Машинное обучение - состояние и перспективы

© Д.П. Ветров

VetrovD@yandex.ru

Аннотация

В работе кратко охарактеризованы основные направления развития теории машинного обучения в настоящее время. Более подробно изложен аппарат т.н. вероятностных графических моделей, являющийся на сегодня популярным средством построения новых моделей и алгоритмов машинного обучения, учитывающих специфику конкретной прикладной задачи. Также рассмотрен байесовский язык описания вероятностных моделей, кратко изложены непараметрические байесовские методы и методы глубинного обучения.

1 Задачи машинного обучения

Теория машинного обучения зародилась практически одновременно с появлением первых компьютеров и на протяжении последних 70 лет является активно развивающейся дисциплиной. Ее постоянное развитие вызвано ростом возможностей современных вычислительных систем, еще более стремительным ростом объемов данных, доступных для анализа, а также постоянным расширением области применения методов машинного обучения на все более широкий класс задач обработки данных. Машинное обучение работает с объектами - элементарными единицами данных, естественным образом, возникающими в конкретных задачах, которые характеризуются наблюдаемыми переменными $\vec{x}$ и скрытыми переменными $\vec{t}$, принимающими значения из некоторых заранее известных множеств. Главной задачей машинного обучения является автоматическое определение взаимозависимостей между наблюдаемыми и скрытыми переменными объекта, с тем, чтобы для произвольного объекта по его наблюдаемым компонентам можно было оценить возможные значения скрытых компонент. Как правило, возможные взаимозависимости задаются заранее с помощью параметрических решающих правил, определяемых значением параметров (весов) $\vec{w}$. Конкретные значения $\vec{w}$ определяются в ходе обучения с исопльзованием обучающей выборки, представляющей собой множество объектов с известными наблюдаемыми и скрытыми переменными (X^{tr}, T^{tr}) (обучение с учителем) или только наблюдаемыми переменными X^{tr} (обучение без учителя). При этом задача определения весов решающего правила $\vec{w}$ по обучающей выборке называется задачей обучения или настройки (training), а задача определения допустимых значений скрытой переменной $\vec{t}$ по заданым наблюдаемым компонентам $\vec{x}$ объекта и заданным весам решающего правила $\vec{w}$ — задачей вывода (inference). Обычно (но не обязательно) предполагается, что каждый объект описывается одним и тем же набором переменных, а номенклатура наблюдаемых и скрытых переменных для всех объектов одинакова. Примером такой стандартной задачи является задача классификации, в которой скрытая переменная для каждого объекта одна и принимает значения из конечного дискретного множества, а каждая наблюдаемая переменная может принимать действительные, либо (реже) дискретные значения. Если скрытая переменная объекта является не дискретной, а непрерывной, задача называется задачей восстановления регрессии, являющейся еще одной стандартной и хорошо изученной задачей машинного обучения.

В разное время предпринимались неоднократные попытки ввести некоторый универсальный язык описания различных постановок и методов решения задач машинного обучения. Начиная с 90ых гг прошлого века широкое распространение получил т.н. байесовский формализм. При его использовании предполагается, что зависимости между наблюдаемыми переменными объекта, весами решающего правила и скрытыми переменными объекта моделируются с помощью совместного распределения на эти группы переменных $p(X, T, \vec{w})$. Если нас интересует только задача определения скрытых переменных по наблюдаемым, рассматривают дискриминативные модели (discriminative models) $p(T, \vec{w}|X)$. Значения наблюдаемых переменных X в этом случае не моделируются, предполагаясь

Proceedings of the 15th All-Russian Conference "Digital Libraries: Advanced Methods and Technologies, Digital Collections— RCDL-2013, Yaroslavl, Russia, October 14-17, 2013.

