

Обучение модельной MRF при помощи S-SVM

Ромов Петр

9 марта 2012 г.

1 Модель

Рассматривается модель для сегментации изображения. Сегментация изображения — задача структурного предсказания [2], в общем виде ее можно сформулировать:

$$y^* = f(x) := \arg \max g(x, y; w), \quad (1)$$

где $x \in \mathcal{X}$ — изображение, $y \in \mathcal{Y}$ — сегментация, f — решающая функция, введенная через некоторую параметризованную модельную функцию совместности g (*compatibility function*). В качестве модели совместности далее будем использовать параметризованную MRF, а функцию совместности g определим как отрицательное значение энергии марковской сети.

Введем MRF на графе $G = (V, \mathcal{E})$, вершины соответствуют пикселям изображения, а множество ребер \mathcal{E} представляет собой некоторую систему соседства между пикселями. Каждая наблюдаемая переменная сети представляет собой вектор $\mathbf{x}_i \in \mathbf{R}^D$, скрытые переменные — номера классов $y_i \in \{1, \dots, K\}$ к которым относятся соответствующие пиксели. Для удобства, вместе с обозначением y_i будем использовать индикаторную нотацию: $y_{ik} = [y_i = k]$, $\mathbf{y}_i = (y_{i1}, \dots, y_{iK})^\top$. Осталось ввести потенциалы марковской сети.

Введем параметризованные унарные потенциалы:

$$U_i(\mathbf{y}_i, \mathbf{x}_i; W^u) = \mathbf{y}_i^\top W^u \mathbf{x}_i, \quad (2)$$

где $W^u \in \mathbf{R}^{K \times D}$ — матрица-параметр, преобразующая вектор признаков в “штрафы за метки”.

Введем параметризованные парные потенциалы как взвешенную сумму B моделей Поттса:

$$V_{ij}(y_i, y_j; \mathbf{w}^b) = \sum_{t=1}^B w_t^b \cdot P_{ij}^{(t)}[y_i \neq y_j], \quad (3)$$

здесь параметром является вектор весов \mathbf{w}^b .

Теперь можем записать функцию совместности:

$$g(x, y; w) = -E(x, y; w) = - \sum_{i \in V} \sum_{k=1}^K y_{ik} \cdot \mathbf{w}_k^\top \mathbf{x}_i - \sum_{(i,j) \in \mathcal{E}} \sum_{p,q=1}^K y_{ip} y_{jq} \left(\sum_{t=1}^T w_t^b \cdot P_{ij}^{(t)} \right) [p \neq q], \quad (4)$$

здесь \mathbf{w}_k^b — k -я строчка матрицы W^u , собирая воедино все параметры обозначим их $w = \{w_{kd}^u, w_t^b\}$

В таком виде функция g линейна по всем параметрам, а значит для обучения параметров можно использовать структурный метод опорных векторов (S-SVM) [2]. Запишем функцию g как скалярное произведение вектора параметров и вектора особенностей:

$$g(x, y; w) = \langle w, \psi(x, y) \rangle, \quad \psi = (\psi_1^u, \dots, \psi_K^u, \psi_1^b, \dots, \psi_T^b)^\top \quad (5)$$

$$\begin{aligned} \psi_k^u(x, y) &= - \sum_{i \in V} y_{ik} \mathbf{x}_i, \\ \psi_t^b(x, y) &= - \sum_{(i,j) \in \mathcal{E}} \sum_{p,q=1}^K y_{ip} y_{jq} \cdot P_{ij}^{(t)} [p \neq q]. \end{aligned}$$

Для обучение структурным методом опорных векторов необходимо ввести функцию потерь:

$$\Delta(y, \hat{y}) = \frac{1}{|V|} \sum_{i \in V} [y_i \neq \hat{y}_i] = \frac{1}{|V|} \sum_{i \in V} \sum_{p=1}^K y_{ip} [\hat{y}_i \neq p]. \quad (6)$$

2 Эксперимент

Детальное описание проводимого эксперимента:

- \mathcal{X} : изображения 100×100 с тремя цветовыми каналами ($D = 3$);
- \mathcal{Y} : разметки пикселей изображения из \mathcal{X} на $K = 3$ класса;
- MRF имеет четырехугольную систему соседства (*grid*);
- задача вычисления $f(x)$ является NP-трудной (т.к. $K = 3$), ее вычисление производится приближенно методом TRW-S [1];
- **генерация выборки**: “загадывается” вектор параметров w , значения всех каналов всех пикселей x_{id} генерируются независимо из равномерного распределения $U[0, 1]$, в качестве правильного ответа берется $f(x; w)$ (как вариант можно было генерировать правильный ответ как случайную конфигурацию марковской сети, используя методы семплирования);
- в обучении участвует 20 объектов;
- в качестве функции потерь $\Delta(y, \hat{y})$ берется число неверно размеченных пикселей;
- в тестировании участвуют 30 объектов, ошибка на тестовой выборке — доля неверно размеченных пикселей во всех изображениях.

