Structured Completion Predictors Applied to Image Segmentation

Dmitriy Brezhnev, Raphael-Joel Lim, Anirudh Venkatesh

December 16, 2011

Abstract

Multi-image segmentation makes use of global and local features in an attempt to classify every pixel in an image into a semantic region. One important relationship is inter-class spacial interaction between small local regions we call "superpixels". While complex models have been built in order to provide for such semantic understanding and define visual grammars, we explore the usefulness of a new technique we name stacking.

This method performs multi-class SVM learning several times by first constructing label probabilities for each superpixel based on extremely local superpixel features, such as raw pixel information, and increases semantic understanding through stacked predictors that augment the feature set with predictions on the surrounding context.

The results of the algorithm with our constructed features are marginally successful, demonstrating that the algorithm can be used to improve semantic understanding. This report analyzes the success of constructed features and analyzes reasons for a lack of strong improvement.

1 Introduction

The goal of multi-image segmentation is to classify every pixel into a semantic region, unlike single-object recognition algorithms that aim to find a particular object. Innovative works in the recent years have applied several strategies for this goal [4]: Conditional Markov Fields, that encode probabilistic preferencs [5]; close-to-medium range effects to impose consistent layout recognized to be part of a known object [6]; among others.

In our paper, we base our algorithm on a strategy presented by Gould et. al [1]. There, Gould et al. proposes to decompose the images into small consistent regions called superpixels, for which around 1500 local features are computed, which include raw RGB values and averages of surrounding pixel information. Subsequently, the authors use this superpixel decomposition to build a complex model to learn the local and global environments. However, due to the complexity of the model, the inference becomes an expensive operation.

Another approach to this problem is to use lower-order models to obtain more features from context probabilities. We use a multi-class SVM to get a probability distribution for surrounding pixel's labelings and we capture this 'environment' classification to design features. This provides semantic understanding without drastically increasing the complexity of the model. In the case of image segmentation, images of natural scenes tend to have a certain structure that we can leverage to make good predictions. For example, the regions of images containing trees, buildings, or mountains are more likely to be backed by 'sky.' We examine this approach in its application to computer vision, analyze possible features and provide analysis for how the algorithm can be improved.

2 Approach

2.1 Theoretical Approach

Constructing features from predictions (stacked learning) goes back to [7], [8]. The essence of the approach is to infer context from a baseline model M_0 to construct new features for models M_1, $M_2 \ldots$, with later models having better data separability and increased accuracy.

The problem of image segmentation can be formulated as prediction of $y \in \{1, \ldots, 8\}$, corresponding to the region label, given a feature vector x for a certain superpixel. To represent our algorithm's steps, consider three models. M_0 represents the model with the region predicted solely from pixel data. Based on model M_0, two models are constructed M_1^* and M_1. Model M_1^* appends context features to represent the semantic local data, such as the superpixel above being region 1. Note that information about the surroundings is encompassed in these additional features and not aboout the superpixel itself. M_1^* is not realistic in practice (as we don't have the correct labelings); we instead use the probability distributions for each region. Hence the predictions of M_o will be encoded in the

distributions $p^{(i)} := p(y|x^{(i)})$. See the figure for a graphical representation of the algorithm. After computing M_1^* and M_1, the algorithm uses the former as training and the latter for testing to produce a 'stacked' predictor. As we will see, this has marginally positive improvements.

2.2 Technical Approach

We first trained a baseline multiclass SVM (M_0) [3] on a random selection of 10 percent of the data set. Then for each feature vector in our dataset, we obtained from M_0 eight scores indicating how likely a given superpixel belonged to some class. These scores were converted into probabilities by performing logistic regression on the set of scores corresponding to each class, as suggested by [9].

To each training example feature vector we then appended features computed from the region labelings of superpixels surrounding the training example. The same features were computed and appended to the test examples by using the probability distribution we obtained from M_0. A second SVM, M_1, was then trained on the new training examples with augmented feature vectors and tested on the new test examples with the appended features. This yielded the performance for the first "stack" in our algorithm. The process can be repeated as many times as one likes with new features appended to the feature vectors at every level of stacking to obtain M_n, which takes as input a feature vector augmented by features computed using the probability distributions obtained from $M_0, \ldots, M_{n-1}$. As we will see, the merit of using more than just one stack is questionable, as the inaccuracy of earlier predictions can accumulate and cause performance to decrease. This approach is based on the idea that while with locally constructed features the algorithm classifies each superpixel independently from others, *up to estimation* by $M_0, \ldots, M_{n-1}$, the algorithm can leverage patterns that are exhibited by groups of superpixels.

