



پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

کاربرد شبکه های منطق مارکوف (Markov Logic Network)

در فهم زبان طبیعی

نگارنده

مریم وحدت پور

استاد راهنما: دکتر باقر باباعلی

پایان نامه برای دریافت درجه کارشناسی

در رشته علوم کامپیوتر

تیر ماه ۱۳۹۶

سپاس گزاری

سپاس فراوان از زحمات جناب آقای دکتر باباعلی که در تمامی این مراحل این پروژه من را یاری نمودند.

چکیده

هدف از درك زبان گفتار، استخراج معنا از گفتار زبان طبيعي است. هدف از استخراج اين معنا می تواند استخراج عبارات مشخص، استخراج هدف گفتار بيان شده و حتی استخراج جزئیات هدف باشد. استخراج دقيق منظور کاربر از زبان گفتاری فرایندی پیچیده است. در این پژوهش، سعی شده با استفاده از شبکه منطق مارکوف، میزان تاثیر استفاده از دادگان در زمینه درك معنا در زبان فارسي کاهش یابد. شبکه های منطق مارکوف، قاعده مندی منطق مرتبه اول را با عدم قطعیت احتمالات ترکیب می کنند. بنابراین، این شبکه ها چارچوب مناسبي را براي مدل سازي روابط معنایی پیچیده زبان گفتار و در نتیجه پیاده سازی واحد درك معنا فراهم می سازند. در این پژوهش، به منظور پیاده سازی واحد درك معنا، از مجموعه ای از دادگان که به تفکیک نقش ها روی ۳۰،۰۰۰ جمله اعمال شده اند استفاده شده است. قواعد از این مجموعه استخراج شده و با وزن دهی به هر قانون در شبکه منطقی مارکوف، احتمال میزان درستی دادگان آزمایشی سنجیده شده است. در پایان، با اجرای این شبکه بر روی دادگان آموزشی ۳۰،۰۰۰ تایی، به کارایی ۹۹ درصدی در تشخیص درستی قواعد ساختاری بر روی ۵۰۰۰ داده آزمایشی دست یافتیم.

پیش گفتار

در این مقاله، با بررسی شبکه منطق مارکوف و میزان درستی تشخیص جملات درست زبان فارسی، به ارایه توضیح مفصلی در این باره میپردازیم. زبان‌های طبیعی به هر کدام از زبان‌هایی گفته می‌شود که توسط بشر در دنیا صحبت می‌شود و خودبه‌خود پدید آمده و تکامل یافته است. معمولاً این عنوان را در مقابل زبان‌های ساختگی مثل منطق‌صوری به کار می‌برند. هدف منطق از دیدگاه بسیاری از متفکران بازسازی ساختار زبان طبیعی است. استخراج دقیق منظور کاربر از زبان گفتاری فرایندی پیچیده است. در این پژوهش، سعی شده با استفاده از شبکه منطق مارکوف، میزان تاثیر استفاده از دادگان در زمینه درک معنا در زبان فارسی کاهش یابد. شبکه‌های منطق مارکوف، قاعده‌مندی منطق مرتبه اول را با عدم قطعیت احتمالات ترکیب می‌کنند. بنابراین، این شبکه‌ها چارچوب مناسبی را برای مدل‌سازی روابط معنایی پیچیده زبان گفتار و در نتیجه پیاده‌سازی واحد درک معنا فراهم می‌سازند. مطالب نظری مربوط به شبکه‌های منطق مارکوف از جمله اهداف، تعاریف پایه، روابط ریاضی و تعدادی از الگوریتم‌های یادگیری و استنتاج مربوط به این مدل‌ها توضیح داده شد. از جمله این مباحث می‌توان به الگوریتم استنتاج MC-SAT و الگوریتم یادگیری وزن پرسپترون مبتنی رأی‌گیری اشاره کرد. الگوریتم‌های یادگیری و استنتاج توسعه یافته برای شبکه‌های منطق مارکوف به موارد مذکور محدود نمی‌شوند؛ بلکه در این فصل، الگوریتم‌های یادگیری و استنتاج مهم استفاده شده در پیاده‌سازی‌های این پایان‌نامه به طور ساده تشریح شد.

فهرست مطالب

۱	مقدمه ای بر فهم زبان های طبیعی	۱
۱	۱.۱ زبان طبیعی	۱
۲	۲.۱ زبان گفتار	۲
۳	۳.۱ پردازش زبان طبیعی	۳
۷	۲ شبکه های منطق مارکوف	۷
۷	۱.۲ مقدمه	۷
۹	۲.۲ شبکه های مارکوف	۹
۱۱	۳.۲ منطق مرتبه اول	۱۱
۱۶	۴.۲ محاسبه احتمالات شرطی	۱۶
۲۰	۵.۲ یادگیری وزن به صورت افتراقی	۲۰
۲۱	۶.۲ روش پرسپترون مبتنی بر رأی گیری	۲۱
۲۲	۷.۲ جمع بندی	۲۲
۲۴	۳ کاربرد شبکه منطق مارکوف در پردازش زبان های طبیعی	۲۴
۲۴	۱.۳ دادگان درخت نحوی	۲۴
۲۶	۲.۳ کاربرد شبکه مارکوف در پردازش زبان طبیعی	۲۶

۳۳	آزمایشها و نتایج	۴
۳۳	تجزیه و استخراج قواعد از دادگان	۱.۴
۳۴	قالب مجموعه قوانین ساخته شده	۱.۱.۴
۳۸	چگونگی استخراج قواعد از مجموعه دادگان	۲.۱.۴
۳۹	وزن دهی به شبکه و پیاده سازی	۲.۴
۳۹	یادگیری وزن ها	۱.۲.۴
۴۰	اجرای الگوریتم و نتایج آن	۲.۲.۴
۴۱	جمع بندی	۳.۴
۴۱	آینده پژوهش	۴.۴
۴۴	واژه نامه فارسی به انگلیسی	۵

لیست تصاویر

۳	شمای کلی یک سامانه محاوره ای مبتنی بر گفتار	۱.۱
۶	شمای کلی پردازش زبان طبیعی	۲.۱
۱۴	مثالی از فرمول‌ها در منطق مارکوف	۱.۲
۱۵	شمای یک منطق مارکوف ساده	۲.۲
۱۸	شبه کد الگوریتم ساخت شبکه برای استنتاج	۳.۲
۱۹	یک نمونه شبکه پایه ساخته شده توسط الگوریتم ساخت شبکه	۴.۲
۲۰	شبه کد مربوط به الگوریتم MC-SAT	۵.۲
۲۵	مراحل کلی آموزش و آزمون سامانه	۱.۳
۳۰	شبه کد مربوط به الگوریتم MAXWalkSat	۲.۳
۳۸	نمونه ای از جملات دادگان	۱.۴
۳۹	نمونه ای از قواعد به دست آمده از روی دادگان	۲.۴

لیست جداول

- ۱.۳ آمار حضور گروه‌های نحوی در دادگان ۲۶
- ۲.۳ آمار حضور برچسب‌های نحوی کلمه در دادگان ۲۷

فصل ۱

مقدمه ای بر فهم زبان های طبیعی

در این بخش به توضیح مختصری از زبان های طبیعی و گفتاری می پردازیم و علت نیاز به فهم زبان های طبیعی را شرح می دهیم. در انتها، پردازش زبان های طبیعی و رابطه آن با زبان طبیعی را مورد بررسی قرار خواهیم داد.

۱.۱ زبان طبیعی

زبان های طبیعی به هر کدام از زبان هایی گفته می شود که توسط بشر در دنیا صحبت می شود و خودبه خود پدید آمده و تکامل یافته است. معمولاً این عنوان را در مقابل زبان های ساختگی مثل منطق صوری به کار می برند. هدف منطق از دیدگاه بسیاری از متفکران بازسازی ساختار زبان طبیعی است.

زبان معیار گونه ای نوشتاری یا گفتاری است که، افراد دارای تحصیلات بالای جامعه از آن استفاده می کنند و به عنوان الگوی زبان غالب، در جامعه پذیرفته شده است. در رسانه ها (مطبوعات، صدا و سیما)، سخنرانی ها و کتاب های درسی مدارس یا در گفتگوها و مصاحبه های رسمی از گونه معیار استفاده می شود. از آنجا که گونه معیار دارای اعتبار اجتماعی بالایی است، سایر گروه های اجتماعی نیز سعی در تقلید

و پیروی از آن دارند.

۲.۱ زبان گفتار

تعریف هایی که تا کنون از زبان معیار صورت گرفته بیشتر ناظر به زبان نوشتار بوده که زبان رسمی و متداول در سازمان هایی نظیر آموزش و پرورش و مطبوعات بوده است. اما این دیدگاه نه تنها بدان معنی نیست که زبان گفتاری فاقد اهمیت است بلکه زبان گفتار از اهمیت بیشتری برخوردار است. زبان گفتار برای انسان ها ساده ترین راه ارتباط است (چه ارتباط با سایر انسان ها و چه ارتباط با ماشین) و این مسئله باعث شده است که در کنار همه فناوری های موجود، قابلیت ارتباط ماشین و انسان از طریق زبان گفتار همواره جزء اهداف محققان باشد. برقراری این ارتباط نیازمند سامانه هایی برای درک معنای گفتار مبادله شده است.

با وجود اینکه عبارت «درک زبان گفتار»^۱ با هدف محاوره انسان و ماشین ابداع شده است متصور شدن کاربردهای دیگری از درک معنا در دیگر سامانه های پردازش زبان طبیعی مانند سامانه های پاسخگویی به پرسش ها^۲، جستجوی صدا^۳، استخراج موجودیت های دارای نام^۴ و دسته بندی موضوعی گفتار^۵ دشوار نیست.

^۱Spoken Language Understanding

^۲Question Answering Systems

^۳Voice Search

^۴Named Entity Recognition

^۵Speech Topic Classification



شکل ۱.۱: شمای کلی یک سامانه محاوره ای مبتنی بر گفتار

۳.۱ پردازش زبان طبیعی

هدف نهائی پردازش زبانهای طبیعی رسیدن به سطح توانائی های انسان در فهم و استفاده از زبانهای انسانی و سپس عبور از آن توانائی ها و رسیدن به توان پردازش سریع، دقیق و جامع این زبانهاست. در چند دهه اخیر ابزارها و روشهای مختلفی برای این منظور ایجاد شده اند که این روشها در بعضی کاربردهای زبان طبیعی منجر به موفقیت های قابل توجهی شده اند.