известными на всех этапах решения задачи, и совместное распределение становится проще. В стандартных постановках задачи машинного обучения предполагалось, что скрытые переменные каждого объекта зависят только от наблюдаемых переменных этого объекта, причем вид зависимости определяется параметрами $\vec{w}$. Это соответствует представлению

$$p(T, \vec{w}|X) = \prod_{i=1}^{n} p(\vec{t}_i|\vec{x}_i, \vec{w})p(\vec{w}).$$

При использовании такого формализма задача настройки параметров $\vec{w}$ решается, например, нахождением наиболее вероятного значения

$$\vec{w}_{MP} = \arg\max p(\vec{w}|X^{tr}, T^{tr}) =$$
$$\arg\max \frac{p(T^{tr}, \vec{w}|X^{tr})}{p(T^{tr}|X^{tr})} = \arg\max p(T^{tr}, \vec{w}|X^{tr}),$$

а задача вывода — путем нахождения[1]

$$\hat{\vec{t}}(\vec{x}) = \arg\max p(\vec{t}|\vec{x}, \vec{w}_{MP}).$$

Таким образом, для формулировки и решения задачи машинного обучения нам достаточно знать две функции: $p(\vec{t}|\vec{x}, \vec{w})$ и $p(\vec{w})$. Если с первой функцией, называемой функцией правдоподобия (likelihood), проблем обычно не возникает, т.к. она естественным образом характеризует степень «истинности» полученного прогноза на скрытую переменную, то вторая компонента, наызваемая априорным распределением (weight prior) или регуляризатором (regularizer), долгое время вызывала споры. В самом деле, меняя априорное распределение, мы влияем на результат процедуры настройки, т.е. на $\vec{w}_{MP}$. При этом способ адекватного определения априорного распределения неочевиден. В 90ые гг. в ряде работ [?] было убедительно показано, что априорное распределение является эффективным способом контроля сложности решающего правила и позволяет осуществлять регуляризацию процедуры настройки. Вместо нахождения весов, обеспечивающих наименьшую ошибку прогноза на обучающей выбобрке (что чревато эффектом переобучения (overfitting)) мы жертвуем толикой точности ради сохранения способности обеспечить ту же ошибку прогноза на других объектах генеральной совокупности. Оказалось, что в любой модели машинного обучения можно выделить самое простое решающее правило (например, отвечающее нулевым значениям весов), в которое помещается мода унимодального априорного распределения. Чем больше расстояние текущих значений весов от моды, тем меньше значение $p(\vec{w})$. Ширина же априорного распределения задается параметром регуляризации, который может быть сравнительно эффективно найден процедурой скользящего контроля (cross-validation) или байесовской процедурой выбора модели (Bayesian model selection). Еще более привлекательным свойством байесовского формализма оказалась возможность учитывать многочисленные априорные знания о возможных зависимостях между наблюдаемыми и скрытыми переменными объектов, которые имеются во многих прикладных задачах. Например, известно, что надежность заемщика (прогнозируемая переменная) должна положительно коррелировать с его доходом и образованием (наблюдаемые переменные). Такие «подсказки» алгоритмам общего назначения, выраженые в виде априорного распределения на $\vec{w}$, позволили добиться значительного увеличения точности и снизить эффект переобчения, благодаря адаптации их под специфику конкретной задачи.

Можно показать, что практически любую задачу машинного обучения возможно (с большей или меньшей степенью естественности) свести к такому формализму. Это, в свою очередь, открывает унифицированный способ анализа различных моделей машинного обучения, например, с целью исследования их обобщающей способности или выработки эффективных приближеных методов настройки и вывода общего назначения.

2 Современные направления развития теории машинного обучения

С конца 90ых гг. байесовский формализм при описании алгоритмов машинного обучения получил всеобщее признание [1]. В рамках него удалось разработать ряд общих методов для оценки апостериорных распределений, байесовского вывода, автоматического выбора модели и пр. Не менее важным успехом байесовского формализма стала возможность успешного обобщения результатов и методов классического машинного обучения на совершенно новые задачи (см. например, [2]).

