МЕТОД СИНТАКСИЧЕСКОГО АНАЛИЗА С ИСПОЛЬЗОВАНИЕМ ОПРЕДЕЛЯЕМЫХ ОБУЧАЮЩИМ НАБОРОМ ЯДЕР, ПОСТРОЕННЫХ НА ОСНОВЕ ВЕРОЯТНОСТНЫХ МОДЕЛЕЙ

PARSING WITH KERNELS INDUCED FROM PROBABILISTIC MODELS

И. Титов[1], Дж. Хендерсон[2]

[1]Женевский Университет, Швейцария, [2]Эдинбургский Университет, Великобритания

[1]ivan.titov@cui.unige.ch, [2]james.henderson@ed.ac.uk

В работе представлен метод построения ядра для синтаксического анализа на основе вероятностных моделей. В экспериментах в качестве вероятностной модели используется парсер на базе нейронной сети. Метод обеспечивает статистически значимое улучшение точности по отношению к базовому парсеру и результаты на уровне лучших парсеров английского языка.

Введение

Методы, основанные на ядрах, продемонстрировали эффективность во многих задачах из области машинного обучения. Ядра позволяют использовать методы, оптимизирующих меры, напрямую связанные с ожидаемой точностью на тестовом наборе (методы, максимизирующие зазор). В отличие от них, вероятностные меры, используемые в статистических моделях, лишь косвенно связаны с обобщающими способностями метода.

Ядро – это функция, представляющая собой скалярное произведение векторов признаков в многомерном (или бесконечномерном) пространстве $K(x,y) = \varphi(x)^T \varphi(y)$, где φ - функция, отображающая данные в пространство признаков. Таким образом, ядро в явном или неявном виде задает отображение данных в пространство признаков.

Большинство работ по построению ядер для задач обработки естественного языка (ОЕЯ) было сконцентрировано на ядрах над деревьями разбора (см., например, [1, 11]). Все эти ядра определяли в явном виде свойства деревьев полезные, по мнению авторов, для классификации деревьев на корректные и некорректные, и, в тоже время, допускающие эффективное их вычисление.

В тоже время в теории машинного обучения рассматривался и другой подход, при котором ядро строится на основе вероятностной модели [8,12]. Построения ядер на основе вероятностных моделей обладает рядом преимуществ.

Во-первых, лингвистические знания отражаются в построении вероятностной модели, а не самого ядра. Построение вероятностных моделей на настоящий момент хорошо изученный процесс как с точки зрения отражения обобщения в обучающем наборе, так и с точки зрения контроля вычислительных расходов. В тоже время, большинство пространств в ОЕЯ неограничено в размере и сложности. В этом случае трудно непосредственно определить все возможные полезные свойства для ядер без получения такого количества свойств, когда вычислительная сложность ядра делает его неприменимым для практических задач или когда данные становятся чрезвычайно разреженными[1].

Во-вторых, ядро определено с использованием значений параметров вероятностной модели, полученных в результате обучения. Следовательно, ядро частично определено обучающим набором и, тем самым, автоматически отражает свойства, полезные для парсинга.

В настоящей работе предлагается новый метод для получения ядра из вероятностной модели, который специально привязан к задаче перестановки (reranking) гипотез [4]. В задаче перестановки гипотез, применяемой в парсинге, для каждого предложения предоставляется набор деревьев разбора, лучших в соответствии с базовой моделью, а модель, осуществляющая перестановку, выбирает гипотезу из списка. В качестве вероятностной модели мы используем статистический парсер на основе нейронной сети [6]. Метод, использующий предлагаемое ядро и перцептрон с голосованием [5] для перестановки 20 гипотез, предлагаемых статистическим парсером, позволяет достичь статистически значимого улучшения по сравнению с базовой моделью для стандартной задачи синтаксического анализа корпуса Penn Tree Bank Wall Street Journal [9].

Построение ядер на основе вероятностных моделей

В последние годы было предложено несколько методов построения ядер на основе обученных вероятностных моделей. Наряду с положительными

[1] См., например, [7], где обсуждается, почему генеративные вероятностные модели лучше, чем модели, параметризированные для оценки апостериорной вероятности.