The motivation for using logistic regression to estimate probabilities from the SVM scores is that each positive score indicates a positive prediction for the corresponding class, whereas a negative score indicates a negative prediction. Since this divides the scores into two categories, logistic regression is perfectly suited to obtaining a probability estimate that predicts that category from a given score. However, for a given superpixel, distinctions between SVM scores that clearly indicate a ranking for the most probable region are often blurred when because of the exponential growth of its denominator, the sigmoid function obtained from logistic regression maps these scores to numbers very close to either 0 or 1. Hence, it may be useful to investigate other methods of obtaining probability estimates from scores. In particular, where there is a clear most-probable class according to the SVM scores, we would like that distinction to be as clearly reflected in the probabilities as well - in this case it may be enough simply to transform the probability distribution such that the aforementioned property is attained. We will later discuss one such transformation.

To measure the success of our algorithm, we compute segmentation score, given by $\frac{\text{true positives}}{\text{true positives+false positives+false negatives}}$ averaged over all classes, on 90% hold out test-set and 10% training set.

2.3 Feature construction

Since outdoor scenes tend to be subdivided along horizontal lines (e.g. sky above trees, trees above grass), we aimed to capture this relationship by considering superpixels surrounding pixel label probabilities to by constructing several features that make use of the semantic context. The most remarkable constructed features are presented in table.

Table 1: Several constructed features

Name	*Description*
`BASELINE`	M_0: computed using features provided in [1]
`LAYER ABOVE`	For each superpixel, compute the above pixels (one layer) and average the predictions for those pixels
`LAYER BELOW`	For each superpixel, compute the below pixels (one layer) and average the probability of each region for those pixels
`WINDOW ABOVE`	Same as `LAYER ABOVE` but for a varying window of 5-10 pixels
`WINDOW BELOW`	Same as `LAYER BELOW` but for a varying window of 5-10 pixels

2.4 Dataset

The data set we obtained from [1] contains a set of roughly 700 images of outdoor scenes, already decomposed into superpixels. See the attached figure for an example of superpixel decomposition of an image. For any given image, each pixel belongs to a superpixel, and each superpixel is given one of eight semantic labeling labelings: $\{sky, tree, road, grass, water, building, mountain, object\}$.

The data set provides pre-computed features for each superpixel: 17 features computed from RGB values, and the average and variance for each feature in 5x5 pixel window. Several GentleBoost algorithms are used to construct a total of 1497 features per superpixel. This data is available at [2].

3 Results

The methods described above produced minimal improvements, but improvements nonetheless, that demonstrate that stacking can be applied to increase semantic understanding, especially with the choice of good features.

3.1 Feature analysis

The constructed features were simple, and hence did not show dramatic improvement in the segmentation score on the test data. However, all tests suggest a positive improvement whenever stacking was used in comparison to the `BASELINE`, M_0. See the corresponding figure for details on each constructed feature.

Overall, the lack of feature improvement suggests several things. First, the constructed features, given their simplicity and the resulting scores, are successful in capturing only a small segment of the surrounding context. Secondly, the linear multi-class SVM fails to capture significant separability in the data. Furthermore, when M_0 produces wrong probabilities, this severely undermines the predictions made by M_1 (see sections below). From

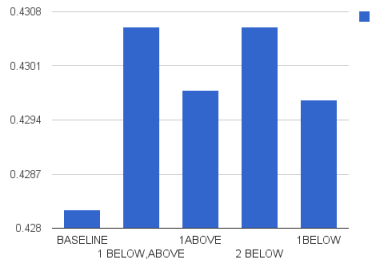

Figure 1: Number refers to stack level; Feature names refer to the `WINDOW` feature and its orientation(above, below)

the constructed feature set, the most impressive improvement was seen in the combination `WINDOW ABOVE`, `WINDOW BELOW`, regardless of being implemented on the same or stacked models. This generated a segmentation score of 43.1%, half a percentage point above `BASELINE`.

We won't list all of the segmentation scores, but only the most notable ones: `WINDOW ABOVE`, `WINDOW BELOW` performed over several layers of stacking. See graphic.

3.2 Scoring function analysis

As mentioned in the Approach section, it seemed reasonable to investigate methods of converting the multiclass SVM scores into probabilities that would preserve a clear ranking of classes from least to most probable. To this end, rather than performing logistic regression, we assumed that given SVM similarity score $s(i, x)$ for class i and example x,

$$k \log p(y = r|x) \propto s(r, x)$$

where k is some constant, yielding

$$p(y = r|x) \propto e^{\frac{s(r,x)}{k}}.$$

Under this assumption, for different values of k we could achieve distributions that appeared more sharply-peaked wherever the SVM scores were higher. This conformed to the property we desired, but for $k \in \{9, 10, 100, 1000, 10000\}$, our algorithm yielded exactly the same segmentation scores (43.30 %), which were also no different from the score achieved when we computed the probabilities using logistic regression. The experiments listed in the table below were performed with one stack, using the `WINDOW ABOVE` and `WINDOW BELOW` features.

Table 2: Several constructed features

K value	*Seg Score Achieved*
k = 9	43.30%
k = 10	43.30 %
k = 100	43.30 %
k = 1000	43.30 %
k = 10000	43.30 %

The uniformity of the results lends evidence to the possibility that our features failed to increase separability as we had hoped. In any case, the results from testing were invariant under this particular transformation of the probability distribution.

