Decision Tree Fields

Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, Pushmeet Kohli

Barcelona, 8th November 2011
Random Fields in Computer Vision

- Markov Random Fields (MRF) 
  (Kindermann and Snell, 1980), (Li, 1995), 
  (Blake, Kohli, Rother, 2011)
- Conditional Random Fields (CRF) 
  (Lafferty, McCallum, Perreira, 2001)
- **Structured prediction** of multiple dependent variables
CRFs: How do we use them?

- Factor graph notation (Kschischang, Frey, Loeliger, 1998)
- x: observed image
- y_i, y_j: dependent variables at pixel i and j
Introduction

CRFs: How do we use them?

- Unary energy $E_A(y_i, x)$
- Machine learning (SVM, Boosting, Random Forests, etc.)
CRFs: How do we use them?

- Pairwise energy $E_C(y_i, y_j, x)$
- Generalized Potts, image independent
- Contrast-sensitive smoothing (e.g. GrabCut, TextonBoost)

$$E_C(y_i, y_j, x) = \exp(-\alpha \|x_i - x_j\|^2)$$
CRFs: How do we use them?

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_k, x) \]

\[ y_i \]

\[ y_j \]

\[ y_k \]
CRFs: How do we use them?

\[ E_A(y_i, x), \quad E_C(y_i, y_j, y_k, x) \]

Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, Pushmeet Kohli
Decision Trees in Computer Vision

- Random Forests (Breiman, MLJ 2000)
- Non-parametric, infinite model capacity
- Fast inference and training, parallelizable
- (Shotton et al., 2008, 2011), (Saffari et al., 2009), (Gall and Lempitsky, 2009), etc.
- No structured prediction
Contributions

1. Learn image-dependent interactions
2. Combine random fields and decision trees
3. Efficient training
4. Superior empirical performance
Decision Tree Classifiers
Decision Tree Classifiers
Decision Trees for Image Labeling
Decision Trees for Image Labeling (cont)

- Apply decision tree, to each pixel \textit{independently}
Decision Trees for Image Labeling (cont)

- Apply decision tree, to each pixel \textit{independently}
Decision Tree Field (DTF) Example
Unary Factor Example

- $x$: entire observed image
- $y_i$: prediction at pixel $i$, $y_i \in \{1, 2, 3, 4\}$
- $E_A(y_i, x)$: energy function
Unary Factor Example

\[ E_A(y_i, x) \]
 Unary Factor Example 

\[ E_A(y_i, x) \]
 Unary Factor Example

\[ E_A(y_i, x) \]
Unary Factor Example

\[ E_A(y_i, x) \]
Unary Factor Example

\[
E_A(y_i, x) = \sum_{q \in \text{Path}(x)} w_A(q, y_i)
\]
Pairwise Factor Example
Pairwise Factor Example

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_j, x) \]
Pairwise Factor Example

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_j, x) \]
Pairwise Factor Example

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_j, x) \]

\[ C \]

\[ x \]

\[ y_i \]

\[ y_j \]
Pairwise Factor Example

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_j, x) \]
Pairwise Factor Example

\[ E_A(y_i, x) \]

\[ E_C(y_i, y_j, x) = \sum_{q \in \text{Path}(x)} w_A(q, y_i, y_j) \]
Full DTF Model

\[ E(\mathbf{y}, x, \mathbf{w}) = \sum_{F \in \mathcal{F}} E_{t_F}(\mathbf{y}_F, x_F, w_{t_F}). \]

\[ p(\mathbf{y}|x, \mathbf{w}) = \frac{1}{Z(x, \mathbf{w})} \exp(-E(\mathbf{y}, x, \mathbf{w})), \]

\[ Z(x, \mathbf{w}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp(-E(\mathbf{y}, x, \mathbf{w})) \]

- \( x \): image, \( y \): predicted labels, one for each pixel
- \( w \): weights/energies, to be learned from data
- How is this different from other models?
- What about learning and inference?
**Full DTF Model**

\[
E(y, x, w) = \sum_{F \in \mathcal{F}} E_{t_F}(y_F, x_F, w_{t_F}).
\]

\[
p(y|x, w) = \frac{1}{Z(x, w)} \exp(-E(y, x, w)),
\]

\[
Z(x, w) = \sum_{y \in \mathcal{Y}} \exp(-E(y, x, w))
\]