результатами применения этих ядер к некоторым практическим задачам, их примение обусловлено теоретическими результатами, которые позволяют ожидать улучшение точности таких классификаторов по сравнению с выбором наиболее вероятной гипотезы в соответствии с исходной моделью. В настоящем разделе будет рассмотрена применимость двух предшествующих ядер, а так же предложено новое ядро, построенное специально для задачи перестановки гипотез.

Для начала рассмотрения ядер нам необходимо сформулировать синтаксический анализ как задачу классификации. Синтаксический анализ может рассматриваться как построение отображения из пространства предложений $x \in X$ в пространство деревьев разбора $y \in Y$ (структурных классов). На основе обучающего набора строится дискриминантная функция $F : X \times Y \to \Re$, где $\Re$ - множество вещественных чисел. Модель возвращает для предложения x дерево разбора y, которому соответствует максимальное значение функции $F(x, y)$. В настоящей работе рассматриваются линейные дискриминантные функции $F_w(x, y) = w^T \varphi(y)$. Далее мы будем характеризовать ядра в терминах, соответствующих им функций отображения $\varphi(y)$.

Ядра Фишера

Пусть задана гладко параметризированная вероятностная генеративная модель $P(z \mid \hat{\theta})$, в качестве функции отображения в ядре Фишера [8] используется градиент логарифма правдоподобия:

$$\phi_{\hat{\theta}}(z) = (\frac{\partial \log P(z \mid \hat{\theta})}{\partial \theta_1},...,\frac{\partial \log P(z \mid \hat{\theta})}{\partial \theta_l}).$$

Можно рассматривать данный вектор, как информацию о том, как должна быть изменена модель для максимизации правдоподобия z. Такое ядро называют *практическим ядром Фишера. Теоретическое ядро Фишера* зависит от Информационной Матрицы, вычисление которой не представляется возможным для большинства практических задач.

Ядро Фишера непосредственно предназначено лишь для бинарной классификации, но оно может быть применено к парсингу, если рассматривать его как классификацию деревьев на корректные и некорректные и считать дерево входом модели. При использовании $\varphi_{\hat{\theta}}(y)$ в дискриминантной функции F, мы можем интерпретировать ее значения как меру уверенности в том, что дерево y правильное, и выбирать те значения y, в которых наиболее уверены.

Ядра TOP

Тсуда [12] предложил другое ядро, конструируемое на основе вероятностной модели, называемое ядро градиентов разности логарифмов апостериорной вероятности (Tangent vectors Of Posterior log-odds kernel, TOP). Его ядро TOP также предназначено только для бинарной классификации. Так же, как и раньше, будем рассматривать y как вход, а выходную категорию $c \in \{+1,-1)$ как правильно/неправильно. Ядро строится таким образом, чтобы минимизировать интеграл вероятности ошибки оптимального линейного классификатора для задачи бинарной классификации. Функция отображения в пространство свойств в этом случае будет задана следующим выражением:

$$\varphi_\theta = (v(y,\hat{\theta}),\frac{\partial v(y,\hat{\theta})}{\partial \theta_1},...,\frac{\partial v(y,\hat{\theta})}{\partial \theta_l}),$$

где $v(y,\hat{\theta}) = \log P(c = +1 \mid y,\hat{\theta}) - \log(1 - P(c = +1 \mid y,\hat{\theta}))$.

В связи с использованием интеграла вероятности ошибки для бинарной классификации предлагаемый Тсудой подход не является адекватным для задачи перестановки и парсинга в целом. Вычисление условной вероятности $P(c = +1 \mid y,\hat{\theta}) = P(y \mid x,\hat{\theta})$ - также нетривиальная задача при использовании модели генеративной вероятности дерева разбора $P(y \mid \hat{\theta})$, так как число возможных деревьев разбора для заданного предложения x очень велико[2].

Ядра TOP для Перестановки

Определим задачу перестановки как задачу выбора дерева разбора из списка кандидатов, предоставляемого фиксированной базовой моделью. В этом случае интеграл вероятности ошибки оптимального линейного классификатора будет зависеть от точности моделирования вероятности:

$$P(y_k \mid y_1,..,y_s) = \frac{P(y_k)}{\sum_t P(y_j)},$$

где $y_1,...,y_s$ - список кандидатов. Руководствуясь требованием минимизации интеграла ошибок, получим следующее отображение:

[2] Метод оценки условной вероятности и ее градиента для используемой в настоящей работе статистической модели синтаксического анализа обсуждается в [7].

