Теория метода опорных векторов (SVM).

**1. Основная идея: Линейное разделение (Напоминание)**

* Лабораторная начинается с напоминания о *линейных моделях* для классификации. Эти модели пытаются найти прямую линию (в 2D) или плоскую гиперплоскость (в более высоких измерениях) для разделения разных классов.
* *Линейный SVM* — один из таких методов. Он ищет гиперплоскость, которая имеет *максимальный зазор* (margin) — наибольшее возможное расстояние между собой и ближайшими точками каждого класса. Эти ближайшие точки критически важны и называются **опорными векторами** (support vectors).

**2. Проблема: Когда данные не являются линейно разделимыми**

* В работе показан пример с использованием make\_blobs (Рис. 1), где невозможно провести одну прямую линию, чтобы идеально разделить синие круги и зеленые треугольники.
* Простая линейная модель (например, LinearSVC) не справится и построит границу, которая неправильно классифицирует много точек (Рис. 2).

**3. Решение 1: Инженерия признаков (Создание нелинейности вручную)**

* Один из способов справиться с нелинейно разделимыми данными — *добавить новые признаки*, вычисленные на основе исходных. Идея в том, чтобы преобразовать данные в пространство более высокой размерности, где они *могут* стать линейно разделимыми.
* **Пример:** Лабораторная берет исходные 2D данные (признак 0, признак 1) и добавляет *третий* признак: признак 1 \*\* 2 (квадрат второго признака).
* При отображении в 3D (Рис. 3) точки данных перераспределяются. Теперь становится возможным найти *плоскость* (эквивалент линии в 3D), которая разделяет два класса (Рис. 4).
* **Ключевой момент:** Когда эта разделяющая плоскость из 3D-пространства проецируется обратно в исходное 2D-пространство, граница решения перестает быть прямой линией. Она становится кривой (в данном случае эллипсом, Рис. 5).
* **Ограничение:** Придумывать вручную, какие новые признаки добавить (квадраты, кубы, взаимодействия вроде признак 0 \* признак 1 и т. д.), может быть сложно. Добавление множества признаков становится очень затратным вычислительно, особенно если исходных признаков много.

**4. Решение 2: Ядерный трюк (Неявная обработка нелинейности)**

* Ядерный трюк — это способ **получить результат, как будто мы перешли в это сложное многомерное пространство, но без реального перехода и вычислений в нем!**
* Оказывается, чтобы найти положение той самой *плоской гиперплоскости* в новом пространстве, SVM не обязательно знать точные новые координаты точек. Ему достаточно знать, **насколько точки похожи (или "близки") друг к другу *в этом новом пространстве***.
* **Ядерная функция — это как "измеритель сходства"**. Ты даешь ему две *исходные* точки, а оно вычисляет, насколько они были бы "близки" или "похожи" друг на друга *в том самом сложном, невидимом пространстве*. Оно делает это напрямую, минуя этап вычисления новых координат.
* **Распространенные ядра:**
  + **Полиномиальное ядро:** Вычисляет сходство, соответствующее полиномиальным комбинациям исходных признаков.
  + **Ядро RBF (Радиальная Базисная Функция / Гауссово ядро):** Наиболее часто используемое, на нем сосредоточена вторая часть лабораторной. Оно измеряет сходство на основе расстояния между точками: чем ближе точки, тем они более схожи. Эффективно оно отображает данные в *бесконечномерное* пространство.

**5. Как работает Ядерный SVM (особенно с ядром RBF):**

* **Опорные векторы:** Как и линейный SVM, ядерный SVM определяет критические точки данных, лежащие на границе классов — опорные векторы.
* **Принятие решения:** Для классификации *новой* точки SVM вычисляет ее "сходство" (используя ядерную функцию) с каждым из опорных векторов. Окончательное решение основывается на этих значениях сходства и "важности", присвоенной каждому опорному вектору во время обучения (хранится в атрибуте dual\_coef\_).
* **Формула ядра RBF:** k\_rbf(x1, x2) = exp(-gamma \* ||x1 - x2||^2)
  + ||x1 - x2||^2: Квадрат евклидова расстояния между точками x1 и x2.
  + gamma: Параметр, который нужно настраивать. Он контролирует, насколько далеко распространяется влияние одного обучающего примера.