- \(x\): image, \(y\): predicted labels, one for each pixel
- \(w\): weights/energies, to be learned from data
- How is this different from other models?
- What about learning and inference?
Relationship to Other Models

- Generalizes random forests (learned weights)
- Markov random fields

Here 2 trees
Relationship to Other Models

- Generalizes random forests (learned weights)
- Markov random fields
CPT-Trees (1995)

- Conditional Probability Table Trees
- (Glesner, Koller, 1995), (Boutilier et al., 1996)
- Decision tree on states of random variables
- Limited to Bayesian networks
Learning a Markov Chain

Dependency Networks

- Learn $p(x_i | x_{\mathcal{V}\setminus{i}})$
- (Heckermann et al., 2000)
- Decision tree on states of random variables
- Inference requires simulation (pseudo-Gibbs sampling)

Random Forest Random Field

- (Payet and Todorovic, 2010)
- Decision tree determines sampler
- Inference: Swendsen-Wang Metropolis MCMC
Learning DTFs

Given iid data \( \{(x, y)_i\}_{i=1,...,N} \), need to learn

- Structure of the factor graph,
- Tree structure defined by split functions,
- Weight parameters in decision nodes.

Let us assume structure and trees are given
Learning DTFs

Given iid data \( \{(x, y)_i\}_{i=1,...,N} \), need to learn

- Structure of the factor graph,
- Tree structure defined by split functions,
- Weight parameters in decision nodes.

Let us assume structure and trees are given.
Training

Maximum Likelihood Estimation, given ground truth $y^*$

$$w^* = \arg\max_w \log p(y^* | x, w)$$
Training

Maximum Likelihood Estimation, given ground truth $y^*$

$$w^* = \arg\max_w \log p(y^* | x, w)$$

Intractable!
Training
Training
Training

Maximum Pseudo-Likelihood Estimation (Besag, 1974)

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \log p(y_i | y_{\mathcal{V}\{i\}}, x, \mathbf{w})$$

with ground truth $y^*$ and

$\mathcal{V}$: set of pixels in all images.

(details in paper)
Efficient Training by Subsampling

1M pixels
Efficient Training by Subsampling

... 1M pixels
Efficient Training by Subsampling

We subsample our training set to train a structured model.
Training Algorithm

1. Fix factor graph structure
2. For each factor: learn classification tree
3. Jointly optimize convex pseudo-likelihood objective in $w$
Test-time Inference in DTFs

1. Energy minimization (MAP)
   E.g. TRW-S

2. Maximum Posterior Marginal (MPM)
   E.g. Gibbs sampling
Experiment: Inpainting
Experiment: Inpainting
Experiment: Inpainting

![Inpainting Example]
Experiment: Inpainting
Experiment: Inpainting

Training set (300 images)
Experiment: Inpainting

Test set (100 images, disjoint characters)
Instances

- Densely-connected, 64 neighbors
- Each instance: 10k variables, 300k factors
- → hard to minimize energy

www.nowozin.net/sebastian/papers/DTF_CIP_instances.zip
Instances

- Densely-connected, 64 neighbors
- Each instance: 10k variables, 300k factors
- $\rightarrow$ hard to minimize energy

www.nowozin.net/sebastian/papers/DTF_CIP_instances.zip
Experiment: Body-part Recognition

- Body part recognition (Shotton et al., CVPR 2011)
- 1500 training images, 150 test images
Experiment: Body-part Recognition (cont)