با این وجود پیشرفت در بعضی از زمینه های زبان طبیعی (مثلا ترجمه، فهم و اصلاح غلط ها) به علت پیچیدگی بالا، محدود بوده است. روشهای جدیدی که در چند سال اخیر ابداع شده اند از قبل نمایش زبان با بردارهای تعبیه شده و استفاده از شبکه های عصبی عمیق منجر به پیشرفت های بسیار خوبی شده است. هدف اصلی در پردازش زبان طبیعی، ایجاد تئوریهای محاسباتی از زبان، با استفاده از الگوریتمها و ساختارهای داده ای موجود در علوم رایانه ای است. بدیهی است که در

راستای تحقق این هدف، نیاز به دانشی وسیع از زبان است و علاوه بر محققان علوم رایانه‌ای، نیاز به دانش زبان شناسان نیز در این حوزه می‌باشد. کاربردهای پردازش زبان طبیعی به دو دسته کلی قابل تقسیم است: کاربردهای نوشتاری و کاربردهای گفتاری. از کاربردهای نوشتاری آن می‌توان به استخراج اطلاعاتی خاص از یک متن، ترجمه یک متن به زبانی دیگر و یا یافتن مستندات خاص در یک پایگاه داده نوشتاری (مثلاً یافتن کتاب‌های مرتبط به هم در یک کتابخانه) اشاره کرد. نمونه‌هایی از کاربردهای گفتاری پردازش زبان عبارتند از: سیستم‌های پرسش و پاسخ انسان با رایانه، سرویس‌های اتوماتیک ارتباط با مشتری از طریق تلفن و یا سیستم‌های کنترلی توسط صدا. در سال‌های اخیر این حوزه تحقیقاتی توجه دانشمندان را به خود جلب کرده است و تحقیقات قابل ملاحظه‌ای در این زمینه صورت گرفته است.

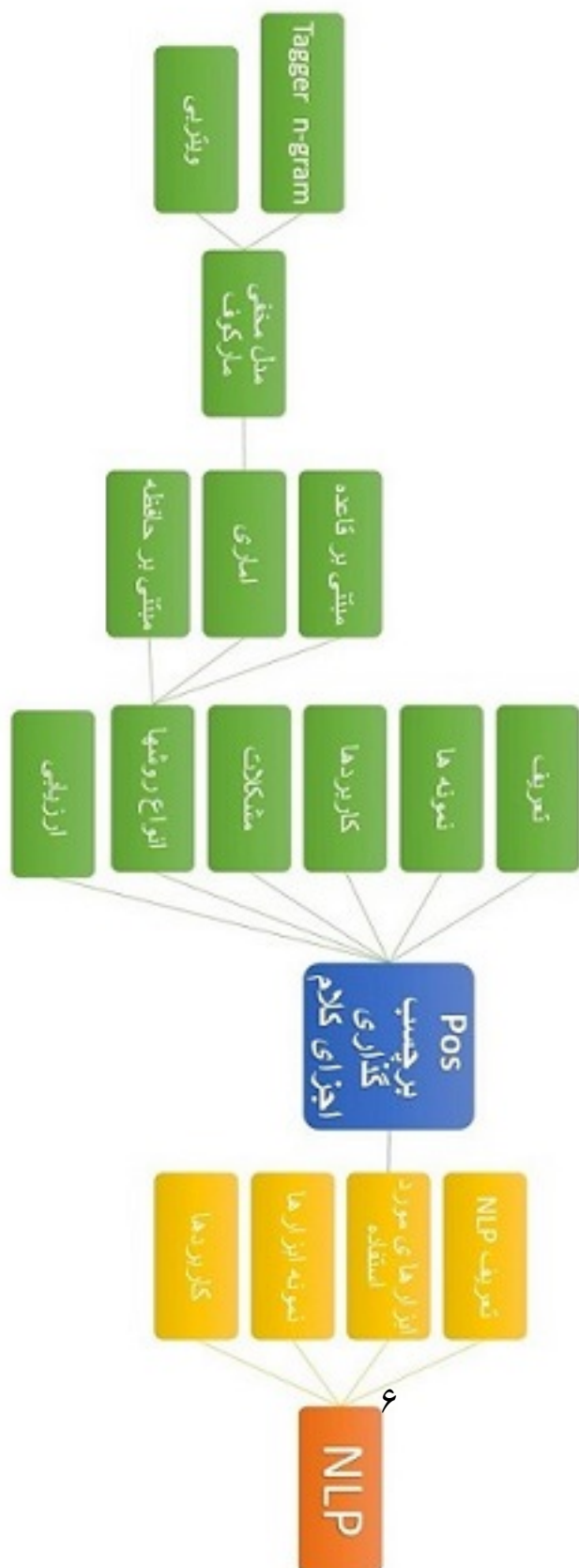
پردازش زبان طبیعی یکی از زیرشاخه‌های با اهمیت در حوزه گسترده هوش مصنوعی و دانش زبان‌شناسی است. تلاش عمده در این زمینه، ماشینی کردن فرایند درک و برداشت مفاهیم بیان شده توسط یک زبان طبیعی انسانی است. به تعریف دقیق‌تر پردازش زبان‌های طبیعی عبارت است از استفاده از رایانه به منظور پردازش زبان گفتاری و نوشتاری. پردازش زبان‌ها و مکالمات طبیعی یکی از اموری است که با ورود فناوری رایانه‌ای به زندگی بشر مورد توجه بسیاری از دانشمندان قرار گرفته است. حتی اندیشه‌ای که تورینگ از ماشین هوشمند خود و تعریفی که او از هوش مصنوعی داشت، در مرحله اول مربوط به پردازش زبان طبیعی می‌شد.

به تعریف دقیق‌تر پردازش زبان‌های طبیعی عبارت است از استفاده از رایانه برای پردازش زبان گفتاری و نوشتاری. با استفاده از آن می‌توان به ترجمه زبان‌ها پرداخت، از صفحات وب و بانک‌های اطلاعاتی نوشتاری جهت پاسخ دادن به پرسش‌ها استفاده کرد، یا با دستگاه‌ها (مثلاً برای مشورت گرفتن) به گفتگو پرداخت.

پردازش زبان‌های طبیعی زیرشاخه‌ای از هوش مصنوعی است که با توسعه و استفاده از مدل‌های رایانشی برای پردازش زبان سروکار دارد. در این زمینه، دو حیطه

اصلی پژوهش وجود دارد: ادراک، که با فرایندهایی سر و کار دارد که اطلاعات را از زبان استخراج می‌کنند (مانند درک زبان طبیعی، بازیابی اطلاعات) و تولید، که با فرایندهایی سر و کار دارد که با استفاده از زبان به انتقال اطلاعات می‌پردازند. معمولاً کارهای مرتبط با گفتار را تحت عناوین جداگانه تشخیص گفتار و تولید گفتار قرار می‌دهند.

با اینکه مجموعه گسترده‌ای از روش‌ها در پردازش زبان طبیعی بکار می‌روند، تکنیک‌های بکار رفته را می‌توان به سه دسته کلی تقسیم نمود: روش‌های آماری، روش‌های ساختاری/مبتنی بر الگو و روش‌های مبتنی بر استنتاج. باید توجه داشت که این راهکارها لزوماً از هم جدا نیستند. در واقع، جامع‌ترین مدل‌ها از ترکیب هر سه این روش‌ها استفاده می‌کنند. تفاوت این راهکارها در نوع عملیات پردازشی است که قادر به انجام آن هستند و میزان قواعدی که در مقابل آموزش/یادگیری خودکار از روی داده‌های زبانی نیاز دارند. در زیر، میتوان شمای کلی پردازش زبان طبیعی و شبکه را مشاهده کرد.



شکل ۲.۱: شمای کلی پردازش زبان طبیعی

فصل ۲

شبکه های منطق مارکوف

در این فصل به توضیح شبکه منطقی مارکوف، وزن دهی آن با روش های مختلف، کاربرد های آن و محاسبات احتمالات شرطی می پردازیم.

۱.۲ مقدمه

شبکه ی منطقی مارکوف یا (MLN) ، یکی از الگوریتم های مورد استفاده در هوش مصنوعی است که در سال ۲۰۰۶ به عنوان پایان نامه دکترای یکی از دانشجویان دانشگاه واشنگتن معرفی گردید. اپلیکیشن های یادگیری ماشین های هوش مصنوعی مدرن با درجه های پیچیدگی و عدم قطعیتشان دسته بندی می شوند. پیچیدگی توسط منطق مرتبه اول و عدم قطعیت توسط مدل های گرافیکی احتمالاتی به خوبی کنترل می شود. آنچه کمبودش احساس می شد ترکیب این دو بود.

شبکه منطق مارکوف این کمبود را با اضافه نمودن وزن به فرمول ها یا بند های منطقی و برخورد با آنها به عنوان قالب هایی برای ویژگی های فیلدهای تصادفی مارکوف برطرف کرد. این شبکه منطقی، یک منطق احتمالاتی است که ایده های شبکه مارکوف و منطق مرتبه اول را به کار می گیرد و استنتاج نا مطمئنی را ارائه می کند.

شبکه‌های منطقی مارکوف، منطق مرتبه اول را تعمیم می‌دهند؛ به این معنا که، در یک محدوده، تمام جملات غیر قابل ارضا احتمال صفر می‌گیرند و تمام گزاره‌ها (جملاتی که ارضا می‌کنند) احتمال یک می‌گیرند.

شبکه منطقی مارکوف به طور خلاصه کلکسیون از فرمول‌هایی از منطق مرتبه اول است که به هر یک از فرمول‌های یک عددی حقیقی تخصیص داده شده است (وزن). راس‌های گراف شبکه فرمول‌های اتمی هستند و یال‌ها رابط‌های منطقی استفاده شده در ساخت فرمول می‌باشند. هر فرمول به‌عنوان یک کلیک در نظر گرفته می‌شود و پوشش مارکوف^۱ مجموعه‌ای از فرمول‌هاست که درون آن یک اتم داده شده ظاهر می‌شود. یک تابع پتانسیل به هر فرمول اختصاص داده می‌شود و هنگامی که فرمول درست است مقدار ۱ می‌گیرد و در غیر اینصورت صفر. تابع پتانسیل ترکیبی از وزن برای ایجاد اندازه‌گیری گیبس^۲ و تابع تجزیه^۳ برای ایجاد گراف مارکوف می‌باشد. نکته‌ی مهم دیگری که در شبکه‌های منطقی مارکوف وجود دارد و در مقاله اصلی ارائه شده برای این روش ذکر شده این است که فرمول‌های اتمی ارزش درستی ندارند مگر آنکه آن‌ها پایه‌گذاری شده^۴ باشند و تفسیر^۵ ارایه دهند؛ یعنی تا زمانی که اتم‌های پایه‌ای با صدها تفسیر باشند، ارزش درستی^۶ دارند. بنابراین شبکه منطقی مارکوف تنها با توجه به پایه و تفسیر ویژه^۷ تبدیل به شبکه مارکوف می‌شود، شبکه مارکوف حاصل از این روش، شبکه مارکوف پایه^۸ نامیده می‌شود. راس‌های گراف شبکه مارکوف پایه، اتم‌های پایه هستند.