2.1 Глубинное обучение

Методы глубинного обучения (deep learning) являются попыткой реинкарнации

[1] Строго говоря, полностью байесовские процедуры настройки и вывода предполагают нахождение апостериорных распределений $p(\vec{w}|X^{tr}, T^{tr})$ и $p(\vec{t}|\vec{x}, X^{tr}, T^{tr})$ вместо соответствующих точечных оценок, поэтому последние можно рассматривать как детерминированные приближения случайных величин, например, в смысле дивергенции Кульбака-Лейблера

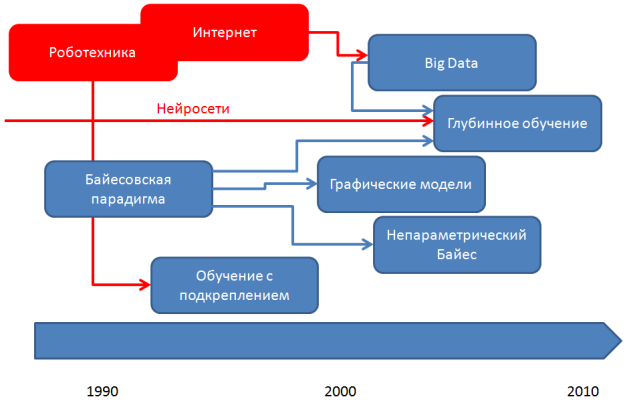

Рис. 1: Приблизительная хронологическая карта появления новых направлений в машинном обучении

нейронных сетей, с конца 80ых гг. прошлого века переживающих кризис. Причинами кризиса традиционных нейронных сетей стали: критическая зависимость качества настройки весов сети от выбора начального приближения и, как следствие, проблемы с воспроизводимостью «успешных» результатов, публиковавшихся в научных журналах; большая подверженность переобучению вкупе со слабыми возможностями контроля обобщающей способности сети; большоее количество локальных минимумов функционала качества, большинство из которых оказывались плохими. С другой стороны, неоспоримой сильной стороной нейронных сетей явилось открытие метода обратного распространения ошибки (backpropagation), позволявшего отслеживать влияние внутренних слоев сети на качество прогноза скрытых переменных объектов обучающей выборки.

Во второй половине 00ых гг стало активно развиваться направление, получившее название глубинного обучения [4]. В его основе лежат нейроные сети, претерпевшие значительные изменения:

- Глубиннное обучение строит не дискриминативные, а порождающие модели (generative models), в которых моделируется общее распределение $p(X, T, \vec{w})$, в отличие от дискриминативных моделей, позволяющее, например, генерировать новые объекты.
- В наиболее распространенной постановке все переменные объектов предполагаются бинарными. Это облегчает моделирование зависимостей между переменными объекта.
- Каждый слой сети сначала обучается независимо, проходя процедуру предобучения (pre-training). Это позволяет «нащупать» хорошее начальное приближение для последующего запуска алгоритма обратного распространения ошибки. Каждый слой, в зависимости от выбранной модели, представляет собой ограниченную машину Больцмана (restricted Boltzmann machine) или сверточную сеть (convolutional network).
- Для обучения используются сотни тысяч и миллионы объектов. Такие гигантские выборки позволяют настраивать сети с десятками тысяч параметров, без риска переобучения. Обученные таким образом сети, не просто позволяют моделировать сложные объекты (например, тексты или изображения), но и генерируют в процессе обучения информативные признаковые описания, которые могут быть использованы другими, более простыми алгоритмами машинного обучения в качестве наблюдаемых переменных объекта.

Методология глубинного обучения позволила добиться невиданых ранее результатов при обучении на больших и сверхбольших объемах данных. В настоящее время она является одним из наиболее перспективных путей развития машинного обучения.

2.2 Непараметрические байесовские методы

Традиционно, методы непараметрической статистики определялись как раздел статистики, в которой число параметров, описывающих данные (например, парамеры плотности распределения объектов) не фиксированно, а растет с ростом числа объектов. Чтобы разъяснить принципы работы непараметрических байесовских методов (non-parametric Bayes), рассмотрим задачу определения числа кластеров (скоплений объектов) в растущей выборке объектов. Данная задача тем более актуальна, что общепринятых методов определения, а из скольки же кластеров состоит даже зафиксированная выборка, на сегодняшний день не существует. Чем больше объектов поступает в наше распоряжение, тем с большим разрешением мы можем находить в них структуру, выделяя кластеры схожих между собой объектов. В случае достаточно неоднородной выборки число кластеров должно постепенно увеличиваться по мере поступления новых объектов. Возникает вопрос, можно ли задать наши представления о том, как быстро должно расти число кластеров с ростом данных (чтобы их не было слишком много или слишком мало) и как, глядя на выборку объектов, учесть эти представления. Формально, ответ может быть задан знаменитой формулой Байеса, которая как раз и объединяет наши априорные представления

