1 Постановка задачи

Задано множество объектов сложной структуры $\mathcal{X}$, класс функций распознавания $\mathcal{F}$. Требуется каждому объекту $x \in \mathcal{X}$ поставить в соответствие оптимальную функцию распознавания $f^* \in \mathcal{F}$, минимизирующую *неизвестную* функцию ошибки $S(x, f)$:

$$f^* = \arg\min_{f \in \mathcal{F}} S(x, f). \tag{1}$$

Задана выборка $D = \{(x_i, f_i, S(x_i, f_i))\}_{i=1}^m$, состоящая из объектов x_i, функций распознавания f_i и значений функции ошибки $S(x_i, f_i)$. Предполагается, что каждая из функций f_i минимизирует функцию ошибки на объектах выборки:

$$f_i = \arg\min_{f \in \mathcal{F}} S(x_i, f).$$

Обозначим за L^* суммарную ошибку на элементах выборки:

$$L^* = \sum_{i=1}^{m} S(x_i, f_i).$$

Обозначим за $a : \mathcal{X} \to \mathcal{F}$ *правило обучения*, отображающее множество объектов X во множество функций распознавания $\mathcal{F}$. Обозначим за $L(a)$ функцию суммарной ошибки правила обучения a на объектах выборки x_i:

$$L(a) = \sum_{i=1}^{m} S(x_i, a(x)).$$

Переформулируем задачу (1) следующим образом: требуется найти правило обучения $a : \mathcal{X} \to \mathcal{F}$, минимизирующее разность

$$L(a) - L^* \quad \to \quad \min(a). \tag{2}$$

2 Структурное обучение

Для перехода от задачи (2) к структурному обучению, введем следующее предположение о связи функции ошибки и функций распознавания. Будем считать, что для всех объектов $x \in \mathcal{X}$ и всех функций $f_i, f_j \in \mathcal{F}$ выполнено неравенство

$$|S(x, f_i) - S(x.f_j)| \leq c\Delta(f_i, f_j), \tag{3}$$

где c — константа, а $\Delta(f_i, f_j)$ — структурное расстояние между функциями f_i и f_j, которое будет определено в дальнейшем. В случае выполнения неравенства (3) автоматически выполняется ограничение сверху целевой функции разности (2):

$$L(a) - L^* = \sum_{i=1}^{m} |S(x_i, a(x_i)) - S(x_i, f_i)| \leq c \sum_{i=1}^{m} \Delta(a(x_i), f_i).$$

Таким образом, при использовании предположения (3) задача поиска оптимального правила обучения (3) сводится к задаче *структурного обучения*:

$$\sum_{i=1}^{m} \Delta(a(x_i), f_i) \quad \to \quad \min(a). \tag{4}$$

Структура на множестве суперпозиций. В этом разделе зададим структуру на множестве суперпозиций для определения вида правила обучения a и расстояния между функциями Δ. Будем рассматривать функцию распознавания f в виде композиции базисных функций $f = h_0 \circ h_{k_1} \circ ... \circ h_{k_y}$, выбранных с возвращением из множества базисных функций $\mathcal{H} = \{h_0, h_1, ..., h_r\}$. Каждой базисной функции h_k соответствует арность $a_k \geq 0$. Крайним левым элементом суперпозиции f является функция $h_0 : h_0(x) = x$, имеющая арность $a_0 = 1$. Кроме того, предполагается, что существует непустое подмножество $\mathcal{H}_0 \subset \mathcal{H}$ функций нулевой арности, называемых свободными переменными.

Поставим в соответствие целевой переменной f крашеное дерево $\Gamma = (V, E)$. На множестве вершин задана функция раскраски $h : V \to \mathcal{H}$. Цвет $h(v)$ вершины $h(v)$ является базисной функцией и определяет количество дочерних вершин у v: оно совпадает с арностью функции $h(v)$. Таким образом, допустимыми деревьями являются деревья, имеющие свои корнем вершину $v_0 : h(v_0) = h_0$, в листьях содержащие функции нулевой арности, а количество дочерних вершин для всех остальных элементов совпадает с соответствующими арностями.