^۱ Markov blanket

^۲ Gibbs measure

^۳ partition function

^۴ grounded

^۵ interpretation

^۶ Truth Value

^۷ specific grounding and interpretation

^۸ ground Markov network

بنابراین اندازه شبکه مارکوف حاصل به شدت (بصورت نمایی) بستگی به تعداد ثابت ها در ناحیه بحث^۹ دارد. تکنیک های استنتاج این روش شامل تست متغیر کمکی^{۱۰} MCMC و قابلیت ارضا شدن^{۱۱} هستند. یادگیری شامل پرسپترون رای گیری شده^{۱۲}، تکنیک های مرتبه دوم، برنامه نویسی منطقی استقرایی^{۱۳} و شبه احتمال^{۱۴} و یا برنامه های منطقی استدلالی، تولید گزاره^{۱۵} و یادگیری انتقالی^{۱۶} می باشد. کاربردهای این روش شامل استخراج اطلاعات یا دانش و یکپارچه سازی، پردازش زبان طبیعی، پیاده سازی کردن ربات ها، شبکه های اجتماعی، محاسبات زیستی^{۱۷} و غیره است. در واقع مجموع منطق مرتبه اول^{۱۸} و شبکه های مارکوف منطق شبکه مارکوف را نتیجه می دهد [۲].

۲.۲ شبکه های مارکوف

برای معرفی شبکه منطقی مارکوف ابتدا شبکه مارکوف را تعریف می کنیم: شبکه مارکوف یک مدل گرافیکی احتمالاتی غیر جهت دارد است که توزیع احتمال مشترک را مشخص می کند. در یک شبکه مارکوف هر راس نشان دهنده یک متغیر تصادفی و هر یال نشان دهنده ارتباط وابستگی میان دو راسی است که به هم وصل نموده

^۹domain of discourse

^{۱۰} auxiliary-variable MCMC

^{۱۱} satisfiability

^{۱۲} voted perceptrons

^{۱۳} inductive logic programming

^{۱۴} pseudo-likelihood

^{۱۵} predicate invention

^{۱۶} transfer learning

^{۱۷} Computational Biology

^{۱۸} First Order Logic

است. نود در شبکه مارکوف از نود دیگر با داشتن مجموعه همسایگی اش استقلال شرطی دارد. هر کلیک در گراف یک تابع پتانسیل مربوط به آن دارد که حالات مورد احتمال المان های کلیک را به اعداد حقیقی غیر منفی نگاشت می دهد. توزیع مشترک شبکه مارکوف بصورت زیر است:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(X) \quad (1.2)$$

که در آن X برداری از متغیرهای تصادفی نشان داده شده در شبکه مارکوف، x ، نشانه ای به این متغیرهای تصادفی k ، تابع پتانسیلی که حالات K امین کلیک را به اعداد حقیقی غیر منفی نگاشت می دهد و Z ، فرم ثابت نرمال شده ای است که اطمینان حاصل می کند از اینکه جمع احتمال همه مقادیر محتمل x برابر با ۱ می شود. به بیان بهتر:

$$Z = \sum_x \prod_k \phi_k(X) \quad (2.2)$$

شبکه های مارکوف معمولاً توسط مدل های لگاریتمی خطی نمایش داده می شوند به طوری که پتانسیل هر خوشه با مجموع وزن دار نمایی حالت خود جایگزین شده و رابطه زیر را نتیجه می دهد:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x)\right) \quad (3.2)$$

در این نمایش $f_j(x)$ ویژگی متناظر با هر حالت محتمل از کلیک و w_j وزن مربوط به ویژگی است که معمولاً مقدار تابع پتانسیل است. در شبکه مارکوف هر استنتاج کامل-P^{۱۹} است. بنابراین الگوریتم های تخمین مانند زنجیره ی مارکوف مونت

^{۱۹}P-complete

کارلو ۲۰ و نمونه گیس ۲۱ مورد استفاده برای استدلال تقریب خوب استفاده می شوند. به این صورت که عمل نمونه گیری را انجام می دهد و مقادیر احتمالات کناری نیز با استفاده از همین نمونه ها محاسبه می شود. پوشش مارکوفی یک گره مجموعه کمینه گره هایی است که استقلال آن گره از بقیه شبکه را نتیجه می دهد. در یک شبکه مارکوف، پوشش مارکوفی همان گره های همسایه هستند.

روش رایج دیگر برای استنتاج در شبکه های مارکوف، روش انتشار باور ۲۲ می باشد. این روش از ساخت یک گراف دوبخشی از گره ها و پتانسیل ها تشکیل شده است. پیام ها که در این روش بازنمود تقریب های فعلی از احتمال های کناری گره ها هستند از گره های متغیر به گره های پتانسیل مربوط ارسال می شوند و برعکس. در یک شبکه مارکوف در حالت کلی تضمینی برای همگرایی این عملیات پیام رسانی و یا محاسبه دقیق احتمال های کناری وجود ندارد؛ اما نشان داده شده است که در بسیاری از مسائل دنیای واقعی عملکرد بسیار مناسبی دارد [۳].

۳.۲ منطق مرتبه اول

منطق مرتبه اول سیستمی از نمایش دانش در سیستم های استدلالی است. این سیستم نسبت به منطق گزاره ای خیلی رساتر است زیرا علاوه بر سیستم های کمی، امکان گزاره ای را هم ارائه می دهد. گزاره های نحوی منطق مرتبه اول از متغیرهای تعریف شده توسط کاربر، متن ها، توابع و گزاره ها تشکیل شده است.

محدوده متغیرها در طول شیء های دامنه است و معمولاً با حروف کوچک نمایش داده می شوند. یک ثابت نشان دهنده یک شیء خاص مانند یک شخص یا یک سازمان است. توابع برای نگاشت شیء های خاص یا چندین شیء به سایر اشیاء

^{۲۰}Markov Chain Monte Carlo

^{۲۱}Gibbs Sampling

^{۲۲}Belief propagation

استفاده می شوند. گزاره ها تعیین کننده ویژگی های خاص یا ارتباطات میان دو شیء هستند. حال که توضیح کوتاهی درباره منطق مرتبه اول و شبکه های مارکوف آورده شد می توانیم در باره شبکه های منطقی مارکوف یا MLN ها صحبت کنیم. شبکه منطق مارکوف در واقع روشی است که توسط ریچاردسون^{۲۳} و دومینگوس^{۲۴} از دانشگاه واشنگتون ارائه شده است [۶]. این دو، روشی را ارائه کردند که منطق مرتبه اول و مدل های احتمالی گرافیکی را ترکیب کرده و بصورت یک نمایش واحد در آورده است.

یک MLN یک پایگاه دانش مرتبه اول است که به هر بند آن یک وزن نسبت داده شده است که همراه با مجموعه ای از ثابت ها، شیء ها را در دامنه نمایش می دهند. این شبکه منطقی یک شبکه مارکوف را نشان می دهد که برای هر بند مرتبه اول محتمل در پایگاه داده همراه با وزن مربوط به آن دارای یک ویژگی است. استنتاج در MLN توسط MCMC روی زیرمجموعه کمینه شبکه پایه ای که برای پاسخگویی به درخواست^{۲۵} مورد نیاز است انجام می شود. وزن ها از پایگاه داده های رابطه ای با بهینه سازی مرتب (تکرارگونه) یک معیار شبه احتمالاتی آموزش داده می شوند. این قسمت نیز بصورت دلخواه است یعنی این گزینه نیز وجود دارد که بند های اضافی دیگر با استفاده از تکنیک های برنامه نویسی منطقی استنتاجی یاد گرفته شوند.

یک پایگاه دانش^{۲۶} مرتبه اول را می توان بصورت مجموعه ای از محدودیت های مشکل^{۲۷} روی مجموعه ای از جهان های محتمل دید: اگر جهانی از حتی یک فرمول تخطی کرد، احتمال آن صفر می شود. ایده اصلی منطق مارکوف این است که این محدودیت های سخت را نرم کند تا عدم قطعیت را کنترل کند: زمانی که جهانی از یک فرمول در پایگاه دانش تخطی کرد احتمال آن کمتر می شود ولی غیر

^{۲۳}Matthew Richardson

^{۲۴}Pedro Domingos

^{۲۵}Query

^{۲۶}KB or Knowledge Base

^{۲۷}hard constraint

محتمل نمی شود. به هر ویژگی یک وزن نسبت داده می شود. اگر گزاره های پایه حداقل در یک پایه از یک بند با هم ظاهر شوند این نشان می دهد که رابطه وابستگی میان گزاره ها وجود دارد در نتیجه شبکه مارکوف برای لینک دادن این نودها از یک یال استفاده می کند در نتیجه هر فرمول یک کلیک در شبکه مارکوف پایه ایجاد می کند. به محض اینکه ساخته شود می توان مانند شبکه های مارکوف از شبکه مارکوف پایه نیز استنتاج کرد.

هر جهان هرچه از تعداد فرمول کمتری تخطی کند احتمالش بیشتر می شود. به هر فرمول یک وزن اختصاص داده شده که در شکل ۱.۲ نمونه آن آورده شده است. این وزن ها نشان دهنده این هستند که قدرت این محدودیت چگونه است: هرچه وزن بالاتر باشد تفاوت میان لاگ میان جهانی که بند را ارضا می کند و جهانی که ارضا نمی کند بیشتر است، فرض می کنیم بقیه موارد مشابه هستند. تعریف دیگری از MLN به این صورت است:

یک شبکه منطقی مارکوف L مجموعه ای از زوج های مرتب (F_i, w_i) است که F_i فرمول یا بندی در منطق مرتبه اول بوده و w_i یک عدد حقیقی است. این مجموعه از زوج های مرتب همراه با مجموعه متناهی از ثابت هایی به نام $C = \{c_1, c_2, \dots, c_C\}$ ، شبکه مارکوف $M_{L,C}$ را تعریف میکند:

۱- $M_{L,C}$ حاوی یک نود باینری برای هر پایه ممکن از هر اتمی که در L ظاهر می شود می باشد. مقدار نود اگر اتم پایه، درست باشد ۱ است در غیر این صورت ۰.

