- мониторинг компьютерных сетей;
- извлечение информации (Data Mining);
- защита информации;
- облачные вычисления [2].

Общим для всех этих сфер является использование языка Datalog как более высокого уровня абстракции данных в выполнении запросов и построении дедуктивных баз данных [3].

Список использованных источников:

1. Цуканова, Н.И. Теория и практика логического программирования на языке Visual Prolog 7/ Н.И. Цуканова, Т.А. Дмитриева // Международный журнал экспериментального образования, №2, 2012. – стр. 77.
2. Datalog and Emerging Applications: An Interactive Tutorial / Shan Shan Huang, Todd J. Green, Boon Thau Loo // SIGMOD'11. – Athens, 2011.
3. Катериненко, Р. Дедуктивные базы данных: построение продукционной модели с помощью современных СУБД / Lambert Academic Publishing, 2012.

СИСТЕМА ОЦЕНКИ КАЧЕСТВА АЛГОРИТМОВ КЛАССИФИКАЦИИ

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

Михалькевич Е. Ф.

Лукашевич М. М. − канд. техн. наук

Система предназначена для оценки качества различных алгоритмов классификации. Оценка может производиться с использованием различных методик и метрик оценки, а также с использованием любых входных данных, что позволяет подобрать алгоритм классификации под конкретную задачу.

Классификатор – это абстрактный автомат (математическая модель), принимающий решение об отнесении вектора информативных признаков к одному из классов. Предполагается, что информативные признаки объединяются по некоторому правилу в интегрированный параметр, по которому и принимается решение. Самым тривиальным решением является дуальное решение - ДА/НЕТ.
Для правильной работы классификатора его необходимо обучать и тестировать. Чтобы настроить классификатор под определенную задачу, необходима репрезентативная обучающая последовательность, по которой можно достаточно точно восстановить параметры классификатора. Обычно, для тестирования и обучения классификаторов, базы берутся в различных открытых репозиториях. Самым известным из них является UCI [1].

При решении любой проблемы связанной с машинным обучением в первую очередь производится анализ исходных данных, специфики требований к интеллектуальной системе, а также может строиться модель системы. Это все требуется для выбора методики классификации или кластеризации.

На сегодняшний день существует большое множество различных классификаторов. Каждый из них имеет свои сильные и слабые стороны. Поэтому основной проблемой является выбор классификатора под конкретную задачу или под конкретные данные.

Как известно, классификация не сводится к методам разделения наборов подмножеств в признаковом пространстве. Для заказчика важно не только получить алгоритм, реализующий (возможно с некоторыми ошибками) требуемое разделение классов, но и иметь оценку надежности решения поставленной задачи, т.е. знать, как часто данный алгоритм будет ошибаться при классификации вновь предъявляемых объектов. Ясно, что указанная оценка напрямую определяет качество решения поставленной задачи. На практике же дать такую обоснованную оценку часто оказывается затруднительным. [2]

Процесс обучения классификатора важен не менее чем построение модели. Хотя не существует общих алгоритмов обучения для всех моделей, используются схожие методы для отдельных методов классификации [4].

Предложенная система позволяет решить проблему подбора классификатора. Однако она может использоваться и для других приложений. Например, исследование влияния каждого из признаков на качество классификаторов и уменьшение размерности пространства признаков.

Система оценки качества классификаторов обладает гибкостью конфигурации тестирования алгоритмов. На сегодняшний день существует огромное количество классификаторов. Многие исследователи адаптируют и модифицируют алгоритмы для своего приложения [3]. Из всех классификаторов можно выделить несколько основных и наиболее используемых.

При работе с системой можно выбрать:

- Количество признаков образов;
- Несколько алгоритмов классификации для сравнения;
- Алгоритмы обучения для каждого алгоритма классификации;
- Критерии остановки обучения;

- Метрики для оценки качества.