$\phi_{\hat{\theta}}(y_k) = (v(y_k, \hat{\theta}), \frac{\partial v(y_k, \hat{\theta})}{\partial \theta_1}, ..., \frac{\partial v(y_k, \hat{\theta})}{\partial \theta_l})$, где

$v(y_k, \hat{\theta}) = \log P(y_k \mid y_1, ..., y_s, \hat{\theta}) -$

$\log \sum_{t \neq k} P(y_k \mid y_1, ..., y_s, \hat{\theta})$.

Назовем соответствующее ядро - *ядром TOP для Перестановки*.

Вероятностная модель

Для завершения определения ядра необходимо выбрать вероятностную модель парсинга. Мы используем парсер, предложенный в [6]. Как и во многих других парсерах (см., например, [3]) в рассматриваемом статистическом парсере используется модель вероятности, использующая предысторию. Парсер определяет взаимоднозначное отображение дерева разбора в последовательность решений $d_1, ..., d_m$, используя модификацию стратегии левоугольного парсинга [6]. Вероятность дерева разбора или, что тоже самое, вероятность всей последовательности действий парсера может быть представлена как произведение условных вероятностей каждого действия парсера d_i:

$$P(d_1, ..., d_m) = \prod_i P(d_i \mid d_1, ..., d_{i-1}).$$

Описанная модель определена с использованием бесконечного числа парметров – вероятностей $P(d_i \mid d_1, ..., d_{i-1})$. В рассматриваемом подходе нейронная сеть в процессе обучения определяет конечное представление истории $h(d_1, ..., d_{i-1})$ - вектор активаций скрытого слоя:

$$P(d_i \mid d_1, ..., d_{i-1}) \approx P(d_i \mid h(d_1, ..., d_{i-1})).$$

Используя архитектуру, называемую Простые Синхронные Сети (Simple Synchrony Networks, SSN), вещественный вектор $h(d_1, ..., d_{i-1})$ инкрементально вычисляется на основе $h(d_1, ..., d_{i-2})$ и конечного числа векторов $h(d_1, ..., d_j)$, $j < i-1$. Выбор этих вектор осуществляется на основе лингвистически мотивированной структурной близости.

После определения вектора активации скрытого слоя расчет распределения вероятности следующих решений парсера производится с использованием нормализованной экспоненты:

$$P(d_i \mid d_1, ..., d_{i-1}, \hat{\theta}) = \frac{\exp(\hat{\theta}_{d_i}^T h(d_1, ..., d_{i-1}))}{\sum_{t \in N(d_{i-1})} \exp(\hat{\theta}_t^T h(d_1, ..., d_{i-1}))},$$

где θ_t - вектор весов выходного слоя, соответствующего действию парсера t, $N(d_{i-1})$ - множество потенциально возможных шагов парсера следующих за шагом d_{i-1}.

При обучении используется модификация алгоритма обратного распространения ошибки для метода градиентного спуска с логарифмом функции правдоподобия в качестве целевой функции.

Параметризация модели для эффективного расчета ядра

В качестве параметров для построения ядра используется набор весов нейронной сети. Экспериментальные результаты показали, что фиксирование всех весов сети, кроме весов выходного слоя, позволяет не только ускорить расчет ядра, но и добиться более высокой точности результирующего метода (см. раздел 5). Наряду с оценкой вероятности шагов парсера, генеративная модель должна рассчитывать вероятность слова при заданной истории. Следовательно, в случае использования большого словаря, мощность множества $N(d_{i-1})$ будет велика, что приводит к крайне высокой размерности пространства признаков. Рассмотренные ранее ядра определяют неразреженные вектора в этом пространстве - вектора зависят от производных знаменателя в выражении для $P(d_i \mid d_1, ..., d_{i-1}, \hat{\theta})$, где участвует не только вектор весов $\hat{\theta}_{d_i}$, соответствующий следующему действию парсера, но и веса для всех потенциально возможных следующих решений парсера $\hat{\theta}_t$, $t \in N(d_{i-1})$.