۲- $M_{L,C}$ حاوی یک ویژگی برای هر پایه ممکن از هر بند F_i در L است. ارزش این ویژگی، اگر درست باشد ۱ است در غیر این صورت ۰ است. وزن ویژگی w_i مربوط به F_i در L است. بنابراین یالی میان دو نود $M_{L,C}$ وجود دارد اگر و تنها اگر اتم های پایه مربوطه حداقل در یک پایه از یک فرمول در L با هم ظاهر شوند. برای مثال در MLN زیر:

برای عبارت "اگر x و y دوست باشند و x سیگار بکشد آنگاه y هم سیگار می

وزن	عبارت منطق مرتبه اول	شناسه فرمول
۱.۵	$\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$	ϕ_1
۱.۱	$\forall x \text{ Smokes}(x) \wedge \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)$	ϕ_2

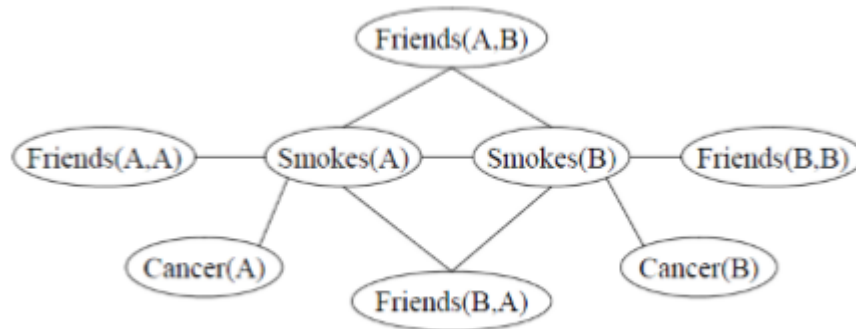
شکل ۱.۲: مثالی از فرمول‌ها در منطق مارکوف

کشد یعنی دوستان عادات سیگار کشیدن یکسانی دارند” برای دو ثابت ، Anna و Bob به کار گرفته شده یا به طور خلاصه A و B که در شبکه مارکوف شکل زیر را به دست می دهد.

توجه کنید که با اینکه دو بند فوق با توجه به اظهارات منطقی جهانی غلط هستند، به عنوان ویژگی های وزن دار یک MLN قواعد آماری معتبری به دست می آورند و در واقع نشان دهند یک مدل شبکه اجتماعی استاندارد می باشند. شکل ۲.۲ شبکه مارکوف پایه ای به دست آمده توسط MLN است که شامل بندهایی که در بالا معرفی کردیم می شود. به عنوان مثال ساده ای از تحلیل شبکه های اجتماعی از شبکه اجتماعی ساده دوستان، سیگار و سرطان استفاده شده است. این شبکه سعی می کند ارتباطات دوستی میان افراد، عادات سیگار کشیدن و احتمال سرطان را مدل کند [۸].

شبکه منطق مارکوف را می توان به عنوان قالبی برای ساخت شبکه های مارکوف دید. توزیع احتمال جهان های محتمل x مشخص شده توسط شبکه منطق مارکوف بصورت زیر است:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x)\right) \quad (۴.۲)$$



شکل ۲.۲: شمای یک منطق مارکوف ساده

زمانی که عملیات جمع روی همه بند ها به پایان برسد، Z ثابت نرمال شده است ، w_i وزن مین i بند است، اگر امین i بند true باشد $F_i = 1$ و در غیر اینصورت $F_i = 0$ خواهد بود. میتوان معادله را متعاقبا بصورت زیر نوشت:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^F w_i n_i(x)\right) \quad (5.2)$$

F تعداد بندهای MLN است و $n_i(x)$ تعداد پایه های صحیح F_i در X می باشد. هرچه وزن بند بیشتر شود، MLN به طور فزاینده یک پایگاه دانش کاملا منطقی را شبیه سازی می کند و با متناهی کردن همه وزنه های نامتناهی تبدیل به یک پایگاه دانش منطقی می شود. زمانی که وزنها مثبت و متناهی باشند و همه فرمول ها بصورت همزمان قابل ارضا باشند، راهکارهای ارضا کننده حالات توزیع نمایش داده شده توسط شبکه منطق مارکوف خواهند بود.

منطق مارکوف تناقضات میان فرمول ها یا بندها را ممکن میکند که با وزن کردن هر دو طرف به سادگی برطرف می شود. این کار باعث می شود به خوبی با چندین

پایگاه دانش ادغام شود. منطق مارکوف یک رویکرد طبیعی و قدرتمند برای مسائل ادغام دانش و داده در نمایش های مختلف که به خوبی هم تراز نمی شوند ارائه می دهد. به راحتی می توان دید که همه مدل های گسسته احتمال از جمله شبکه های مارکوف و شبکه های بیزین را می توان با منطق مارکوف نمایش داد. بسیاری از مدل هایی که اغلب در هوش مصنوعی استفاده می شوند را می توان با MLN ها نمایش داد و آنها را با اضافه نمودن فرمول های مربوطه (بندهای مربوطه) ترکیب کرده و بسط داد. مهم تر اینکه منطق مارکوف ساخت مدل هایی که در آنها شیء ها مستقل نیستند و به طور یکسان توزیع شده اند را آسان می کند. زمان کار با منطق مارکوف تخمین برای نمایش منطقی در نظر گرفته می شود: ثابت های مختلف نشان دهنده آججت های مختلف (نام های منحصر به فرد) هستند، تنها شیء هایی که در دامین وجود دارند آنهایی هستند که با استفاده از سمبل های تابع و ثابت قابل نمایش اند و مقدار هر تابع برای هر تعداد استدلال^{۲۸} همواره یک ثابت شناخته شده است (توابع شناخته شده). این تخمین ها تضمین می کند که تعداد جهان های محتمل متناهی است و اینکه شبکه منطقی مارکوف توزیع احتمال به خوبی تعریف شده ای ارائه می دهد. این تخمین ها در بسیاری کاربردهای عملی کاملاً منطقی هستند و استفاده از MLN ها را ساده می سازند.

۴.۲ محاسبه احتمالات شرطی

شبکه های منطق مارکوف می توانند پرس و جو هایی به فرم زیر را پاسخ دهند: احتمال صادق بودن فرمول F_i در صورت صادق بودن فرمول F_2 چند است؟ اگر F_1 و F_2 دو فرمول بیان شده به زبان منطق مرتبه اول، C مجموعه ای متناهی از ثوابت و L یک شبکه منطق مارکوف باشد، آنگاه:

^{۲۸}Argument

$$P(F_1|F_2, L, C) = P(F_1|F_2, M_{L,C}) = \frac{P(F_1 \wedge F_2|M_{L,C})}{P(F_2|M_{L,C})} \quad (6.2)$$

$$= \frac{\sum_{x \in XF_1 \cap XF_2} P(X = x|M_{L,C})}{\sum_{x \in XF_2} P(X = x|M_{L,C})} \quad (7.2)$$

که در آن XF_i مجموعه دنیاهایی است که F_i در آن‌ها صادق می‌باشد، $M_{L,C}$ شبکه مارکوفی است که توسط L ، C تعریف می‌شود پرس و جویهای شرطی معمول در مدل‌های گرافیکی، حالات خاصی از رابطه بالا هستند که در آن فرمول‌ها به صورت ترکیب عطفی هستند. . تصمیم‌گیری در مورد ایجاب شدن فرمول F توسط پایگاه دانش با بررسی احتمال $P(F|L_{K,B})$ و $C_{K,F}$ انجام می‌شود؛ در این عبارت احتمالی، L شبکه منطق مارکوفی است که با انتساب وزن‌های بی‌نهایت به همه فرمول‌های درون KB به دست می‌آید. شبه کد الگوریتم ساخت شبکه در شکل ۲-۴ و یک نمونه از شبکه نتیجه، در شکل ۵-۲ یک نمونه شبکه پایه ساخته شده توسط الگوریتم ساخت شبکه، نشان داده می‌شود. در بدترین حالت، شبکه دارای گره خواهد بود که در آن a حداکثر برابر با بیشترین تعداد آرگومان مسندهای دامنه و در عمل بسیار کوچک‌تر است. پس از ایجاد شبکه، می‌توان از روش‌های استنتاج استاندارد شبکه‌های مارکوف مانند نمونه‌گیری گیبس استفاده کرد. با این وجود، یک مشکل مهم روش‌هایی مانند نمونه برداری گیبس برای استنتاج در شبکه‌های منطق مارکوف، عدم امکان مدیریت وابستگی‌های قطعی است. این نوع وابستگی‌ها، فضای دنیاهای ممکن را به ناحیه‌هایی تقسیم می‌کنند که به یک دیگر دسترسی ندارند و این مسئله یکی از اصلی‌ترین پیش‌نیازهای الگوریتم‌های $MCMC$ را نقض می‌کند.

الگوریتم $MS-SAT$ این مشکل را با ترکیب $MCMC$ با روش‌های آزمون صدق‌پذیری حل می‌کند. این الگوریتم، یک روش نمونه برداری مقطعی است که از ترکیب صدق‌پذیری با تبرید شبیه‌سازی برای نمونه برداری از هر مقطع استفاده می‌کند. این الگوریتم از روش‌های $MCMC$ عادی مانند نمونه برداری گیبس و تبرید

function ConstructNetwork(F_1, F_2, L, C)

inputs: F_1 , a set of ground predicates with unknown truth values (the “query”)
 F_2 , a set of ground predicates with known truth values (the “evidence”)
 L , a Markov logic network
 C , a set of constants

output: M , a ground Markov network

calls: $MB(q)$, the Markov blanket of q in $M_{L,C}$

$G \leftarrow F_1$

while $F_1 \neq \emptyset$

for all $q \in F_1$

if $q \notin F_2$

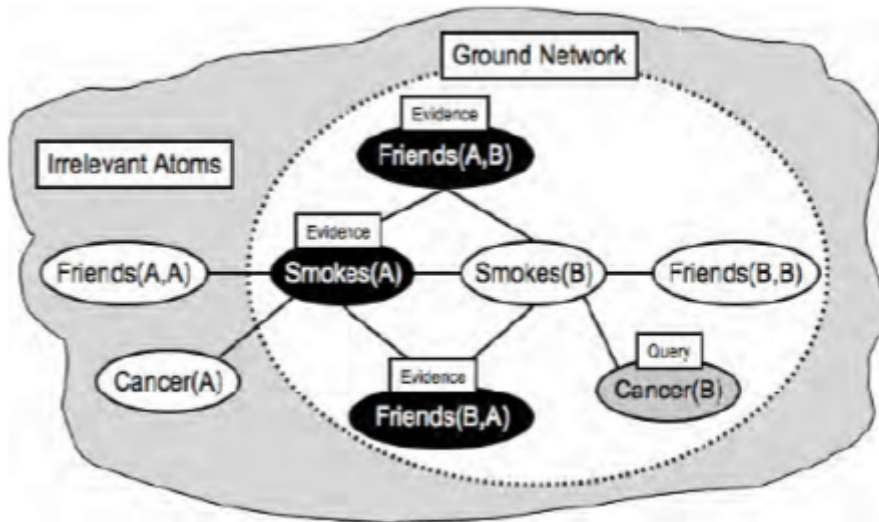
$F_1 \leftarrow F_1 \cup (MB(q) \setminus G)$

$G \leftarrow G \cup MB(q)$

$F_1 \leftarrow F_1 \setminus \{q\}$

return M , the ground Markov network composed of all nodes in G , all arcs between them in $M_{L,C}$, and the features and weights on the corresponding cliques.