с текущими наблюдениями

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}.$$

В непараметрическом случае, нам необходимо задать распределение над всевозможными разбиениями произвольного количества объектов. Такое распределение (как и многие другие в непараметрических байесовских методах) задается с помощью случайных процессов. В данном случае, это процесс Дирихле (Dirichlet process), также известный как процесс китайского ресторана (Chinese restraunt process) [7].[2] С его помощью, удается не только расчитать для любого разбиения проивольного числа объектов на кластеры его априорную вероятность, но и учесть характеристики объектов (их наблюдаемые переменные), чтобы перейти к апостериорному распределению на всевозможные разбиения. Как это часто бывает при применении байесовских методов, апостериорное распределение имеет острый пик, который соответствует устойчивому разбиению выборки объектов на некоторое число клстеров. Фактически, процесс Дирихле позволяет задавать распределения над всевозможными дискретными распределениями. При выводе используются приближенные методы Монте-Карло с марковскими цепями (Markov chain Monte Carlo) и методы вариационного вывода (variational inference). Описанная схема допускает многочисленные обобщения на случай иерархий кластеров, множественных выборок, и др.

2.3 Обучение с подкреплением

Еще одной активно развивающейся областью машинного обучения является обучение с подкреплением, предназначенное для обучения агентов (автономных модулей, самостоятельно принимающих решения в реальном времени на основании располагаемых данных) в условиях неопределенности, порождаемой, как неполнотой информации об окружающей обстановке, так и возможными действиями других агентов. В зависимости от текущего состояния среды и действий агентов расчитывается функция выгоды, которую получит агент в следующий момент времени. В роли наблюдаемых переменных объекта выступает информация, располагаемая агентом, а скрытыми переменными являются долгосрочные оценки полученной выгоды. Важным достоинством алгоритмов обучения с подкреплением является возможность обучения агента «с нуля» за счет балансируемого сочетания режимов «исследование-использование» (exploration-exploitation) и выучивания стратегий, позволяющих жертвовать малым сейчас ради получения большей выгоды в дальнейшем. Алгоритмы обучения с подкреплением нашли широкое применение не только в таких традиционных областях как роботехника, но и, например, на фондовых рынках.

2.4 Анализ больших объемов данных

Термин «большие данные» (англ. big data) вошел в употребление в конце 2000-х годов, когда стал возможным сбор и хранение огромных объемов данных. Феномен больших данных можно наглядно продемонстрировать на примере большого адронного коллайдера, который в прошлом году произвел около 25 петабайт экспериментальных данных [3]. Традиционные методы машинного обучения не всегда применимы для анализа выборок такого размера, поскольку в них зачастую неявно предполагается, что вся выборка помещается в памяти компьютера, или же они имеют недостаточно высокие показатели масштабируемости (скорости роста вычислительной сложности в зависимости от размера выборки). Для преодоления этих ограничений часто используются приемы из следующих категорий:

- Распараллеливание. Независимые части алгоритма могут выполняться параллельными обработчиками (в т.ч. на разных компьютерах) и в произвольном порядке. В некоторых случаях параллельной реализации классичесского алгоритма может быть достаточно для конкретной задачи. В той или иной форме параллельность лежит в основе практически всех вычислительных систем, ориентированных на большие данные. Примечательно, что параллельность накладывает существенные ограничения на взаимодействие между обработчиками, так как накладные расходы на «общение» между ними может превышать выигрыш от использования большого вычислительного кластера.

- Аппроксимация. Известно, что многие сложные задачи могут быть решены приближенно с достаточно большой (а иногда и контролируемой) точностью, достаточной для данного эксперимента. Примерами могут служить фильтр Блума или приблеженный алгоритм поиска ближайшего соседа, которые допускают ошибки первого рода, но имеют существенно

[2] Вообще, терминология в непараметрическом Байесе грешит восточными гастрономическими наклонностями. Известен еще процесс китайской франшизы [ресторанов] и процесс индийского буфета :)

более низкую вычислительную сложность чем их «точные» аналоги.

