌کند و تاثیر این امر در میزان hit@10 مشخص است اما میانگین رتبه این روش به میزان خوبی از روش رسکال بهتر است که نشان می‌دهد روش رسکال برای سوالاتی که به میزان کافی نمونه‌ی آموزش ندیده است و نتایج خیلی پرتی بدست می‌دهد ولی برای ۴۲.۱٪ حقیقت که روابط بین آن‌ها به خوبی آموزش دیده شده است و پاسخ‌های درست در ۱۰ پاسخ اول بوده است.

روش Structured Embedding

تعداد پارامتر: در این روش برای هر موجودیت یک بردار dبعدی و برای هر رابطه دو ماتریس d\*d بعدی آموزش دیده می‌شود، یکی برای انتقال موجودیت اول و دیگری برای انتقال موجودیت دوم. بنابراین تعداد پارامتر این روش از مرتبه‌ی O(ned + 2nrd2) است. برای آموزش این مدل نیز بردار‌هایی با اندازه d=50 کفایت می‌کند و برای آموزش این مدل باید حدود ۸میلیون پارامتر آموزش دیده شوند. (ned + 2nrd2 = 15k\*50 + 2\*1.3k\*50\*50 = 7.250k)

کیفیت خروجی: در این روش hit@10 برابر با ۳۹.۸٪ است و مقدار میانگین رتبه ۱۶۲ بوده است. مشاهده می‌شود که علی‌رقم کاهش بسیار زیادی که در تعداد پارامتر‌های مساله نسبت به دو روش قبل داشته‌ایم، نتیجه‌ی خوبی حاصل شده است و در معیار hit@10 با روش رسکال و در معیار میانگین رتبه با روش NTN رقابت می‌کند.

روش TransE:

تعداد پارامتر: در این روش تعداد پارامتر‌ها به میزان بسیار خوبی کاهش پیدا کرده است، همانطور که در معرفی این روش مشاهده شد، برای هر موجودیت یک بردار dبعدی در نظر گرفته شد و برای روابط از ماتریس استفاده نشده و به عنوان یک بردار انتقال در کنار موجودیت اول در نظر گرفته شده است که باعث می‌شود تعداد پارامتری که برای رابطه‌ها نیزا داریم نیز dبعد باشد. پس تعداد پارامتری که در این روش باید آموزش دیده شود از مرتبه‌ی O(ned + nrd) است که در مجموع تعداد پارامتر‌های مساله را بسیار کاهش می‌دهد و همین مساله روی همین پایگاه‌دانش را با حدود ۸۰۰هزار پارامتر حل می‌کند. (ned + nrd = 15k\*50 + 1.3k\*50 = 815k).

کیفیت خروجی: این روش علی‌رقم کاهش چشم‌گیری که در تعداد پارامتر‌ها داشت و مقیاس‌پذیری بسیار بالایی که ایجاد کرده است، در نتایج بدست آمده نیز بهتر از روش‌های معرفی شده‌ی قبلی کار کرده است. در این روش hit@10 برابر ۴۵.۱٪ و میانگین رتبه برابر ۱۲۵ است که مشاهده می‌شود در این روش هم جواب‌های درست بسیار بیشتر بوده و هم داده‌های پرت بسیار کمتر شده است و جواب سوال‌ها به جواب‌های منطقی نزدیک‌تر شده است.

روش TransH:

تعداد پارامتر: O(nek +2nrk)

کیفیت خروجی: خوب ☺

جمع‌بندی

در این قسمت روش‌های موجود را ...

در جدول + تعداد پارامتر‌هایی که هر یک از روش‌های بالا برای آموزش دیدن نیاز دارند، آورده شده است، همانطور که مشاهده می‌شود روش‌های Rescal و NTN تعداد پارامتر خیلی زیادی باید آموزش دهند و به مشکل مقیاس‌پذیری بر خواهند خورد و روی پایگاه‌دانش‌های بزرگ خیلی کند خواهند بود. در مقابل روش‌های SE و TransE و TransH با ایده‌هایی که برای کاهش تعداد پارامتر‌های مساله پیاده‌سازی کردن این تعداد را بسیار کاهش داده و مقیاس پذیری خوبی دارند، به صورتی که مدل TransE روی پایگاه‌دانش Freebase 1M در مدت تقریبا یک روز با hit@10 برابر با ۳۴٪ به جواب رسیده است در صورتی که روش‌های RESCAL و NTN روی این پایگاه‌دانش به جواب نمی‌رسند، که این موضوع بر عدم مقیاس‌پذیری روش‌های RESCAL و NTN تاکید می‌کند.