شکل ۳.۲: شبه کد الگوریتم ساخت شبکه برای استنتاج



شکل ۴.۲: یک نمونه شبکه پایه ساخته شده توسط الگوریتم ساخت شبکه

شبیه سازی شده بسیار سریع تر عمل می کند و برای هر مدلی که بتوان در چارچوب منطق مارکوف بیان کرد قابل اعمال است. الگوریتم MC-SAT در هر مرحله از نمونه برداری، با استفاده از مجموعه تمام بندهای پایه ای که توسط حالت فعلی دنیا ارضا می شوند، زیرمجموعه ای مانند M تشکیل می دهد که باید توسط حالت بعدی دنیا که نمونه برداری می شود، ارضا گردد. یک بند پایه ارضاشده با احتمال به M اضافه می شود؛ که w وزن بند است. سپس، یک نمونه دیگر از مجموعه حالت هایی که M را ارضا می کنند، توسط یک توزیع یک نواخت به عنوان حالت بعدی انتخاب می شود. شبه کد مربوط به الگوریتم MC-SAT در شکل ۲-۶ آمده است.

```

function MC-SAT( $L, n$ )
  inputs:  $L$ , a set of weighted clauses  $\{(w_j, c_j)\}$ 
   $n$ , number of samples
  outputs:  $\{x^{(1)}, \dots, x^{(n)}\}$ , set of  $n$  samples
   $x^{(0)} \leftarrow \text{Satisfy}(\text{hard clauses in } L)$ 
  for  $i \leftarrow 1$  to  $n$ 
     $M \leftarrow \emptyset$ 
    for all  $(w_k, c_k) \in L$  satisfied by  $x^{(i-1)}$ 
      With probability  $1 - e^{-w_k}$  add  $c_k$  to  $M$ 
    Sample  $x^{(i)} \sim u_{\text{SAT}(M)}$ 

```

شکل ۵.۲: شبه کد مربوط به الگوریتم MC-SAT

۵.۲ یادگیری وزن به صورت افتراقی

علاوه بر طراحی پایگاه دانش، یک شبکه منطق مارکوف به وزن‌های مربوط به هر کدام از بندهای پایگاه دانش نیاز دارد. در بعضی کاربردها می‌توان این وزن‌ها را به صورت دستی به بندها نسبت داد؛ اما در مسئله مربوط به طراحی واحد درک معنا امکان تخمین دستی سختی همه وابستگی‌ها و بندها وجود ندارد. بنابراین باید برای مدلی برای یادگیری خودکار وزن هر بند آموزش داده شود. الگوریتم‌های از داده‌ها برای بهبود یا ساخت خودکار شبکه منطق مارکوف استفاده می‌کنند و معمولاً شبکه یادگیری مدل‌های بسیار بهتری را با هزینه کمتر نسبت به مدل‌های دستی تولید می‌حاصل کند.

یادگیری وزن در شبکه‌های منطق مارکوف به دو صورت مولد و افتراقی انجام می‌شود. در بعضی کاربردها مسندهای مربوط به شواهد و مسندهای مربوط به پرس‌وجو از پیش مشخص هستند و هدف نیز پیش‌بینی صحیح مسند پرس‌وجو با داشتن شواهد است. در سامانه درک معنا مسند مورد پرس‌وجو توسط کاربر یعنی

برچسب معنایی کلمات از پیش مشخص است. بنابراین استفاده از یک رویکرد افتراقی مؤثرتر خواهد بود؛ زیرا در یادگیری مولد سعی بر بیشینه‌سازی مقدار عبارت $P(q, e)$ است در حالی که در یادگیری افتراقی سعی بر بیشینه‌سازی مقدار عبارت $P(q|e)$ است که در آن q مسند مورد پرس و جو و e مجموعه مسند های شواهد است.

اگر مجموعه اتم‌های پایه را به دو دسته مجموعه شواهد X و مجموعه پرس و جو های Y افراز کنیم، درست‌نمایی شرطی Y در صورت داشتن X توسط رابطه زیر تعریف می‌شود:

$$P(Y = y|X = x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^F w_i n_i(x, y)\right) \quad (۸.۲)$$

که در آن Z_x ثابت نرمال سازی بر روی تمام دنیا‌های سازگار با x و $n_i(x, y)$ برابر با تعداد پایه سازی های صادق فرمول i در دادگان است. یادگیری افتراقی وزن‌ها با استفاده از بیشینه‌سازی درست‌نمایی لگاریتمی شرطی انجام می‌شود. از آن جا که هدف بیشتر مسائل بهینه سازی، کمینه سازی یک تابع است، در ادامه فصل، هدف، کمینه سازی منفی درست‌نمایی لگاریتمی شرطی خواهد بود.

۶.۲ روش پرسپترون مبتنی بر رأی‌گیری

الگوریتم پرسپترون مبتنی بر رأی‌گیری نوعی الگوریتم کاهش گرادیان است که ابتدا وزن همه بندها را برابر با صفر قرار می‌دهد. سپس داده‌های آموزشی را پیمایش می‌کند و وزن هر بند را با توجه درستی مقدار پیش‌بینی شده برای داده آموزشی به‌روزرسانی می‌کند. الگوریتم های کاهش گرادیان، در هر مرحله، وزن w را با استفاده از گرادیان g و نرخ یادگیری η توسط رابطه زیر روزرسانی می‌کنند:

$$w_{t+1} = w_t - \eta g \quad (9.2)$$

در يك شبکه منطق مارکوف، مشتق منفي درست‌نمايي لگاریتمی شرطی نسبت به يك وزن برابر است با تفاضل امید ریاضی تعداد پایه سازی های صحیح بند مربوط به آن وزن و تعداد واقعی آن ها با توجه به دادگان. رابطه زیر این مقدار را به صورت ریاضی نشان می دهد.

$$\frac{\partial(-\log P_w(Y = y|X = x))}{\partial w_i} = -n_i(x, y) + \sum_z P_w(Y = z, X = x)n_i(x, z) \quad (10.2)$$

$$= E_{w,y} [n_i(x, y)] - n_i(x, y) \quad (11.2)$$

ایده اساسی الگوریتم پرسپترون مبتنی بر یادگیری، تخمین عبارت امید ریاضی توسط شمارش $n_i(x, y)$ در حالت بیشینه احتمال پسین (محتمل ترین حالت دنیا با توجه به شواهد) است. در نهایت برای جلوگیری از بیش برآزش طبق رابطه زیر میانگین وزن های هر پیمایش به جای وزن های نهایی به عنوان خروجی الگوریتم، نتیجه داده می شود.

$$w_i = \frac{\sum_{t=1}^T w_{i,t}}{T} \quad (12.2)$$

۷.۲ جمع بندی

در این فصل، مطالب نظری مربوط به شبکه های منطق مارکوف از جمله اهداف، تعاریف پایه، روابط ریاضی و تعدادی از الگوریتم های یادگیری و استنتاج مربوط به این مدل ها توضیح داده شد. از جمله این مباحث می توان به الگوریتم استنتاج-MC SAT و الگوریتم یادگیری وزن پرسپترون مبتنی بر رأی گیری اشاره کرد. الگوریتم های

یادگیری و استنتاج توسعه یافته برای شبکه های منطق مارکوف به موارد مذکور محدود نمی شوند؛ بلکه در این فصل، الگوریتم های یادگیری و استنتاج مهم استفاده شده در پیاده سازی های این پایان نامه به طور ساده تشریح شد.

فصل ۳

کاربرد شبکه منطق مارکوف در پردازش زبان های طبیعی

در بخش قبل مفهوم و ویژگی های شبکه منطق مارکوف را بررسی کردیم. در این بخش، قبل از پرداختن به کاربرد آن در پردازش زبان های طبیعی، نیازمند بررسی دادگان استفاده شده در این مقاله هستیم. ابتدا ما به معرفی دادگان درخت نحوی مبتنی بر دستور زبان فارسی میپردازیم، سپس بعد از توضیح مختصری از این دادگان و ویژگی های آنها، به چگونگی کاربرد این شبکه خواهیم پرداخت.

۱.۳ دادگان درخت نحوی

دادگان درخت نحوی^۱ یک پیکره متنی است که در آن هر جمله به صورت نحوی تجزیه شده و به صورت درخت تجزیه نحوی نمایش داده می شود. دادگان درخت نحوی در بسیاری از کاربردهای پردازش زبان طبیعی و هم چنین در زبان شناسی پیکره ای برای مطالعه پدیده های نحوی زبان مورد استفاده قرار می گیرند. از جمله

^۱Tree Bank

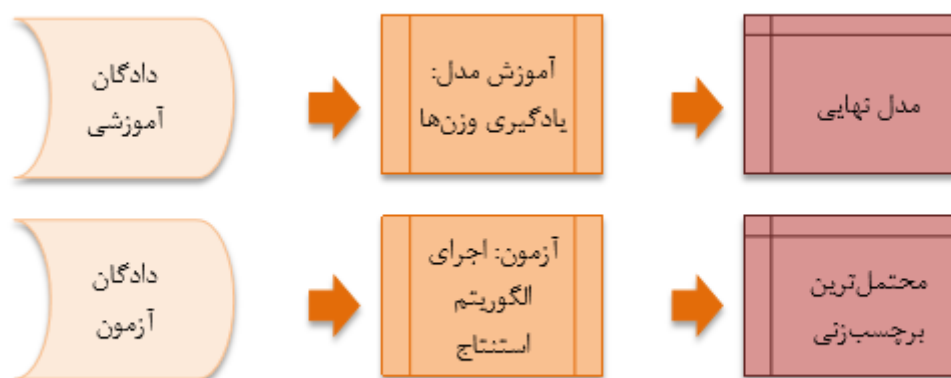
کاربردهای این گونه پیکره‌ها می‌توان به ترجمه خودکار متون، بازشناسی گفتار، استخراج اطلاعات، مقایسه کارایی مدل‌های تجزیه نحوی، گسترش نظریه نحوی متفاوت و محاسبه بسامد تکرار قوانین نحوی زبانی‌های اشاره کرد. سازکار کلی در شکل ۱.۳ قابل نمایش است.