**6. Ключевые параметры SVM с ядром RBF:**

* **gamma:** Контролирует "ширину" гауссова ядра.
  + **Низкий gamma:** Означает большой радиус/ширину. Точки на большем расстоянии считаются схожими. Это приводит к более гладкой, простой границе решения (более общая модель, возможно недообучение). (См. Рис. 7, левый столбец).
  + **Высокий gamma:** Означает малый радиус/ширину. Только очень близкие точки считаются схожими. Граница становится сложнее, плотнее подгоняется к данным, потенциально слишком сильно фокусируясь на отдельных точках (более сложная модель, возможно переобучение). (См. Рис. 7, правый столбец).
* **C:** Параметр регуляризации (та же роль, что и в линейных моделях). Он задает компромисс между гладкой/простой границей (широкий зазор) и правильной классификацией точек обучающей выборки.
  + **Низкий C:** Создает более простую, гладкую границу, допуская некоторые ошибки классификации (более терпим к ошибкам, простая модель). (См. Рис. 7, верхняя строка).
  + **Высокий C:** Пытается классифицировать *каждую* обучающую точку правильно, приводя к более сложной, извилистой границе, которая очень точно следует обучающим данным (менее терпим к ошибкам, сложная модель, возможно переобучение). (См. Рис. 7, нижняя строка).
* **Взаимодействие:** C и gamma взаимодействуют. Поиск наилучшей модели часто требует настройки обоих параметров вместе.

**7. Критическая роль предобработки данных (Масштабирование):**

* SVM (особенно с ядрами вроде RBF, которые опираются на расстояния) *очень чувствительны* к масштабу признаков.
* Если один признак имеет значения от 0 до 10000, а другой — от 0 до 1, то вычисление расстояния (||x1 - x2||^2) будет полностью доминироваться первым признаком. Второй признак почти не будет иметь влияния.
* **Пример:** Набор данных Breast Cancer (Рис. 8 показывает, что признаки имеют сильно различающиеся диапазоны). Без масштабирования SVM работает плохо (Точность 0.63 на тесте).
* **Решение:** *Масштабировать* данные перед обучением SVM. Распространенный метод (использованный в лабораторной "вручную" и доступный как MinMaxScaler в scikit-learn) — привести значения каждого признака к диапазону от 0 до 1.
* **Результат:** После масштабирования производительность SVM на данных Breast Cancer значительно улучшается (Точность ~0.95-0.97).

**8. Итог: Плюсы и Минусы**

* **Плюсы:**
  + Мощный: Может создавать сложные границы решений.
  + Эффективен в пространствах высокой размерности (даже когда число признаков > числа примеров).
  + Хорошо работает на малых и средних наборах данных с четким разделением классов.
* **Минусы:**
  + Чувствителен к выбору параметров (C, gamma) и типу ядра.
  + *Очень* чувствителен к масштабированию данных. Требует тщательной предобработки.
  + Вычислительно дорог и требователен к памяти на очень больших наборах данных (плохо масштабируется).
  + Менее интерпретируем ("черный ящик") по сравнению с моделями вроде деревьев решений – сложнее объяснить, *почему* было сделано конкретное предсказание.

По сути, лабораторная работа показывает, как стандартные линейные модели терпят неудачу на сложных данных, как ручное добавление признаков может помочь, но неудобно, и как Ядерный SVM предоставляет элегантный и мощный способ справиться с нелинейностью путем неявного отображения данных в высшие размерности через ядерные функции, с оговоркой, что для хорошей производительности необходимы тщательная настройка параметров и масштабирование данных.