Для решения этой проблемы мы предлагаем зафиксировать нормирующий множитель при расчете вектора признаков, осуществив репараметризацию модели. Перепишем выражение для вероятности дерева разбора следующим образом:

$$\log P(y \mid \hat{\theta}) =$$

$$\sum_i \log(\frac{\exp(\hat{\theta}_{d_i}^T h(d_1, ..., d_{i-1}))}{\sum_{t \in N(d_{i-1})} \exp(\hat{\theta}_t^T h(d_1, ..., d_{i-1}))}) =$$

$$\sum_i (\hat{\theta}_{d_i}^T h(d_1, ..., d_{i-1})) -$$

$$\sum_i \log(\sum_{t \in N(d_{i-1})} \exp(\hat{\theta}_t^T h(d_1, ..., d_{i-1}))).$$

Тогда, принимая параметры, входящие в слагаемое, соответствующее числителю, за параметры независимые от параметров, входящих в знаменатель, мы определяем *Эффективное ядро TOP для Перестановки* с использованием лишь

параметров, входящих в числитель. Это означает, что функция отображения $\phi_{\hat{\theta}}(y_k)$ для ядра будет содержать ненулевые компоненты только в первой компоненте и в компонентах, соответствующих решениям d_i, присутствующим в кандидатах для данного предложения.

Экспериментальные результаты

Мы рассматривали стандартную задачу парсинга корпуса Penn Treebank WSJ [9] для получения практических результатов[3]. Для слов, частота которых в обучающем наборе ниже фиксированного порога, при синтаксическом анализе нами учитывается только их идентификатор части речи.

Были приведены две серии экспериментов: с частотным порогом для формирования словаря 200, что соответствует 508 словам в словаре и с частотным порогом 20, что приводит к 4215 словам[4]. Для обучения использовалась модификация алгоритма Перцептрон с Голосованием [5]. Проведены эксперименты по обучения методов на основе всех рассмотренных ядер с малым словарем, и лишь для методов на основе Эффективного ядра TOP для Перестановки с большим словарем. Во избежании повторного тестирования модели сравнивались на наборе для промежуточного тестирования (таблица 1). Частотный порог для модели приведен в скобках. Заметим, что этот набор не использовался при обучении или для подбора параметров методов, использующих ядра. Для вычисления точности и полноты использовалась стандартная утилита *evalb* [3], мера F_1 - их гармоническое среднее.

Метод (ядро)	набор параметров	полнота	точность	F_1
TOP для Перестановки (200)	все веса	86.8	88.4	87.6
Базовая модель (200)	**-**	**87.2**	**88.5**	**87.8**
TOP (200)	вых. слой	87.1	88.8	87.9
Ядро Фишера (200)	вых. слой	87.2	88.8	87.9
TOP для Перестановки (200)	вых.слой	87.3	88.9	88.1
Эфф. TOP для Перестановки (200)	вых.слой	87.3	88.9	88.1
Базовая модель (20)	**-**	**88.1**	**89.2**	**88.6**
Эфф. TOP для Перестановки (20)	вых.слой	88.2	89.7	88.9

Таблица 1. Сравнение моделей на наборе для промежуточного тестирования

Заметим, что улучшения в мере F_1 по отношению к базовому парсеру, продемонстрированные методами, использующими ядра TOP для Перестановки, являются статистически значимыми[5]. Также отметим, что модификация этих ядер для эффективного вычисления демонстрирует такие же результаты. По мнению авторов, лучшие результаты этих ядер объясняются тем, что они построены специально для задачи перестановки.

В Таблице 2 представлено сравнение нашей лучшей модели с результатами других статистических парсеров (см. [1, 2, 3, 4, 6, 10, 11] и обзор результатов в [7]) на тестовом наборе. Во-первых, заметим, что парсер, использующий Эффективное ядро TOP для Перестановки, демонстрирует результаты лучшие чем модель, использующая ту же модель синтаксического анализа (парсер "*Henderson03*"), хотя обученные параметры и были другими. По сравнению с другими методами, использующими ядра, наш метод демонстрирует результаты значительно лучшие, чем парсеры на базе ядра свертки для деревьев разбора (*"CollinsDuffy02*", *"CollinsRoark04"*) и лишь на 0.2% хуже, чем лучшие методы, использующие ядра (парсеры "*ShenJoshi04*" и "*Shen и др. 03*").