- Стохастичность (рандомизация). При наличии большого числа независимых объектов в выборке, многие необходимые статистики могут быть оценены по случайной подвыборке, при этом сохраняются теоретические гарантии оптимальности и сходимости алгоритма. В случае, если выбирается подвыборка некоторого фиксированного размера это позволяет получать алгоритмы с сублинейной масштабируемостью. Наиболее известным алгоритмом, где применяется данный подход, является метод стохастического градиентного спуска.

В последнее время стали также набирать популярность т.н. потоковые алгоритмы (streaming algorithms, online learning), способные обучаться инкрементально в режиме реального времени на постоянно поступающих данных без необходимости хранить их где-либо в памяти. Спрос на них возникает, как правило, в приложениях, где данные поступают в таких количествах и с такой скоростью, что нет никакой возможности сохранять их, по крайней мере, надолго. С такими задачами анализа данных сталкиваются, например, исследователи в ЦЕРНе, где данные генерируются со скоростью 700 мегабайт в секунду.[3]

3 Вероятностные графические модели

Одним из наиболее впечатляющих результатов использования байесовского формализма для описания задач обработки данных явился аппарат вероятностных графических моделей, в общих чертах разработанный к концу 90ых-началу 00гг [5]. Графические модели позволили радикально пересмотреть области применения методов машинного обучения и анализа данных за счет отказа от требования независимости скрытых переменных для разных объектов. Дискриминативная модель выборки объектов задается совместным распределением $p(T, \vec{w}|X) = p(T|X, \vec{w})p(\vec{w})$, которое, в отличие от классического случая, больше не факторизуется по отдельным объектам.

Прежде чем продолжить дальнейшее изложение приведем несколько примеров, иллюстрирующих, насколько более широкий пласт задач можно решать за счет отказа от предположения о независимости.

- Социальные сети. Пользователи социальных сетей характеризуются, как наблюдаемыми переменными (например, анкетной информацией, которую пользователь сообщил о себе в сети), так и скрытыми переменными (например, его реальными интересами, предрасположенностью к положительной реакции на адресную рекламу и т.п.). Хотя мы можем формально анализировать каждого пользователя независимо, представляется довольно очевидным, что информация о значениях скрытых переменных его друзей, может значительно расширить наши представления о данном пользователе.

- Компьютерное зрение. В задаче семантической сегментации изображений, являющейся первым этапом любой системы компьютерного зрения, требуется сопоставить каждому пикселю некоторую метку класса, соответствующую предмету, в изображение которого входит данный пискель. Очевидно, что помимо информации о данном пикселе (цвет, значения дескрипторов, интенсивность и др.) или других пикселях, важную роль играют метки соседних пикселей, т.к. неявно предполагается, что соседние пиксели чаще всего имеют одинаковые метки.

- Имитационное моделирование. При моделировании сред взаимодействующих агентов (например, транспортных потоков в городах) состояние каждого агента зависит, помимо прочего, от состояний других агентов, находящихся в пределах зоны взаимодействия. Состояние каждого агента можно рассматривать как скрытую переменную обеъекта, зависящую от скрытых переменных других объектов. Исследование таких взаимодействий играет важную роль, т.к. позволяет установить условия скачкообразных переходов от локальных взаимодействий к глобальным (т.н. фазовые переходы), например, когда из-за резкого кратковременного торможения одной машины в потоке возникает многокилометровая пробка.

- Коллаборативная фильтрация (collaborative filtering). С развитием интернет-коммерции все большую актуальность получают рекомендательные сервисы. В ситуации, когда посетитель физически не может просмотреть весь ассортимент интернет-магазина, включающий в себя десятки тысяч наименований, возникает задача формирования ограниченного списка товаров, которые его потенциально могут заинтересовать. Ясно, что кроме наблюдаемых переменных объекта

[3]Автор хотел бы выразить благодарность Сергею Бартунову за помощь при данного раздела.

