

Графмодели ДЗ № 2
Бузун Назар, postrealist@gmail.com

1. Пояснения к методу сегментации

Формула для ошибки, усредненной по классам, в терминах суперпикселей: потеря на одном суперпикселе (loss) равна

- a) При $\hat{y}_i = y_i$: $1/2 * \text{количество левых пикселей в суперпикселе} / \text{количество пикселей в изображении того же гласа что и левые пиксели};$
- b) При $\hat{y}_i \neq y_i$: $1/2 * \text{количество доминирующих (по классу) пикселей в суперпикселе} / \text{количество пикселей в изображении того же гласа что и доминирующие пиксели}.$

Решение задачи минимизации энергии, дополненной функцией потерь, при помощи алгоритма построения разреза графа:

$$E_i(y_i) = w_{y_i}^T x_i - \text{loss}(y_i)$$

$$E_{ij}(y_i, y_j) = w_{y_i y_j}^T e_{ij},$$

где x_i – признаки вершины, e_{ij} – признаки ребра.

$$\begin{aligned}[i - \text{src}]_w &= E_i(0) + E_{i < j}(0, 0) \\ [i - \text{sink}]_w &= E_i(1) + E_{i < j}(1, 0) + E_{k < i}(1, 1) - E_{k < i}(1, 0) \\ [j - i]_w &= E_{i < j}(1, 0) + E_{i < j}(0, 1) - E_{i < j}(1, 1) - E_{i < j}(0, 0) > 0\end{aligned}$$

SVM struct оптимизация:

использовалась java библиотека JLIS с параметрами: MAX_SVM_ITER = 1000, DUAL_GAP = 0.07, check_inference_opt = false, C = 0.04, L2 norm.

JLIS имеет преимущества в сравнении с C++ реализацией – параллельное решение задач минимизации энергии, удобная интеграция с другими частями алгоритма, проверка корректности задачи минимизации. К недостаткам можно отнести плохие критерии остановки, часто приводящие к застыванию, отсутствие документации, неразборчивый код.

Унарные признаки

$$f_i = (\vec{x} \cdot [y_i = 0], \vec{x} \cdot [y_i = 1])$$

Парные признаки

$$f_{ij} = (\vec{c}_1 \cdot [y_i = 0, y_j = 0], \vec{c}_2 \cdot [y_i = 1, y_j = 1], \vec{c}_3 \cdot [y_i \neq y_j])$$

В качестве компонент векторов $\vec{c}_1, \vec{c}_2, \vec{c}_3$ можно брать следующие величины:

- 1.0

- длина границы
- $I(\text{сила границы} > c)$
- $\max(\text{сила границы}, c)$
- $\cos(x_i, x_j)$
- $\exp(\gamma \|x_i - x_j\|^2)$
- \vec{e}_c – базисный вектор пары категорий соседних суперпикселей. Данный признак позволяет учесть какие объекты с сильно отличающимися признаками могут сочетаться в одном классе (например красные фары часто сочетаются и черным цветом остальных частей автомобиля). Для задания данной части парного признака было создано 10 категорий из унарных признаков при помощи метода weka.SimpleKMeans. Аналогичную кластеризацию можно было провести по усредненным RGB цветам суперпикселей.

2. Тестирование

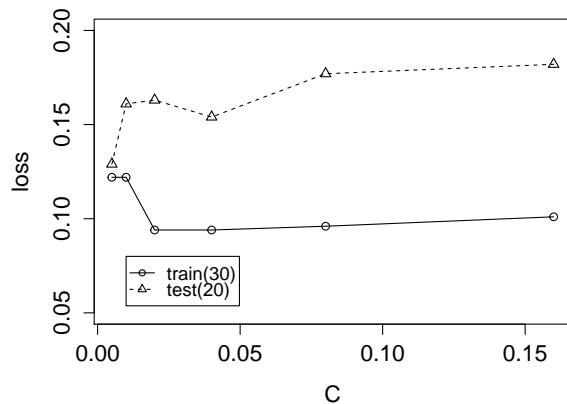


Рис. 1: Подбор параметра регуляризации по разбиению выборки на 2 блока.

Для итогового обучения были выбраны изображения с 1 по 100, остальные для проверки качества. Время работы: 1536 сек, средний loss на обучающей выборке 0.117, средний loss на тестовой выборке **0.123**.

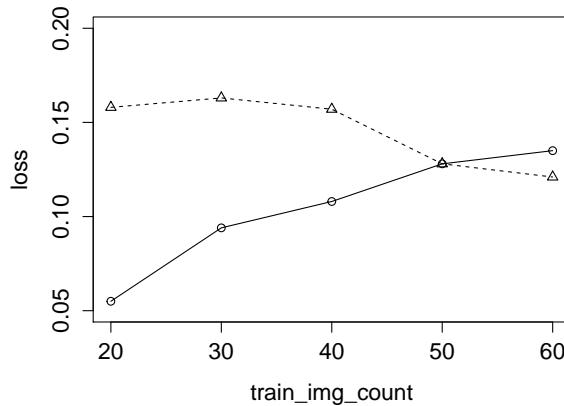


Рис. 2: Переобучение в зависимости от размера обучающей выборки.