Парсер	Полнота	Точность	F_1 [6]
Collins99	88.1	88.3	88.2
CollinsDuffy02	88.6	88.9	88.7
CollinsRoark04	88.4	89.1	88.8
Henderson03	88.8	89.5	89.1
Charniak00	89.6	89.5	89.5
Эфф. TOP для Перест. (20)	**89.1**	**90.1**	**89.6**
Collins00	89.6	89.9	89.7
ShenJoshi04	89.5	90.0	89.8
Shen и др. 03	89.7	90.0	89.8
Henderson04	89.8	90.4	90.1
Bod03	90.7	90.8	90.7

Таблица 2. Сравнение парсеров на стандартном наборе для тестирования

Заключение

В настоящей работе предложен метод построения ядра для перестановки гипотез

[3] Стандартный набор для обучения (секция 2-22, 39 832 предложений), предварительного тестирования (секция 24, 1346 предложений) и тестирования (секция 23, 2416 предложений) [3].

[4] Величина порога 20 соответсвуюет наилучшим результатам базовой модели на наборе для предварительного тестирования [6].

[5] Статистическая значимость устанавливалась с использованием теста значимости, предложенного в [13].

[6] Мера F_1 для предшествующих моделей может содержать ошибку округления.

(reranking) на основе вероятностной модели, демонстрирующий результаты на уровне лучших методов при применению к проблеме синтаксического анализа естественного языка. Примение описанного метода позволяет достичь результатов, которые лишь на 0.2% уступают лучшему парсеру, использующему ядра, и соответствуют лучшим результатам современных статистических парсеров.

В последние годы использование вероятностных моделей стало стандартными методами для решения задач из области естественного языка, и рассмотренный подход к определению ядер может быть очень полезен для многих из них. Особенно это оправдано для случаев с меньшим доступным объемом обучающих данных, так как при этом методы, максимизирующие зазор, демонстрируют значительное преимущество по сравнению с другими подходами.

Список литературы

1) Collins M. and Duffy N., New ranking algorithms for parsing and tagging: Kernels over discrete structures and the voted perceptron // In Proc. 40th Meeting of Association for Computational Linguistics, 2002.
2) Collins M. and Roark B., Incremental parsing with the perceptron algorithm // In Proc. 42nd Meeting of Association for Computational Linguistics, Barcelona, Spain, 2004.
3) Collins M., Head-Driven Statistical Models for Natural Language Parsing // PhD thesis, University of Pennsylvania, Philadelphia, PA, 1999.
4) Collins M., Discriminative reranking for natural language parsing // In Proc. 17th Conf. on Machine Learning, Stanford, CA, 2000.
5) Freund Y. and Shapire R.E., Large margin classification using the perceptron algorithm // In Proc. 11th Annual Conf. on Computational Learning Theory, Madisson, WI, 1998.
6) Henderson J., Inducing history representations for broad coverage statistical parsing // In Proc. Joint meeting of North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conf., Edmonton, Canada, 2003.
7) Henderson J., Discriminative training of a neural network statistical parser // In Proc. 42nd Meeting of Association for Computational Linguistic, Barcelona, Spain, 2004.
8) Jaakkola T.S. and Haussler D., Exploiting generative models in discriminative classifiers // Advances in Neural Information Processing Systems 11, 1998.
9) Marcus M.P., Santorini B., Marcinkiewicz M.A., Building a large annotated corpus of English: The Penn Treebank // Computational Linguistics, 19(2), 1993.
10) Shen L. and Joshi A.K., Flexible margin selection for reranking with full pairwise samples // In Proc. 1st Int. Joint Conference on Natural Language Procession, Hainnan Island, China, 2004.
11) Shen L., Sarkar A. and Joshi A.K., Using LTAG based features in parse reranking // In Proc. of Conf.on Empirical Methods in Natural Language Processing, Sapporo, Japan, 2003.
12) Tsuda K., Kawanabe M., Ratsch G., Sonnenburg G. and Muller K., A new discriminative kernel from probabilistic models // Neural Computation, 14(10), 2002.
13) Yeh A., More accurate tests for the statistical significance of the results differences // In Proc. 17th Int. Conf. on Computational Linguistics, Saarbrucken, Germany, 2002.