Определим расстояние между функциями f и $\hat{f}$ как расстояние между деревьями Γ_f и $\Gamma_{\hat{f}}$, задающими суперпозиции f и $\hat{f}$. Другими словами, определим расстояние $\Delta(f, \hat{f})$ в виде количества несовпадающих элементов бинарных векторов $\mathbf{f}$ и $\hat{\mathbf{f}}$:

$$\Delta(f, \hat{f}) = \sum_{i,j} |f_{ij} - \hat{f}_{ij}|, \tag{5}$$

где элемент вектора $\mathbf{f}$, индексируемый f_{ij}, равен 1 в случае, если последовательность цветов (h_i, h_j) принадлежит множеству ребер E_f дерева Γ_f, и равен 0 в противном случае.

Таким образом, согласно формулам (4) и (5) для произвольного объекта x будем искать функцию распознавания $\hat{f} = a(x)$ в виде

$$\sum_{i,j} |f_{ij} - a_{ij}(x)| \quad \to \quad \min(a), \tag{6}$$

где a_{ij} является элементом вектора $\mathbf{a}$ и равен 1 в случае, если последовательность цветов (h_i, h_j) принадлежит множеству ребер $E_{a(x)}$ дерева $\Gamma_{a(x)}$, и равен 0 в противном случае.

Отметим, что элементы оптимизируемого вектора $\mathbf{a}$ должны быть бинарными и удовлетворять условию корректности дерева $\Gamma_{a(x)}$. Для решения задачи (6) предложен следующим аппроксимационный алгоритм.

1. Оценка параметров $\theta_{ij} \in [0, 1]$ с использованием выборки $(x_i, \Gamma_i)_{i=1}^m$ по правилу
$$\hat{\theta}_{ij} = \arg\min_{\theta_{ij}} \|f_{ij} - \theta_{ij}(x)\|.$$

2. Поиск оптимального вектора $\mathbf{a}$, максимизирующего дискриминантную функцию
$$a(x) = \arg\max_{f \in \mathcal{F}} \prod_{(i,j) \in E_f} \hat{\theta}(h(v_i), h(v_j)).$$

Для решения первой задачи предлагается стандартный метод многоклассовой классификации.

Для решения второй задачи предлагается алгоритм на основе динамического программирования. Утверждается, что алгоритм находит оптимальное решение за $O(|\mathcal{H}|^3)$ вычислений.

Алгоритм максимизации дискриминантной функции. Алгоритм основывается на принципе динамического программирования. На шаге k алгоритм хранит массив из $|\mathcal{H}|$ элементов, элемент i которого содержит стоимость оптимального дерева Γ_i^k с корнем в вершине, раскрашенной h_i, с количеством вершин не более k.

На шаге $k + 1$ для каждой базисной функции $h_{i'}$ выполняется процедура присоединения корня $(h_{i'}, \Gamma_i^k)$ для всех деревьев $\Gamma_i^k, i = 1, ..., |\mathcal{H}|$.

Если максимальная стоимость построенных деревьев превосходит стоимость дерева $\Gamma_{i'}^k$, то происходит замена оптимального дерева $\Gamma_{i'}^k$ на дерево $\Gamma_{i'}^{k+1} \equiv (h_{i'}, \Gamma_i^k)$ с максимальной стоимостью. Иначе дерево остается прежним, $\Gamma_{i'}^{k+1} \equiv \Gamma_i^k$.

Если после итерации $k+1$ не произошло ни одной замены дерева, то алгоритм останавливается, а его решением является наилучшее из деревьев с присоединенным корнем (h_0, Γ_i^k). Утверждается, что количество шагов k на превосходит количество базисных функций $k \leq O(|\mathcal{H}|)$. На каждом шаге алгоритм выполняет $O(|\mathcal{H}|^2)$ проверок.