Semantic Scene Segmentation using Random Multinomial Logit

Ananth Ranganathan
Honda Research Institute, USA
Analysis of Traffic Scenes

- Segment objects of interest in a street scene
- Use in intelligent transportation systems
  - Recognition should be perspective invariant
  - Wide intra-class variability
  - Need to work with video
  - Need to be fast
What this talk is about

• An Algorithm for Classification: Random Multinomial Logistic Regression

• Fast
• Scales better with large intra-class variability, perspective etc
• Scales well with number of labels
• Very simple to implement

• A system for Scene Analysis:
  Segment scenes into constituent object and concept labels

Camvid:
mi.eng.cam.ac.uk/research/projects/VideoRec
Multinomial Logistic Regression

\[
\log p(y=i) = \left( R_0 + R_1 x_1 + R_2 x_2 + R_3 x_3 + R_4 x_4 \right)
\]

Simple linear model for log-probability

\[
\log p(y) = \begin{bmatrix}
\begin{array}{cc}
R_{10} & R_{11} \\
R_{20} & R_{21} \\
\vdots & \vdots \\
R_{1N} & R_{2N} \\
\end{array}
\end{bmatrix}
\begin{bmatrix}
1 \\
x_1 \\
x_2 \\
\vdots \\
x_N \\
\end{bmatrix}
\]

Parameters
\[ \log p(y=i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \]

\[ p(y = i|\beta_i, \Phi) = \frac{\exp \beta_i \cdot \Phi}{1 + \sum_j \exp \beta_j \cdot \Phi} \]

- **Supervised learning for \( \beta \) using non-linear least squares**
  - L-BFGS used in this work
  - Also gives variances of coefficient estimates
- **MAP learning with L2-regularization**
  - Avoids overfitting and large parameter values
Multinomial Logistic Regression

The Good

• Fast predictions at runtime
  - Scales well with number of classes
  - Labeling probability is available
• Model is stable w.r.t slight changes in training set
• Used widely in biology, sociology, machine learning

The Bad

• Variance of coefficients increases with number of features
• Not suited for large feature spaces
• Sensitive to noise in training data
• Training with large datasets is slow

... and the Beautiful
Random Multinomial Logit!

RML Result
Random Multinomial Logit

Basic idea similar to Random Forests of Decision trees

Randomly generated trees
Result obtained by averaging

High variance, overfitting, sensitive to noise, unsuitable for large feature spaces
Random Multinomial Logistic Regression

- Use multiple Multinomial Logit models each trained from a different subset of training data
- Randomly selected feature set
- Final prediction is simply the average
- Model bias and prediction variance are both reduced
- Robust to noise and can work in large feature spaces

From Prinzie & Van den Poel, 2008
RML Training

Feature computation

Sampling

Random feature selection

Learning

RML

[β] ... [β]
Classification using RML

Compute feature responses

Get model outputs

[\beta] \ldots [\beta]

\sum

Average
Scene Segmentation using RML

- RML used as texture-based classifier
- Texture space is discretized into *Textons*
- Leung-Malik filter bank to compute texture
  - 17 filters Gaussian, DoG, LoG
- Shape-texton features used as input to RML

**Extract texture**
filter bank

**Clustering**
k-means

**Feature computation**
Shape-texton features

**Random Multinomial Logit Classifier**
• Introduced in (Shotton et al., ECCV 2006)
• Features computed on texton-mapped images
• Each feature comprises a rectangular region $r$ and a texton $t$
• The feature measures the proportion of the texton $t$ inside the rectangle $r$, and is applied to each pixel of the image
• Size of rectangle $r$ and the texton $t$ are generated randomly
• Fast computation using integral images
• Layout and context is captured by rectangular regions
Caveat

• Feature space is huge
• Many features are useless
• Large number of models will be required to get good results
Need for Feature Selection

Replace statistically insignificant features

Multi-collinearity
Hard to detect!
Replacing Insignificant features

\[
\log p(y=i) = \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4
\]

- Statistically insignificant features have “small” \( \beta \) coefficients
- Small - \( \beta_i < 2 \times \text{std. dev}(\beta_i) \)
  - Variances available from Least-squares learning
- Select new feature randomly to replace the insignificant feature
- Re-learn multinomial logit model
Random Search in Feature Space