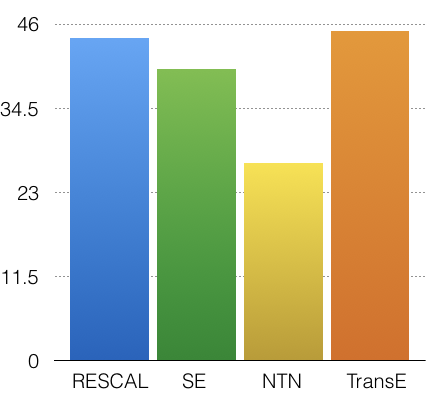
|  |  |  |
| --- | --- | --- |
| **Method** | **#Params** | **On FB15K** |
| **RESCAL** | O(ned + nrd2) | 88M (d=250) |
| **MLP (NTN)** | O(ned + nrd3) | 165M (d=50) |
| **SE** | O(ned + 2nrd2) | 8M (d=50) |
| **TransE** | O(ned + nrd) | 0.8M (d=50) |

در جدول + میزان Hit@10 و میانگین رتبه‌ی هر یک از روش‌های معرفی شده نمایش داده شده است که در این جدول مشاهده می‌شود که روش NTN در hit@10 نتیجه‌ی مطلوبی نگرفته است که در معرفی این روش اشاره کردیم که این روش به سرعت به بیش‌برازش میل می‌کند و همین امر باعث شده است که پاسخ‌های درستی در مورد مساله نداشته باشد.

نکته‌ی مهم دیگری که در این جدول دیده می‌شود کارایی خوب روش‌های SE و TransE هست که علاوه بر کاهش پارامتری که داشته‌اند، هم از نظر hit@10 و هم از نظر میانگین رتبه نتایج خوبی داشته‌اند.

در روش RESCAL هم hit@10 خوب بوده است اما رتبه‌ی میانگین مقدار بسیار بالایی نسبت به روش‌های دیگر داشته که نشان از این دارد که علی‌رقم اینکه بسیاری از روابط را درست فهمیده است و نتایج خوبی در آن‌ها گرفته است، اما بسیاری از روابط را نیز اصلا درک نکرده و پاسخ‌هایی که به سوالات در مورد این روابط داده است جواب‌های تقریبا تصادفی‌ای داشته است و رتبه‌های پرتی گرفته‌اند که باعث شده میانگین رتبه‌ی این روش چنین افزایشی داشته باشد.

|  |  |  |
| --- | --- | --- |
|  | **Hit@10** | **Mean** |
| **RESCAL** | 42.1% | 683 |
| **SE** | 39.8% | 162 |
| **NTN** | 27% | 164 |
| **TransE** | 45.1% | 125 |



نقاط قوت و ضعف روش‌های موجود

در این بخش

1. Tensor [↑](#footnote-ref-1)
2. مشخص کننده (head, relation, tail) می‌باشد که به صورت (subject, predicate, object) هم تعریف می‌شود. [↑](#footnote-ref-2)
3. Relational latent factor [↑](#footnote-ref-3)
4. Relational Learning [↑](#footnote-ref-4)
5. Shared representations [↑](#footnote-ref-5)
6. jointly [↑](#footnote-ref-6)
7. global dependency [↑](#footnote-ref-7)
8. Semantic embeddings [↑](#footnote-ref-8)
9. relational domain [↑](#footnote-ref-9)
10. resulotion [↑](#footnote-ref-10)
11. large-scale hierarchical clustering [↑](#footnote-ref-11)
12. Tensor factorization [↑](#footnote-ref-12)
13. gradient-based minimization [↑](#footnote-ref-13)
14. stochastic gradient descent [↑](#footnote-ref-14)
15. sequence [↑](#footnote-ref-15)
16. efficient close-form update [↑](#footnote-ref-16)
17. state-of-the-art [↑](#footnote-ref-17)
18. stocastic relational learning [↑](#footnote-ref-18)
19. entity resulotion [↑](#footnote-ref-19)
20. link-based clustring [↑](#footnote-ref-20)
21. taxonomies [↑](#footnote-ref-21)
22. hierarchical clusterings [↑](#footnote-ref-22)
23. uncategorized data via [↑](#footnote-ref-23)
24. Multi-layer perceptrons [↑](#footnote-ref-24)
25. feedforward neural networks [↑](#footnote-ref-25)
26. Entity-MLP [↑](#footnote-ref-26)
27. hidden layer [↑](#footnote-ref-27)
28. Entity-Relation-MLP [↑](#footnote-ref-28)
29. semantically similar [↑](#footnote-ref-29)
30. nearst neighbors [↑](#footnote-ref-30)
31. Neural tensor networks [↑](#footnote-ref-31)
32. bilinear hidden layer [↑](#footnote-ref-32)
33. overfit [↑](#footnote-ref-33)
34. Latent distance models [↑](#footnote-ref-34)
35. uni-relational data [↑](#footnote-ref-35)
36. Euclidean distance [↑](#footnote-ref-36)
37. multi-relational [↑](#footnote-ref-37)
38. Translation Embedding [↑](#footnote-ref-38)
39. Batch Sampling [↑](#footnote-ref-39)
40. local optimom [↑](#footnote-ref-40)
41. Translating on Hyperplanes [↑](#footnote-ref-41)