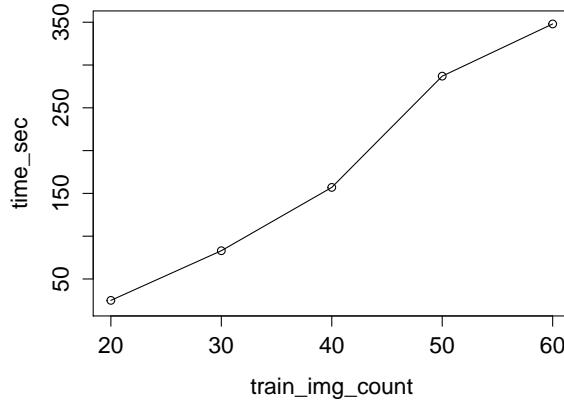


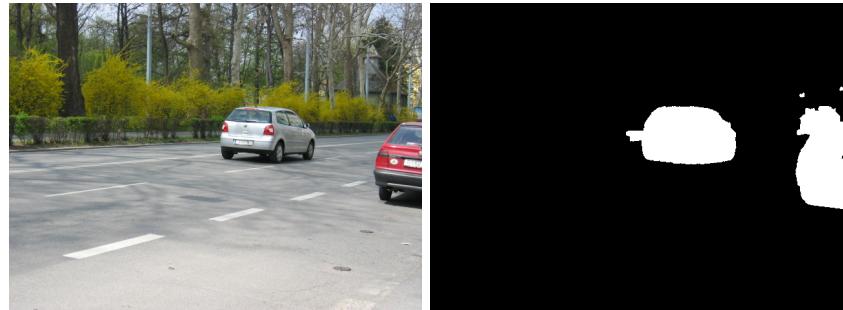
Рис. 3: Время работы в зависимости от размера обучающей выборки.

3. Резюме

Минимизации энергии при помощи алгоритма построения разреза графа кажется не самым удачным способом, так как очень часто удаление несубмодулярных ребер вносит значительное искажение в величину разреза по сравнению с энергией модели. Модно либо дополнить SVM struct дополнительным условием, либо использовать MF Var Inference к примеру.

При изменении параметра С и парных признаков оптимизация часто перестает сходится. Но при сильно заниженных С сходимость всегда есть.

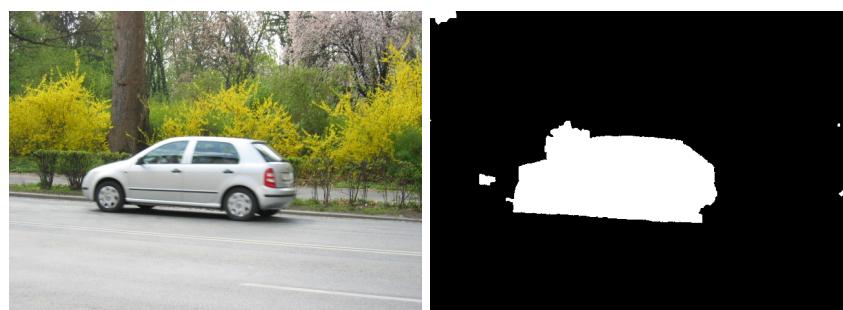
По итогам сегментации: плохо выделяются машины малого размера, часто к машинам прилепляются куски фона, граница машины перестает быть гладкой и овальной, появляется много объектов ошибочно принятых



loss=0.036



loss=0.021



loss=0.036



loss=0.031



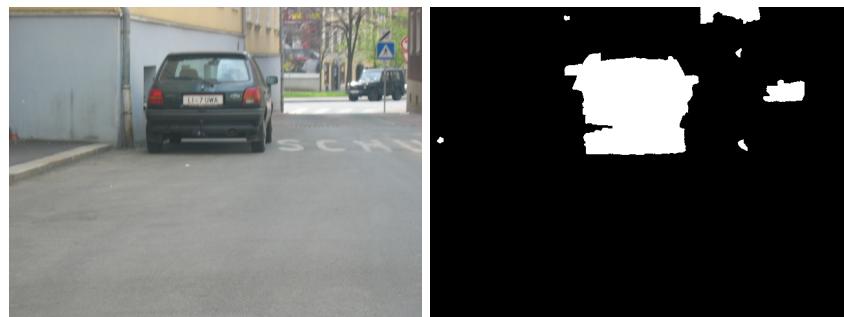
loss=0.198



loss=0.119



loss=0.139



loss=0.152

за машину.



loss=0.234