(клиента), характеризующих его социально-демографический профиль и историю покупок, необходимо анализировать покупки других клиентов и близость их предпочтений к предпочтениям рассматриваемого клиента.

Характерное число объектов в выборке, с которым приходится сталкиваться в современных задачах составляет величину порядка десятков тысяч – миллионов. Основная трудность, возникающая при попытке построить вероятностную модель, содержащую взаимозависимости между скрытыми переменными объектов, заключается в невозможности задать такое распределение в общем виде. В самом деле, пусть имеется тысяча объектов, у каждого из которых есть одна скрытая переменная, принимающая два значения. Для того, чтобы задать $p(T|X,\vec{w})$ нам понадобилось бы задать $2^{1000} \approx 10^{300}$ значений вероятностей. Такое количество на много порядков превосходит объемы доступной памяти любого хранилища данных. При использовании графических моделей предполагается, что совместное распределение может быть представленно в виде произведения т.н. факторов, каждый из которых зависит от небольшого подмножества объектов, причем подмножества пересекаются. Благодаря этому удается смоделировать ситуации, когда скрытая компонента произвольного объекта зависит от скрытой компоненты каждого из оставшихся объектов выборки. С другой стороны, за счет факторизации, можно уменьшить требования к памяти вплоть до линейных по числу объектов, что позволяет хранить совместные распределения на сотни тысяч объектов.

3.1 Условная независимость объектов

Ключевым понятием, необходимым для понимания логики работы аппарата графических моделей, является понятие условной независимости слуайных величин. Случайные величины a и b называются незвисимыми при условии c, если верно[4]

$$p(a,b|c) = p(a|c)p(b|c).$$

Простейшим примером условно независимых величин являются: рост человека (величина a), длина его волос (величина b) и его пол (величина c). Хорошо известно, что рост обратно коррелирует с длиной волос, однако, после добавления в вероятностную модель фактора пола человека, рост и длина волос становятся независимыми величинами.

Напомним, также, два основных правила работы со случайными величинами. Рассмотрим совместную плотность n случайных величин $p(a_1,\ldots,a_n)$. Правило произведения говорит о том, что любую многомерную плотность можно представить в виде произведения одномерных условных плотностей

$$p(a_1,\ldots,a_n) = p(a_n|a_1,\ldots,a_{n-1})\times \\ \times p(a_{n-1}|a_1,\ldots,a_{n-2})\ldots p(a_2|a_1)p(a_1).$$

Аналогичные представления можно выписать для произвольного переупорядочивания переменных.

Правило суммирования позволяет получать безусловные распределения меньшей размерности путем исключения (маргинализации) части переменных

$$p(a_1,\ldots,a_k) = \int p(a_1,\ldots,a_n)da_{k+1}\ldots da_n = \\ = \int p(a_1,\ldots,a_k|a_{k+1},\ldots,a_n)\times \\ \times p(a_{k+1},\ldots,a_n)da_{k+1}\ldots da_n.$$

Все операции, осуществляемые с вероятностными моделями при использовании байесовского формализма, опираются на применение этих двух правил.

3.2 Байесовские сети

Байесовские сети позволяют моделировать причинно-следственные связи между величинами. Для этого на множестве переменных $Y = (X,T,\vec{w})$ нашей вероятностной модели задается ориентированный граф, в котором ребра отражают отношения причинности. По смыслу построения в таком графе запрещены ориентированные циклы. Граф причинности задает систему факторизации совместного распределения

$$p(Y) = \prod_{i=1}^{n} p(y_i|\mathrm{pa}_i),$$

где pa_i — множество родителей i-ой вершины. Заметим, что размер каждого фактора (а именно размерность факторов служит мерой сложности распределения как на этапе его задания, так и на этапе работы с ним) определяется числом родителей вершины. Такая система факторизации значительно упрощает расчеты произвольных условных и маргинальных распределений (а именно к этому, как мы помним, сводятся задачи настройки и вывода в байесовских моделях). Так, используя факторизацию совместного распределения, заданную байесовской сетью

[4]Не ограничивая общности будем полагать величины непрерывными и имеющими плотности. Индексы у функций плотностей будем опускать, считая, что они однозначно идентифицируются своим аргументом.