- Multi-collinearity is expensive to detect
- Easier to randomly swap features that improve model
- Improvement is quantified by log-likelihood on training data
  - Higher log-likelihood => better feature
In one round of feature selection do -
- If there are insignificant features
  - replace the feature and re-learn model
- Else if all features are significant
  - pick a model feature $\Phi_i$ at random
  - replace $\Phi_i$ with randomly picked feature $\Phi_j$ and re-learn model
  - if log-likelihood of new model is greater then keep it, else discard it
• Pixel-wise classification based on texture gives noisy results

\[ p(x) \propto \exp \left( - \sum_i V_{col}(x_i) + V_{tex}(x_i) + V_{loc}(x_i) - \sum_{ij} V_{edge}(x_i, x_j) \right) \]

• Include color, location, and edge information in a Conditional Random Field (CRF) - details as in Shotton et al., IJCV 2009
Overall System

Color modeling using mixture of Gaussians

Texture modeling using textons

Texture classifier learnt using RML

CRF includes all constraints

Inference using Graphcut

Edge potentials

Location potentials

$r_i, t_i = (\text{texture}, \text{color})$
Experiments

• Comparison against random forests and TextonBoost on two datasets
  - implementations in Matlab

• TextonBoost implemented from (Shotton et al., IJCV 2009)
• Boosting selects decision stumps based on shape-texton features
Random Forests implementation

- Extremely Random Trees (Geurts et al., Machine Learning, April 2006)
- Decision trees have randomly selected shape-texton feature at nodes with random threshold
- Label histogram at each leaf
- Final output is average of histogram from all the trees

$P(c|l)$
Results - Motorbike video dataset

- Videos from moving vehicles with camera pointed backwards
- 4 categories detected - bike, road, sky, other
- Different types of bikes and road conditions
- 63 labeled frames from 6 sequences, 5800 frames in total
- 15 multinomial logit regressors each with 15 features each
Comparison

<table>
<thead>
<tr>
<th></th>
<th>RML</th>
<th>RML + Feature selection</th>
<th>RML + Feat. Sel. + CRF</th>
<th>Random Forests</th>
<th>Depth-limited Random Forests</th>
<th>Random Forests + CRF</th>
<th>TextonBoost</th>
<th>TextonBoost + CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall (%)</td>
<td>73.6</td>
<td>77.1</td>
<td>82.1</td>
<td>63.2</td>
<td>49.8</td>
<td>66.7</td>
<td>78.5</td>
<td>81.2</td>
</tr>
<tr>
<td>Bike (%)</td>
<td>51.6</td>
<td>53.1</td>
<td>62.0</td>
<td>42.7</td>
<td>31.6</td>
<td>45.9</td>
<td>57.8</td>
<td>60.7</td>
</tr>
</tbody>
</table>
VOC 2008 Dataset

- Test dataset for the Pascal VOC 2008 object detection challenge
- 20 classes
- 25 multinomial regressors with 20 features each
- Compared against winner of 2008 challenge (Csurka & Perronin, BMVC 2008)
## VOC 2008 Dataset

<table>
<thead>
<tr>
<th>XRCE_Seg</th>
<th>Input</th>
<th>RML+CRF</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1" alt="Input Image" /></td>
<td><img src="image2" alt="RML+CRF Image" /></td>
<td><img src="image3" alt="Manual Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image4" alt="Input Image" /></td>
<td><img src="image5" alt="RML+CRF Image" /></td>
<td><img src="image6" alt="Manual Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image7" alt="Input Image" /></td>
<td><img src="image8" alt="RML+CRF Image" /></td>
<td><img src="image9" alt="Manual Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image10" alt="Input Image" /></td>
<td><img src="image11" alt="RML+CRF Image" /></td>
<td><img src="image12" alt="Manual Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XRCE_Seg</th>
<th>RML+CRF</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.8</td>
<td>15.2</td>
<td>16.7</td>
</tr>
<tr>
<td>31.2</td>
<td>20.1</td>
<td>16.7</td>
</tr>
</tbody>
</table>
• Performance levels off with increasing number of features
• Regularization is essential with large number of features
• Performance also levels off with number of models

• Runtime on motorbike dataset:
  • RML - 0.12 sec/frame
  • Random forests - 4.1 sec/frame
  • TextonBoost - 6.03 sec/frame
Future

• Sparse multinomial logit
  - L1 regularization
  - No need for feature selection
  - Slow!

• Temporal constraints - optic flow, label tracking
• Superpixels and region statistics
• Shape models etc
QuickTime™ and a decompressor are needed to see this picture.