در این مقاله ما نیز از دادگان درخت نحوی برای تجزیه و تحلیل قوانین مورد نیاز برای شبکه منطق مارکوف استفاده کرده ایم. این دادگان که دارای حجمی در حدود بیست و هشت هزار جمله است، به صورت خودکار و با استفاده از روش‌های قاعده‌مند پردازش زبان طبیعی توسط رایانه حاصل شده است.

دادگان درخت نحوی شامل ۳۹۲۱۴۹ کلمه و ۲۷۷۱۰ جمله و با میانگین طول جملات ۱۴.۱۵ است که از متون مختلف فرهنگی، سیاسی، هنری، اقتصادی، مذهبی و غیره از وب گردآوری شده است.

در زیر، جدولی از اختصارهای استفاده شده در این مقاله با توضیح مختصری از آنها آورده شده است.

همچنین جدول برچسب‌های نحوی کلمات در دادگان نیز به صورت ذیل است:



شکل ۱.۳: مراحل کلی آموزش و آزمون سامانه

جدول ۱.۳: آمار حضور گروه‌های نحوی در دادگان

تعداد تکرار	درصد حضور	توضیح	اختصار
۲۱۰۳۰۰	۵۵.۴۹	گروه اسمی	NP
۶۰۰۹۰	۱۵.۸۵	گروه فعلی	VP
۵۱۶۲۲	۱۳.۶۲	گروه حرف اضافه‌ای	PP
۳۳۳۲۷	۷.۸۹	گروه صفتی	ADJP
۷۹۶۰	۱.۹۷	بند اسم	SBAR-NCL
۷۵۰۲	۲.۱	گروه قیدی	ADVP
۵۰۰۷	۱.۳۲	بند متممی فعل	SBAR-VCL
۲۷۷۱	۰.۰۹	بند افزوده فعل	SBAR-AJUCL
۳۵۱	۰.۷۳	بند صفت	SBAR-ACL

۲.۳ کاربرد شبکه مارکوف در پردازش زبان طبیعی

از شبکه منطقی مارکوف برای استخراج اطلاعات استفاده می‌شود. به عبارت دقیق‌تر برای پیدا کردن موجودیت‌های نام‌هویک جمله برای مثال تگ کردن نام‌های

جدول ۲.۳: آمار حضور برچسب‌های نحوی کلمه در دادگان

تعداد تکرار	درصد حضور	توضیح	اختصار
۳۱۸۰۲	۷.۵۵	صفت	ADJ
۷۴۷۴	۱.۷۷	قید	ADV
۱۱۴	۰.۰۲	نقش نمای ندا	ADR
۹۷۲۸	۲.۳۱	نقش نمای همپایگی	CONJ
۷۷۴	۰.۱۸	شاخص	IDEN
۱۷۱۵۵۰	۴۰.۷۵	اسم	NN
۴۷۸	۱.۱۳	جزء دستوری	PART
۵۱۹	۰.۱۲	صفت شمارشی پسین	POSNUM
۱۳۶۴۱	۳.۲۴	حرف اضافه پسین	POSTP
۱۶۶۲۰	۳.۹۴	ضمیر	PR
۹۵۱۴	۲.۲۶	پیش توصیفگر	PREM
۵۲۸۱۱	۱۲.۳۱	صفت شمارشی پیشین	PRENUM
۵۱۶۲۲	۱۲.۲۶	حرف اضافه پیشین	PREP
۲۵۸	۰.۰۶	شبه جمله	PSUS
۳۵۷۶۱	۸.۴۹	علامت نگارشی	PUNC
۱۱۹۹۳	۲.۸۴	نقش نمای وابستگی ^{۲۷}	SUBR
۵۳۷۶۹	۱۲.۷۷	فعل	V

افراد به عنوان اسم شخص و تگ کردن نام های Iran Iraq و US و ... به عنوان موجودیت های سیاسی. برای این کار می توان از روش MLN که یک مدل یادگیری رابطه ای آماری قدرتمند است و نمایش قوی ارائه می کند استفاده نمود. با استفاده از این روش برای تگ کردن اسامی می توان وابستگی های پیچیده در استخراج نام را مدل کرد. در این مثال مربوط به مدل تگ کردن موجودیت می توان برخی فرمول های منطق مرتبه اول را برای به دست آوردن مدل مورد نظر تعریف کرد:

$$\text{SameToken}(x, y) \wedge \text{Label}(x, l) \Rightarrow \text{Label}(y, l)$$

در این فرمول x و y متغیرهایی از جنس token هستند و l متغیری برای نوع برچسب است. برای این فرمول می توان وزن هم تعریف کرد و هم می تواند از داده های آموزشی وزنها را یاد بگیرد. استفاده از این فرمول برای توکن های A و B و برچسب های p برای person و o برای organization شبکه مارکوف پایه زیر را می دهد:

استنتاج در MLN نیازمند منطق و استنتاج در احتمالات و در وابستگی های قطعی است. الگوریتم های MCMC از قدیم برای استنتاج در مدل های احتمالاتی و الگوریتم های قابل ارضا برای سیستم های کاملاً منطقی استفاده می شدند. از آنجایی که MLN هم شامل وابستگی های قطعی و هم احتمالاتی است هیچ یک از این الگوریتم ها نتایج خوبی نمی دهند. در این کاربرد از الگوریتم MC-SAT استفاده شده تا ارزش گزاره های پرسشی را تعیین کند. الگوریتم MC-SAT الگوریتمی است که تکنیک های MCMC و الگوریتم های قابل ارضا را ادغام میکند بنابراین در استنتاج MLN خوب عمل می کند. یکی از کاربردهای اساسی استنتاج در منطق مارکوف، یافتن محتمل ترین حالت دنیا با استفاده از مجموعه ای از شواهد داده شده می باشد. به این نوع استنتاج در ادبیات شبکه های مارکوف از استنتاج پیشینه احتمال پسین گفته می شود. این روش در منطق مارکوف توسط رابطه زیر بیان می شود:

$$\arg \max_y P(y|x) = \arg \max_y \frac{1}{Z_x} \exp\left(\sum_{i=1}^F w_i n_i(x, y)\right) = \arg \max_y \sum_i w_i n_i(x, y) \quad (1.3)$$

ثابت نرمال‌سازی به صورت Z_x نوشته شده است تا بیان‌گر این مسئله باشد که عمل نرمال‌سازی فقط بر روی دنیاهای سازگار با شواهد x ، انجام می‌شود. در تساوی دوم Z_x و عبارت تابع نمایی به علت ثابت بودن و عدم ایجاد تأثیر بر روی عبارت بیشینه‌سازی حذف شده است. بنابراین مسئله یافتن محتمل‌ترین حالت دنیا در منطق مارکوف تقلیل می‌یابد به یافتن انتساب ارزش صدقی که مجموع وزن‌های بندهای صادق را بیشینه کند.

این مسئله در حالت کلی سخت است اما راه‌حل‌های دقیق و تقریبی مفیدی برای آن وجود دارند. رایج‌ترین الگوریتم مورد استفاده برای حل این مسئله، الگوریتم MaxWalkSAT است که نسخه وزن‌دار نسخه وزن‌دار الگوریتم جست‌وجوی محلی WalkSAT است که مسائل سخت با صدها هزار متغیر را در چند دقیقه حل می‌کند. الگوریتم MaxWalkSAT این جست‌وجوی تصادفی را توسط انتخاب مکرر و تصادفی یک بند ارضاننده، و تغییر ارزش صدق یکی از اتم‌های داخل آن انجام می‌دهد. با یک احتمال معین، اتم به صورت تصادفی انتخاب می‌شود؛ و در غیر این صورت اتم طوری انتخاب می‌شود که مجموع وزن‌های بندهای صادق با تغییر ارزش صدق بیشینه گردد.

این ترکیب گام‌های تصادفی و حریصانه اجازه می‌دهد تا MaxWalkSAT از گیر افتادن در بهینه محلی در حین جست‌وجو جلوگیری کند. شبه‌کد مربوط به الگوریتم MaxWalkSAT در شکل زیر مشاهده می‌شود. تابع DeltaCost تغییرات مجموع وزن‌های بندهای ارضاننده را که با تغییر ارزش متغیر Φ در جواب فعلی ایجاد شده محاسبه کند.

سایر کاربردهایی که MLN دارد با توجه به آنچه در دانشگاه واشنگتن تدریس می‌شود از ابزار نرم افزاری به نام Alchemy استفاده می‌شود و گفته شده که هر

```

function MaxWalkSAT( $L, m_t, m_f, target, p$ )
  inputs:  $L$ , a set of weighted clauses
            $m_t$ , the maximum number of tries
            $m_f$ , the maximum number of flips
            $target$ , target solution cost
            $p$ , probability of taking a random step
  output:  $soln$ , best variable assignment found
   $vars \leftarrow$  variables in  $L$ 
  for  $i \leftarrow 1$  to  $m_t$ 
     $soln \leftarrow$  a random truth assignment to  $vars$ 
     $cost \leftarrow$  sum of weights of unsatisfied clauses in  $soln$ 
    for  $i \leftarrow 1$  to  $m_f$ 
      if  $cost \leq target$ 
        return "Success, solution is",  $soln$ 
       $c \leftarrow$  a randomly chosen unsatisfied clause
      if Uniform(0,1) <  $p$ 
         $v_f \leftarrow$  a randomly chosen variable from  $c$ 
      else
        for each variable  $v$  in  $c$ 
          compute DeltaCost( $v$ )
         $v_f \leftarrow v$  with lowest DeltaCost( $v$ )
         $soln \leftarrow soln$  with  $v_f$  flipped
         $cost \leftarrow cost + DeltaCost(v)$ 
    return "Failure, best assignment is", best  $soln$  found

```

شکل ۲.۳: شبه کد مربوط به الگوریتم MaxWalkSat

کس که نیاز به پایگاه دانشی با عدم قطعیت دارد Alchemy را مفید خواهد یافت. برای استفاده از این ابزار نیاز به آشنایی با الگوریتم های کلاسیک یادگیری ماشین و منطق مرتبه اول و منطق مارکوف و برخی تئوری های احتمالاتی وجود دارد. کد این نرم افزار برای دانلود قرار داده شده است که به زبان لینوکس می باشد و آموختار یا tutorial آن نیز در سایت قرار داده شده و انواع اپلیکیشن هایی که این ابزار ارائه می دهد نام برده شده است. این آموختار نشان می دهد که یادگیری های متداول چگونه انجام می شود و کابر را با استنتاج و یادگیری در منطق مارکوف آشنا می کند طوری که کاربر بتواند از ابزارهای اولیه برای توسعه اپلیکیشن های مخصوص به خود در این framework استفاده کند. این نرم افزار توسط ارائه دهنده روش شبکه منطقی مارکوف ساخته شده و هنوز همه روش های جدید در آن در نظر گرفته نشده است و تمام دیتابیس های مرتبط با آن هم در اینترنت برای دانلود موجود است [۱].