3.3 Class accuracy

Segmentation score is computed by taking the average of the segmenation scores for each semantic region. Hence if one semantic region is classified extremely poorly, since all classes are weighted equally, the segmentation score is pulled down.

This was particularly true for class 7 in which the baseline score was 0.077. This class corresponds to the label 'mountain', which is heavily underrepresented in the training set. However our algorithm was able to raise this segmentation score to 0.086 for class 7, by accounting for the context. Additionally, foreground object accuracy was increased, as demonstrated by the sample image below. We can see that the boat's waves are represented as a foreground object, but our algorithm picks out the hull of the boat, and correctly classifies the waves as water. The rest of the classes were left largely unimproved.

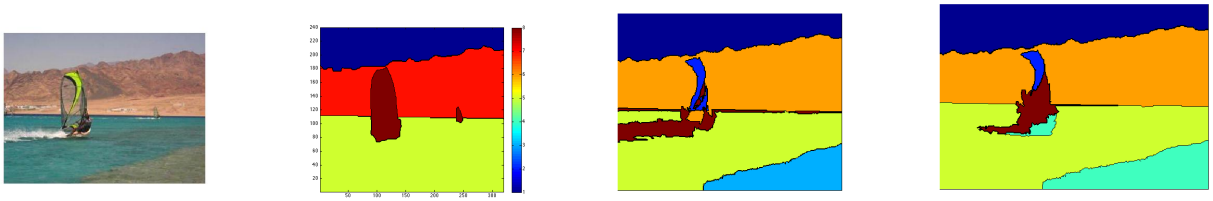

Figure 2: Shows improvement made by stacking. First image is the original. Second is the correct labeling. Third is the labeling made by the baseline model M_0. Fourth is with one level of stacking using `WINDOW ABOVE` and `WINDOW BELOW`.

3.4 Repeated Stacking of one feature

Incidentally, we tried stacking a feature on top of itself, but as expected, this did not increase predictor accuracy after the first few levels of stacking. After the first level of stacking, appending the same feature in subsequent stacks adds little new information, and differs from the same feature appended in previous stacks in that it is computed from a probability distribution obtained from an SVM trained in a later stack level. Probabilities from later levels are not necessarily more accurate, especially where M_0 made wrong predictions, and the graph below indicates that the probability distributions might converge as the number of stacks approaches infinity.

4 Discussion and Future Work

Stacking with the features we designed yielded small improvements of around .4 % in segmentation score over the baseline model. While these improvements are small, they do suggest that we successfully managed to capture at least some of the interaction between superpixels. A heuristic reason stacking failed to give larger improvements is that our features were themselves too simple and did not interpret the context in a way that is linearly separable - for any given superpixel, our features at best gave an estimate for the "composition" of the regions above and below that superpixel, but the variability of this composition in over 700 images of nature might hide any pattern that an SVM could possibly learn. Put simply, we largely failed to design good context features.

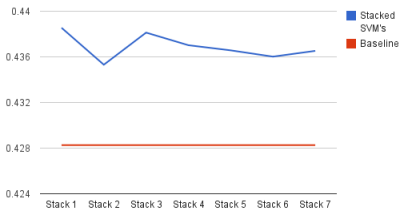

Figure 3: Behavior of Segmentation Score as number of stacked levels increases

As mentioned before, another direction for further research is determining good ways of obtaining probability estimates given scores from multiclass SVMs. In order to make accurate predictions in later levels of stacking, it is important that the probabilities of the correct context are kept comparatively high, so that accuracy of models in previous stacks are not blurred by our a bad probabilistic interpretation of SVM scores. Aside from logistic regression, we gave one possible method to estimate probabilities, but it is unclear whether this method works, since our features might not have been good enough to make use of more accurate probabilties, and since our predictor accuracy neither improved nor worsened with the transformation.

5 Acknowledgements

We express a special thank you to Daphne Koller's lab and especially to Huayan Wang for providing initial guidance for this project.

References

[1] S. Gould, R. Fulton, D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. Proceedings of International Conference on Computer Vision (ICCV), 2009.

[2] http://users.cecs.anu.edu.au/ sgould/index.html

[3] http://svmlight.joachims.org/svm_multiclass.html

[4] S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller. "Multi-Class Segmentation with Relative Location Prior," in *ICML*, 2007.

[5] J.D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[6] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *CVPR*, 2006.

[7] D. H. Wolpert. Stacked Generalizations. *Neural Networks*, 5:241-259, 1992.

[8] L. Breiman. Stacked regressions. *Machine Learning*, 24:49-64, 1996

[9] B. Zadrozny, C. Elkan. "Transforming Classifier Scores into Accurate Multiclass Probability Estimates." *ACM*, 2002.

[10] T. Joachims, T. Finley, and C Yu. Cutting plane training of strcutural SVMs. *Machine Learning*, 77(1):27-59, 2009.