Обучение параметров проводилось структурным методом опорных векторов в формулировке с ослабляющими переменными (*slack variables*):

$$\frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{n=1}^N \xi_n \rightarrow \min_{w, \xi}$$

$$g(x^n, y^n; w) - g(x^n, y, w) \geq \Delta(y^n, y) - \xi_n, \quad \forall y \in \mathcal{Y}$$

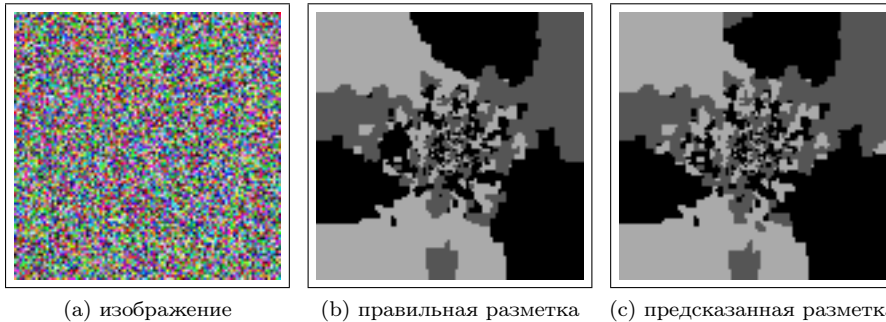


Рис. 1: Визуальная иллюстрация модельного примера и результата работы S-SVM: (a) случайное изображение из тестовой выборки; (b) разметка, полученная по “загаданной” модели; (c) разметка, полученная по модели, обученной при помощи S-SVM.

3 Результаты

Регуляризация. Из Рис. 2 видно, что переобучения не происходит, в регуляризации нет необходимости.

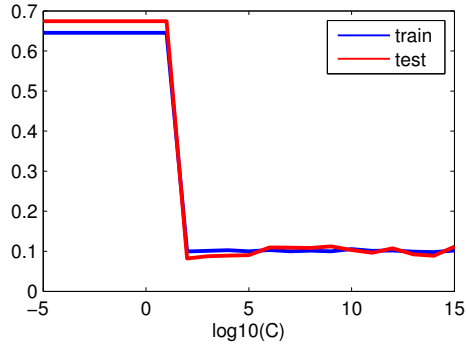


Рис. 2: Зависимость ошибки (среднее значение функции потерь Δ) от коэффициента регуляризации на тренировочной и тестовой выборках.

Время обучения Запуск производился на машине Intel Core i7 (4 ядра), параллельно в отдельном потоке обучалось 4 модели. Обучение проделывалось методом N-Slack S-SVM при помощи алгоритма Cutting-Plane.

Параметров	Число итераций	Время обучения (с)
11	40	831
11	57	1054
11	45	928
11	42	872

Обучение при помощи SVM^{struct} работало очень долго, в этом неплохо было бы разобраться.

4 Реализация

Эксперимент реализован в MATLAB, код прокомментирован, краткое описание основных файлов с кодом:

experiment_toy.m сценарий с общей схемой проделанного эксперимента, используется для запуска эксперимента и получения результатов;

model_generate.m функция для создания модели, под моделью подразумевается характеристики задачи в целом (размер изображения, модели Поттса, и др.) и “загаданные” параметры;

model_inference.m функция находит разметку по изображению, имея обученную модель (реализация решающей функции $f(x; w)$);

model_learn_svm_struct.m функция обучения параметров модели при помощи SVM^{struct} ;

model_learn_under_nslack_ssvm.m функция обучения параметров модели методом N-Slack S-SVM, Cutting-Plane алгоритмом.

Список литературы

- [1] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1568–1583, 2006.
- [2] Sebastian Nowozin and Christoph H Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6:185–365, 2011.