

БИБЛИОТЕКА КОДА ДЛЯ РАБОТЫ С ВЕРОЯТНОСТНЫМИ СЕТЯМИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ярошевич Ю. А.

Волосевич А. А. – канд. физ.-мат. наук, доцент

Вероятностные сети являются разновидностью вероятностных графовых моделей и представляют причинно-следственные взаимоотношения между переменными в информационном статистическом моделировании [1]. Предлагаемая разновидность сети является модификацией классической байесовой сети, с тем отличием, что таблицы условных распределений $P(X_k | Pa_1(X_k), \dots, Pa_n(X_k))$, ассоциированные с вершинами графа и имеющими размерность $\|X_k\| \cdot \prod_{i=1}^n \|Pa_i(X_k)\|$, заменяются на n таблиц меньшего размера, представляющих условные распределения $P(X_k | Pa_i(X_k))$, ассоциированных с дугами графа и имеющими общую размерность $\|X_k\| \cdot \sum_{i=1}^n \|Pa_i(X_k)\|$. В рассматриваемой модификации вершины графа содержат таблицы безусловного распределения $P(X_k)$. Такая сеть, построенная и обученная по известным данным, позволяет получать уточненные распределения неизвестных случайных величин по неполному набору известных и на основе полученных данных производить суждения в условиях неопределенности.

Библиотека для представления вероятностных сетей разработана на языке программирования F#. Данный язык лаконично сочетает в себе возможности объектно-ориентированного и функционального программирования. Большинство наиболее часто встречаемых программистских задач могут быть записаны на F# проще и короче, чем на других языках для платформы .NET, например C#. Система типов языка позволяет избегать наиболее частых ошибок, например, вызванных нулевыми ссылками и неконтролируемым изменяемым состоянием. Встроенная возможность автоматического вывода типов позволяет в большинстве случаев не производить аннотацию типов, проверка правильности типов производится во время компиляции и является строгой, без возможности неявного приведения типов. Язык хорошо вписывается в экосистему .NET, позволяя как создавать библиотеки удобные для использования из других языков, так и пользоваться обилием готовых библиотек для решения широкого круга задач.

Библиотека содержит набор типов для представления, загрузки и сохранения, управления структурой сети. Основной тип для представления сети – обобщенный класс `BNet<T, U>`. Разработанная структура данных для представления сети может быть одинаково эффективно использована в различных приложениях: вычислительных – когда достаточно знать лишь основные атрибуты случайной величины, такие как размерность; графических – когда для визуализации сети также важны дополнительные параметры, такие как метки состояний и названия переменных. Такая настраиваемая «детализация» представления сети достигается за счет использования многослойного представления сети, где каждый слой содержит какие-то атрибуты сети, но может быть расширен следующим слоем, содержащим дополнительные атрибуты, необходимые, например, для работы специализированного алгоритма или для визуализации сети пользователю. Технически это реализовано с использованием набора обобщенных типов и алгоритмов для работы с ними. В библиотеке объявлены типы для разных уровней сети, например, классы `NodeAttributes<T>` и `LinkAttributes` предназначены для промежуточного уровня и содержат дополнительные атрибуты сети необходимые для проведения статистического вывода в сети. Класс `VarAnnotations` предназначен для представления детальной информации о случайной переменной (вершине сети) и содержит информацию, необходимую для визуализации сети пользователю. Разработанная библиотека придерживается подхода декларативного представления [2], т.е. представление сети отделено от алгоритмов работы с сетью, что позволяет использовать одну библиотеку для решения задач из различных предметных областей.

Функциональность, предоставляемая разработанной библиотекой кода, покрыта большим количеством модульных тестов для проверки корректности работы библиотеки. Модульные тесты написаны с использованием библиотек `xUnit` и `Unquote`, первая предоставляет общую инфраструктуру для выполнения тестов для платформы .NET и адаптирована для работы с языком программирования C#, вторая – позволяет использовать обычный синтаксис языка F# внутри блоков цитирования кода для описания проверяемых утверждений. Большинство типов данных, используемых и объявленных в библиотеке являются неизменяемыми. Частично это обусловлено выбранным языком программирования, в котором поощряется использование неизменяемых данных, частично – желанием уменьшить множество потенциальных ошибок. Использование неизменяемых данных сказалось на производительности библиотеки, она ожидаемо ниже, чем у аналогичной библиотеки, написанной в императивном стиле с изменяемыми данными, это связано в первую очередь с тем, что во многих случаях где изменяемые структуры данных предоставляют операции за константное время, неизменяемые – предоставляют операции работающие за логарифмическое время.