Одним из ключевых моментов является возможность добавления собственных алгоритмов классификации и обучения. Это достигается с помощью обобщения алгоритмов в группы с одинаковыми интерфейсами и возможностями языка Python.

После получения результатов тестирования их можно сравнить между собой, сохранить для последующего использования, а также экспортировать для печати.

Таким образом, была разработана система оценки качества алгоритмов классификации. Она имеет возможность выбирать различные алгоритмы классификации и разные метрики для оценки. Задачей дальнейших исследований является реализация функционала для алгоритмов кластеризации.

Список использованных источников:

1. http://archive.ics.uci.edu/ml/ - Электронный ресурс.
2. Гуров, С.И. Как оценить надежность алгоритма классификации // Таврический вестник информатики и математики №1. – Симферополь, 2002.
3. Ramdass, D. Document Classification for Newspaper Articles / D. Ramdass, S. Seshasai // 6.863 Final Project – MIT, Spring 2009.
4. Michie, D. Machine Learning, Neural and Statistical Classification / D. Michie, D.J. Spiegelhalter, C.C. Taylor // Overseas Press, 2009 edition (August 28, 2009) – 290 p.

АЛГОРИТМ ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ТОПОЛОГИИ МИКРОСХЕМ НА МИКРОСКОПНЫХ ИЗОБРАЖЕНИЯХ СЛОЕВ ТОПОЛОГИИ СБИС

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

Питкин А.Б.

Воронов А. А. – канд. техн. наук, доцент

В процессе производства сверхбольших интегральных микросхем (СБИС), возможно появление дефектов на топологических слоях. Такие дефекты возникают например в процессе переноса изображения или в процессе фотолитографии. Автоматический контроль качества позволит в кратчайшие сроки выявлять различные дефекты. Идея заключается в поиске заданных шаблонов известных дефектов, либо шаблонов с изображением эталонов элементов на значительно большем изображении представляющем собой снимок слоя готовой микросхемы.

Простейшим способом детектирования на изображении является поиск по шаблону(template matching) [2]. Предварительно изображение обрабатывается с использованием фильтра дистантного преобразования [3], что хорошо выделяет контуры объектов на изображении. Далее происходит постепенный параллельный перенос изображения шаблона в окне над исходным изображением слоя и вычисление корреляции между ними по одной из формул [4]. На основании полученного значения корреляции изображений делается вывод о совпадении шаблона с фрагментом под ним.

Однако элементы на слое интегральной микросхемы могут располагаться с вращением кратным 90° в плоскости изображения, что увеличивает число проходов по изображению в четыре раза. Также изображение элемента может незначительно отличаться от эталонного изображения в связи с особенностями технологического процесса производства микросхемы.

Целью алгоритма является задача разработать метод поиска изображения устойчивый к поворотам изображения и масштабированию и позволяющий с высокой степенью вероятности найти все повторения заданного шаблона на изображении.

Универсальными методами поиска изображений являются методы, основанные на выделении ключевых точек на изображении. Поиск на изображении определенных характерных примитивов позволит перейти от описания изображения в виде массива цветных пикселов к массиву параметров описывающих эти примитивы. Таким образом, сокращается количество параметров, по которым необходимо производить дальнейшее сравнение изображение. Поиск на изображении при использовании методов основанных на ключевых точках состоит из трех этапов:

1. Этап детектировании ключевых точек. На данном этапе происходит сканирование изображения и нахождение на нем характерных точек интереса.
2. Этап создания дескрипторов для ключевых точек, найденных на предыдущем шаге. Таким образом, на данном этапе формируется математическое описание найденных ключевых точек в виде вектора значений.
3. Сопоставление наборов значений дескрипторов полученных в процессе обработки двух изображений и нахождение близких друг к другу значений.

Наиболее эффективными типами дескрипторов в данный момент являются дескрипторы SIFT (SIFT - Scale Invariant Feature Transform) [5] и SURF (Speeded Up Robust Features) являющийся улучшением