Alchemy می تواند اعمال اصلی که دو مورد اول شامل یادگیری ساختار^۲ و یادگیری وزن^۳ را انجام دهد. پارامترهای یک مدل با دادن یک دیتابیس آموزشی شامل اتم های پایه است. مورد آخر احتمال یا حالتی که محتمل تر است برای اتم های کوئری با دادن یک پایگاه داده تست شامل اتم های پایه خواهد را شامل می شود. سایر کاربردهای شبکه منطقی مارکوف که در این نرم افزار قابل پیاده سازی است عبارتند از:

- Collective classification
- Link prediction
- Entity resolution
- Social network modeling
- Information extraction

^۲structure learning

^۳weight learning and inference

در ادامه، ما با استفاده از Alchemy و یاد دهی قوانین به آن، طرز پردازش طبیعی شبکه منطق مارکوف را بیان خواهیم کرد.

فصل ۴

آزمایشها و نتایج

هدف اصلی این فصل، انجام يك ارزیابی دقیق و کامل از روشی است که در فصل گذشته توضیح داده شد. در ابتدا نحوه استفاده از ویژگی های مجموعه دادگان توضیح داده می‌شوند و سپس نحوه پیاده سازی و یادگیری شبکه و نتایج تجربی حاصل از پیاده‌سازی روند بیان‌شده مورد بررسی قرار می‌گیرد.

۱.۴ تجزیه و استخراج قواعد از دادگان

در فصل گذشته به چگونگی استخراج دادگان و برچسب زنی آنها و توضیح کامل برچسب‌ها پرداختیم. در این بخش، هدف بیان چگونگی استخراج قواعد از این مجموعه دادگان است. برای درک بهتر، ابتدا به توضیح قالب مجموعه قوانین ساخته شده می‌پردازیم و سپس به بیان چگونگی تبدیل آن به این قواعد خواهیم پرداخت.

۱.۱.۴ قالب مجموعه قوانین ساخته شده

قبل از توضیح مجموعه قواعد، نیازمند تعریف گرامرهای مستقل از متن هستیم. برای توصیف متدهایی که به وسیله آنها در برخی از زبان‌های طبیعی که عبارت‌ها از بلوک‌های کوچکتری ساخته شده‌اند، گرامر مستقل از متن روشی ساده و در عین حال از لحاظ ریاضی دقیق را به وسیله گرفتن ساختارهای بلوکی جملات از راه طبیعی ایجاد می‌کند. سادگی آن باعث می‌شود تا برای رسمی سازی مطالعه ریاضیات سخت جوابگو باشد. نوام چامسکی رسمی سازی گرامر مستقل از متن و همین‌طور کلاس بندی نوع خاصی از گرامر رسمی آن را در اواسط ۱۹۵۰ به وجود آورد. [۳]

گرامر مستقل از متن^۱ با ۴ عنصر $G = \{V, \Sigma, R, S\}$ مشخص می‌شود که دارای ویژگی‌های زیر است:

۱. V یک مجموعه محدود است که هر عضو $v \in V$ را یک متغیر گویند. هر متغیر نشان‌دهنده یک متن یا یک عبارت متفاوت در متن است. متغیرها را در بعضی اوقات دسته نحوی نیز می‌گویند. هر متغیر یک زیر زبان از زبان‌های تعریف شده با G را تعریف می‌کند.

۲. Σ مجموعه محدودی از ترمینال‌هاست که جدا از V محتوای واقعی جمله را تشکیل می‌دهند. مجموعه ترمینال‌ها مجموعه‌ای از حروف الفبا است که با گرامر G تعریف می‌شود.

۳. R یک رابطه محدود از V به $(V \cup \Sigma)^*$ ، که در آن ستاره نشان‌دهنده عملیات ستاره کلین است. اعضای R را قاعده یا محصول گرامر گوئیم.

^۱Context-free Grammar

۴. S را متغیر ابتدایی گوئیم؛ و برای نمایش کل جمله (برنامه) استفاده می‌شود؛ و باید عضو V باشد.

حال می‌توان تجزیه آماری^۲ را تعریف نمود. تجزیه آماری عبارت است از محتمل‌ترین تجزیه از یک جمله با توجه به گرامر مستقل از متن وزن‌دهی شده آن. وزن‌های گرامر مستقل از متن، با توجه به محتوای وزن یادگرفته می‌شوند. حال نگاه مختصری به چگونگی وزن‌دهی از گرامر مستقل از متن به شبکه منطق مارکوف می‌اندازیم. هر گرامر مستقل از متن در فرم نرمال چامسکی^۳ دارای قوانینی با فرم زیر است:

$$A \rightarrow BC$$

$$A \rightarrow a$$

به‌طوری که A و B و C عبارات غیر پایانی^۴ و a یک عبارت پایانی^۵ است. در اینجا می‌خواهیم یک جمله ساده انگلیسی را تجزیه کنیم و برای آن قواعد بسازیم. یک مجموعه قواعد ساده مانند قواعد زیر خواهد بود:

$$S \rightarrow NP VP$$

$$NP \rightarrow Adj N$$

$$NP \rightarrow Det N$$

$$VP \rightarrow V NP$$

^۲Statistical parsing

^۳Chomsky normal form

^۴non-terminal

^۵terminal

که در اینجا S به معنی جمله^۶، NP به معنی عبارت اسمی^۷، N به معنای اسم^۸، Adj به معنای صفت^۹، V به معنی فعل^{۱۰}، VP به معنی عبارت فعل^{۱۱} و Det به معنی ضمیر اشاره^{۱۲} می‌باشد. برای ترجمه این قوانین به زبان MLN، هر قاعده گرامر را به صورت عبارت های زیر می‌نویسیم.

$$NP(i,j) \wedge VP(j,k) \Rightarrow S(i,k)$$

$$Adj(i,j) \wedge N(j,k) \Rightarrow NP(i,k)$$

$$Det(i,j) \wedge N(j,k) \Rightarrow NP(i,k)$$

$$V(i,j) \wedge NP(j,k) \Rightarrow VP(i,k)$$

همانطور که دیده می‌شود، i و j نشان دهنده‌ی اندیس کلمات هستند. پس یک جمله با n کلمه دارای n+۱ اندیس است. برای مثال در جمله‌ی زیر:

$$.The_1 dog_2 chases_3 the_4 cat_5 \quad (1.4)$$

موقعیت کلمه "Dog"، (۱،۲) است. همچنین، ما نیاز به یک تجزیه کننده‌ی ای داریم که قسمت های مختلف جمله را به ما بگوید. مانند:

Determiners

^۶Sentence

^۷Noun Phrase

^۸Noun

^۹Adjevtive

^{۱۰}Verb

^{۱۱}Verb Phrase

^{۱۲}Determiner

Token("a",i) => Det(i,i+1)

Token("the",i) => Det(i,i+1)

Adjectives

Token("big",i) => Adj(i,i+1)

Token("small",i) => Adj(i,i+1)

Nouns

Token("dogs",i) => N(i,i+1)

Token("dog",i) => N(i,i+1)

Token("cats",i) => N(i,i+1)

Token("cat",i) => N(i,i+1)

Token("fly",i) => N(i,i+1)

Token("flies",i) => N(i,i+1)

Verbs

Token("chase",i) => V(i,i+1)

Token("chases",i) => V(i,i+1)

Token("eat",i) => V(i,i+1)

Token("eats",i) => V(i,i+1)

Token("fly",i) => V(i,i+1)

Token("flies",i) => V(i,i+1)

حال میتوانیم ادعا کنیم که تمام قواعد لازم برای یادگیری شبکه منطق مارکوف را دارا هستیم. در بخش بعدی، به توضیح چگونگی تبدیل دادگان خود به شکل توضیح داده شده در این بخش می پردازیم.

۲.۱.۴ چگونگی استخراج قواعد از مجموعه دادگان

مجموعه دادگان فارسی از لحاظ نوشتاری و گفتاری با یکدیگر متفاوت است. از این رو برای تشخیص جملات درست، ما مجبور به استفاده از مجموعه دادگان نوشتاری برای پیشبرد این پایان‌نامه شدیم. برای استفاده از مجموعه دادگان، بعد از برچسب زنی و مشخص کردن دقیق و صحیح نقش هر کلمه در جملات، نیازمند تغییر قالب آن برای یافتن قواعد دستوری زبان فارسی شدیم.

بدین منظور، ما با تغییر داده‌ها و بی اثر نمودن خود کلمات، برنامه‌ای پیاده‌سازی نمودیم که با توجه به نقش کلمات، قواعد را استخراج کند. در حالت اول، ابتدا ما با جملاتی رو به رو بودیم که نقش هر کلمه رو به روی آن داخل پرانتز نوشته شده است. برای درک بهتر، داریم:

```
(S (NP (PRP من)) (VP (NP (NP (DT این)) (NN گفته)) (SBAR (IN که) (S (NP (NN برخی)) (VP (VBD می‌شکستند)) (CC و) (VP (VP (NP (NN یطری)) (VBD می‌کردند)) (PUNC I))) (POSTP را)) (VP (NP (NN قبول)) (VBD ندارم)) (PUNC I))) (S (VP (PP (IN در) (NP (NN بررسی)) (NP (NN العوطاً)) (VBD می‌بینیم)) (SBAR (S (VP (NP (NN م)) (PUNC KK) (NN م)) (PUNC UU))) (NP (NN های گفته)) (NP (NP (NN صحابه)) (NP (NN تابعین)) (NN امتزاج)) (VBD کرده‌است)) (PUNC I))))))
```

شکل ۱.۴: نمونه‌ای از جملات دادگان

بدین منظور، برای استفاده از این دادگان در شبکه منطق مارکوف جهت یادگیری قواعد، ما با از بین بردن کلمات و پرانتزها، توانستیم به مجموعه قوانین به شکل زیر دست یابیم:

در ادامه، یکی از چالش‌های پیش‌رو، تکراری بودن مجموعه قواعد ساخته شده با توجه به تعداد بالای دادگان بود که ما قواعد تکراری و همچنین قواعدی که قاعده‌ی کلی به ما نمیدهد و به اصطلاح دارای حشو است را حذف نمودیم. برای مثال، قانون: «عبارت فعلی = عبارت اسمی + عبارت اسمی + عبارت

$NP(a, b) \wedge VP(b, c) \Rightarrow S(a, c)$
$NN(a, b) \wedge NP(b, c) \Rightarrow NP(a, c)$
$NN(a, b) \wedge PP(b, c) \Rightarrow NP(a, c)$
$PRP(a, b) \Rightarrow NP(a, b)$
$NP(a, b) \wedge VBD(b, c) \Rightarrow VP(a, c)$
$PP(a, b) \wedge NN(b, c) \Rightarrow NP(a, c)$
$CC(a, b) \wedge NP(b, c) \Rightarrow CONJP(a, c)$
$NN(a, b) \wedge PP(b, c) \wedge PP(c, d) \Rightarrow NP(a, d)$

شکل ۲.۴: نمونه ای از قواعد به دست آمده از روی دادگان

اسمی + فعل» و قانون های مشابه آن را با قانون «عبارت فعلی = عبارت اسمی + فعل» جایگزین کردیم زیرا که قانون اول زیر مجموعه ای از قانون دوم است زیرا ما عبارتی مانند : «عبارت اسمی = عبارت اسمی + عبارت اسمی» نیز در مجموعه قوانین خود داریم که مجموع این ۲ قانون، قانون اول را نتیجه خواهد داد. با از بین بردن این قواعد اضافیریال مجموعا به مجموعه قوانین ۸۰ تایی زبان فارسی دست یافتیم.