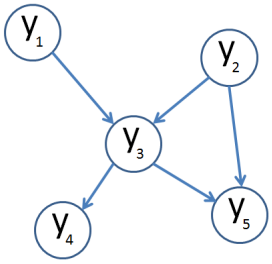

Рис. 2: Пример байесовской сети

на рис. 2 и применяя правила произведения и суммирования, легко получить выражение для, например, такого условного распределения $p(y_5|y_2)$:

$$p(y_5|y_2) = \int p(y_5|y_2, y_3)p(y_3|y_1, y_2)p(y_1)dy_1dy_3.$$

3.3 Марковские сети

Часто возникает необходимость моделировать системы случайных величин между которыми есть зависимости, но некорректно говорить о причинно-следственных связях. Примером таких величин могут быть метки соседних пикселей в задаче сегментации изображений или профили друзей в социальной сети. Для моделирования таких зависимоетй на множестве величин задается неориентированный граф, определяющий факторизацию совместного распределения таким образом

$$p(Y) = \frac{1}{Z}\prod_{c\in\mathcal{C}}\psi_c(Y_c) = \frac{\prod_{c\in\mathcal{C}}\psi_c(Y_c)}{\sum_Y\prod_{c\in\mathcal{C}}\psi_c(Y_c)},$$

где $\psi_c(.)$ — неотрицательные функции, заданные на максимальных кликах графа. Заметим, что в отличие от байесовских сетей, множители (факторы) не имеют вероятностного смысла, поэтому необходима дополнительная нормировка произведения факторов. Легко показать, что если величины y' и y'' никогда не входят в один фактор (т.е. не соединены ребром), то они являются независимыми при условии, что все остальные величины известны. Таким образом, ребра графа определяют отношения условной независимости.

Одним из достоинств систем факторизации, задаваемых графическими моделями, наравне с удобством представления многомерных распределений, является возможность параллельной и распределенной обработки информации при подсчете условных распределений, например, с помощью интерфейса передачи сообщений (message-passing interface).

3.4 Основные задачи, возникающие в графических моделях

Аппарат графических моделей активно используется для точного или приближенного решения следующих основных задач

- Обучение с учителем $\arg\max_{\vec{w}} p(\vec{w}|X^{tr}, T^{tr})$;
- Обучение без учителя $\arg\max_{\vec{w}} p(\vec{w}|X^{tr}) = \arg\max_{\vec{w}} \sum_T p(\vec{w}, T|X^{tr})$;
- Подсчет нормировочной константы Z;
- Подсчет наиболее вероятной конфигурации скрытых переменных $\arg\max_T p(T|X, \vec{w})$
- Подсчет маргинального распределения фиксированной переменной $p(t_i|X, \vec{w})$.

Заметим, что все эти задачи сводятся к подсчету тех или иных условных распределений на неизвестные переменные при условии наблюдаемых переменныхи, быть может, маргинализации по нерелевантным переменным. Можно заметить, что те же задачи возникают в классическом машинном обучении. Перенесение классических результатов на (более сложные) графические модели является одним из важнейших направлений работ в современном машинном обучении.[5]

Список литературы

[1] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[2] D. Blei, A. Ng, M. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003, 3(4-5): 993Ц1022.

[3] G. Brumfiel. "High-energy physics: Down the petabyte highway". Nature 469, 2011, pp. 282-283.

[4] G. Hinton, S. Osindero, Y. Teh. A Fast learning Algorithm for Deep Belief Nets. Neural Computation, 2006, 18(7): 1527-1554.

[5] D. Koller, N. Friedman. Probabilistic Graphical Models. MIT Press, 2009.

[6] D. MacKay. Bayesian Interpolation. Neural Computation, 1992, 4, 415-447.

[7] C. E. Rasmussen. The infnite Gaussian mixture model. In Advances in Neural Information Processing Systems, Vol. 12, 2000

[8] R. Sutton, A. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

[5] Работа выполнена при поддержке гранта РФФИ 12-01-00938.