۲.۴ وزن دهی به شبکه و پیاده سازی

در قسمت قبل، به بررسی چگونگی تولید و جمع آوری قواعد پرداختیم. در قسمت بعدی به پیاده سازی و وزن دهی پرداخته و نتایج حاصل را گزارش خواهیم کرد.

۱.۲.۴ یادگیری وزن ها

در قسمت قبل، با خیل عظیمی از قواعد رو به رو بودیم زیرا که عبارات تکراری بسیار زیادی تولید میشد و ما برای یادگیری به ماشین نیازمند ۸۰ قواعد اصلی زبان فارسی بودیم. از این رو، با بهینه نمودن مجموعه قوانین و از بین بردن تکراری های آن، مجموعه قوانین ۸۰ تایی را به ماشین خود آموزش دادیم. پس از طراحی مجموعه

قوانین و بازگردانی دادگان به شکل منطقی ویژگی‌ها، دادگان آموزشی و قوانین طراحی شده، برای آموزش و نسبت دادن وزن به هر یک از پایه‌سازی‌های ممکن فرمول‌ها استفاده می‌شوند. فرمول‌ها همان مجموعه قوانینی هستند که در مرحله قبل له آنها پرداختیم. این کار، ابتدا برای مدل‌های تشخیص هدف و بر روی تمام جملات انجام شده و سپس با کاهش فضای جستجو، آموزش وزن‌ها برای مدل‌های برجسته زنی دستورزبانی به طور مجزا انجام میشود. بدین منظور برای هر مدل، ۹۰ درصد دادگان برای آموزش و ۱۰ درصد برای آزمون استفاده می‌شود. بدین ترتیب شبکه مارکوف پایه برای دامنه مورد نظر تشکیل داده می‌شود. وزن دهی به این صورت پیاده سازی می‌شود که ماشین با استفاده از دادگان آموزشی، با استفاده از میزان تکرار آن قواعد، میزان درست بودن هر قاعده را به آن توسط نرم فرمال چامسکی اختصاص می‌دهد. سپس در ادامه، با استفاده از این وزن‌ها برای دادگان آزمایشی می‌تواند میزان درستی جملات داده شده را تخمین بزند.

۲.۲.۴ اجرای الگوریتم و نتایج آن

تا اینجای کار، برای شبکه منطق مارکوف خود همه ی الزامات آن را فراهم نموده ایم. در اینجا، برای دادگان آزمایشی خود، با توجه به شبکه یادگرفته شده، میزان درستی تشخیص جملات غلط را بررسی نمودیم. با توجه به وزن دهی قواعد و همچنین دادگان آموزشی، بر روی ۵۰۰۰ داده آزمایشی، بازدهی ۹۹ درصدی از درستی تشخیص جملات غلط دریافت کرده ایم. از این رو، میتوان نتیجه گرفت که شبکه منطق مارکوف، بر اساس احتمالات و قوانین موجود می‌تواند درصد خوبی از درستی جملات زبان فارسی را بررسی کند.

۳.۴ جمع بندی

در این فصل، ما به بررسی چگونگی یادگیری شبکه منطق مارکوف پرداختیم. ابتدا به چگونگی پیدا کردن و تبدیل قواعد زبان فارسی پرداختیم و سپس وزن دهی هر یک از قواعد را با توجه به دادگان آموزشی توضیح دادیم. در انتها دادگان آزمایشی را آزموده و درصد ۹۹ را برای میزان تشخیص درستی جملات دریافت کردیم.

۴.۴ آینده پژوهش

برآیند در آینده این پروژه، به بررسی چگونگی فهم جملات و پاسخگویی آن پردازیم تا شاید بتوانیم تکنولوژی فهم ماشین در زبان فارسی را پیش ببریم.

کتاب نامه

- [۱] Sumner, Marc, and Pedro Domingos. *The alchemy tutorial*. (2013).
- [۲] Al-Muhtaseb, Husni A., Sabri A. Mahmoud, and Rami S. Qahwaji. *Recognition of off-line printed Arabic text using Hidden Markov Models*. *Signal processing* 88.12 (2008): 2902-2912.
- [۳] Richardson, Matthew, and Pedro Domingos. *Markov logic networks* *Machine learning* 62.1 (2006): 107-136.
- [۴] Singla, Parag. *Markov Logic Theory, Algorithms and Applications*, *PhD in Computer Science and Engineering, University of Washington, Seattle, WA*. (2009).
- [۵] Singla, Parag, and Pedro Domingos. *Entity Resolution with Markov Logic*. *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006.
- [۶] Hoifung Poon, Pedro Domingos. *Joint Inference in Information Extraction*. 2007.

- [V] Tsamardinos, Ioannis, Laura E. Brown, and Constantin F. Aliferis. *The max-min hill-climbing Bayesian network structure learning algorithm*. Machine learning 65.1 (2006): 31-78.
- [Λ] Wasserman, Stanley, and Philippa Pattison. *Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p*. Psychometrika 61.3 (1996): 401-425.
- [¶] Poon, H, Domingos, P. *Joint Unsupervised Coreference Resolution with Markov Logic*, Department of Computer Science and Engineering University of Washington. 2007.
- [∧•] Jue Wang, Domingos, P. *Hybrid Markov Logic Networks*. 2007.
- [∧∧] Domingos, Pedro, et al. *Markov logic* Lecture notes in computer science 4911 (2008): 92-117.

فصل ۵

واژه‌نامه فارسی به انگلیسی

Atom.....	اتم
Logical Connective	ادات منطقي
Truth value	ارزش صدق
Grammar Induction	استنتاج گرامر
Cross Validation	اعتبارسنجي متقابل
K-fold Cross Validation	اعتبار سنجي متقابل k تايی
Discriminative	افتراقي
Lexical pattern	الگوي واژگاني
Belief Propagation	انتشار باور
Turing test	آزمون تورینگ
Statistical	آماري
Recall	بازخواني
Pattern Recognition.....	بازشناسي الگو
Speech Recognition	بازشناسي گفتار
Knowledge Representation	بازنمود دانش

Utterance	پاره گفتار
Terminal	پایانه
Knowledge Base	پایگاه دانش
Grounding	پایه سازی
Voted Perceptron	پرسپترون مبتنی بر رأی گیری
Query	پرس و جو
Markov blanket	پوشش مارکوفی
Function	تابع
Partition function	تابع بخش بندی
Potential function	تابع پتانسیل
Concave function	تابع مقعر
Simulated Annealing	تبرید شبیه سازی شده
Parse	تجزیه
Parser	تجزیه گر
Top-down parser	تجزیه گر بالا به پایین
Maximum A-Posteriori estimation	تخمین بیشینه احتمال پسین
Maximum-Likelihood estimation	تخمین بیشینه درست نمایی
Named Entity Recognition	تشخیص موجودیت های دارای نام
Interpretation	تفسیر
Gibbs Distribution	توزیع گیبس
Voice Search	جست و جوی صدا
Speech Summarization	خلاصه سازی گفتار
Clique	خوشه
Spoken Language Understanding	درک زبان گفتار
Speech Topic Classification	دسته بندی موضوعی گفتار

Augmented Transition Network (ATN)	شبکه گذار افزوده
Markov Network	شبکه مارکوف
Ground Markov Network	شبکه مارکوف پایه
Markov Logic Network	شبکه منطق مارکوف
Conditional	شرطي صادق
Total accuracy	صدق پذيري
Case grammar	گرامر حالت
Context Free Grammar	گرامر مستقل از متن
Finite State Machine (FSM)	ماشين حالت محدود
Computing Machinery and Intelligence	ماشين محاسباتي و هوشمندی
Data driven	مبتني بر داده
Markov Model	مدل مارکوف
Hidden Understanding Model	مدل مخفي درك
Hidden Markov Model (HMM)	مدل مخفي مارکوف
Prior semantic model	مدل معنایي پيشين
Lexical model	مدل واژگانی
First Order Logic (FOL)	منطق مرتبه اول
Functional Entity	موجودیت تابعي
Named Entity	موجودیت دارای نام
Artificial Intelligence	هوش مصنوعی
Reinforcement Learning	یادگیری تقويي
Machine Learning	یادگیری ماشین

Abstract

Markov logic networks (MLNs) combine logic and probability by attaching weights to first-order clauses, and viewing these as templates for features of Markov networks. In this paper we develop an algorithm for learning the structure of MLNs from relational databases, combining ideas from inductive logic programming (ILP) and feature induction in Markov networks. The algorithm performs a beam or shortest-first search of the space of clauses, guided by a weighted pseudo-likelihood measure. This requires computing the optimal weights for each candidate structure, but we show how this can be done efficiently. The algorithm can be used to learn an MLN from scratch, or to refine an existing knowledge base. We have applied it in two real-world domains, and found that it outperforms using off-the-shelf ILP systems to learn the MLN structure, as well as pure ILP, purely probabilistic and purely knowledge-based approaches.



College of Science

School of Mathematics, Statistics, and Computer Science

Markov Logic Networks

Application in Natural Language

Understanding

Maryam Vahdatpour

Supervisor: Dr. Bagher BabaAli

A thesis submitted to Graduate Studies Office
in partial fulfillment of the requirements for the degree of
B.Sc. in
Computer Science

July 2017