Training set
Experiment: Body-part Recognition, Results

Input / Truth

unary

+1

+1,5,20

MRF

DTF
## Experiment: Body-part Recognition, Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Shotton et al.</th>
<th>unary</th>
<th>+1</th>
<th>+1,20</th>
<th>+1,5,20</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>avg-acc</td>
<td>34.4</td>
<td>36.15</td>
<td>37.82</td>
<td>38.00</td>
<td>39.30</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>6h34</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(30h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.3M</td>
<td>6.2M</td>
<td>6.2M</td>
<td>6.3M</td>
</tr>
<tr>
<td>DTF</td>
<td>avg-acc</td>
<td>-</td>
<td>-</td>
<td>39.59</td>
<td>40.26</td>
<td>41.42</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>(40h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>-</td>
<td>6.8M</td>
<td>7.8M</td>
<td>8.8M</td>
</tr>
</tbody>
</table>
## Experiment: Body-part Recognition, Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Shotton et al.</th>
<th>unary</th>
<th>+1</th>
<th>+1,20</th>
<th>+1,5,20</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>avg-acc</td>
<td>34.4</td>
<td>36.15</td>
<td>37.82</td>
<td>38.00</td>
<td>39.30</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>6h34</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(30h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.3M</td>
<td>6.2M</td>
<td>6.2M</td>
<td>6.3M</td>
</tr>
<tr>
<td>DTF</td>
<td>avg-acc</td>
<td>-</td>
<td>-</td>
<td>39.59</td>
<td>40.26</td>
<td>41.42</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>(40h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.8M</td>
<td>7.8M</td>
<td>8.8M</td>
<td></td>
</tr>
</tbody>
</table>
### Experiment: Body-part Recognition, Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Shotton et al.</th>
<th>unary</th>
<th>+1</th>
<th>+1,20</th>
<th>+1,5,20</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>avg-acc</td>
<td>34.4</td>
<td>36.15</td>
<td>37.82</td>
<td>38.00</td>
<td>39.30</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>6h34</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(30h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.3M</td>
<td>6.2M</td>
<td>6.2M</td>
<td>6.3M</td>
</tr>
<tr>
<td>DTF</td>
<td>avg-acc</td>
<td>-</td>
<td></td>
<td>39.59</td>
<td>40.26</td>
<td>41.42</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>-</td>
<td></td>
<td>*</td>
<td>*</td>
<td>(40h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.8M</td>
<td>7.8M</td>
<td>8.8M</td>
<td></td>
</tr>
</tbody>
</table>
## Experiment: Body-part Recognition, Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Shotton et al.</th>
<th>unary</th>
<th>+1</th>
<th>+1,20</th>
<th>+1,5,20</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>avg-acc</td>
<td>34.4</td>
<td>36.15</td>
<td></td>
<td>37.82</td>
<td>38.00</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>6h34</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.3M</td>
<td>6.2M</td>
<td>6.2M</td>
<td>6.3M</td>
</tr>
<tr>
<td>DTF</td>
<td>avg-acc</td>
<td>-</td>
<td>-</td>
<td></td>
<td>39.59</td>
<td>40.26</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>-</td>
<td>6.8M</td>
<td>7.8M</td>
<td>8.8M</td>
</tr>
</tbody>
</table>
DTF Summary

- Decision Tree Fields: non-parametric CRF model for discrete image labeling tasks
- Non-parametric: model class can scale with training set size
- Scalable, can make use of large training sets,
- Conditional interactions: richer models without latent variables
DTF Summary

- Decision Tree Fields: non-parametric CRF model for discrete image labeling tasks
- Non-parametric: model class can scale with training set size
- Scalable, can make use of large training sets,
- Conditional interactions: richer models without latent variables

Code will be made available after CVPR deadline!
Thank you!
DTF: Linearity

\[ E_{t_t} (y_F, x_F, w_{t_F}) \] can be written as a function linear in \( w_{t_F} \),

\[
\sum_{n \in \text{Tree}(t_F)} \sum_{z \in Y_F} w_{t_F}(q, z) B_{t_F}(q, z; y_F, x_F),
\]

where

\[
B_{t_F}(q, z; y_F, x_F) = \begin{cases} 
1 & \text{if } n \in \text{Path}(x_F) \text{ and } z = y_F, \\
0 & \text{otherwise}.
\end{cases}
\]

- → overall energy function is \textit{linear} in \( w \)
- → (pseudo-)likelihood function is log-concave
- Here: not necessarily unique maximizer
Learning the Decision Trees

How to learn the decision tree?

- Ideal world: learn entire model jointly
- Here: learn decision trees using common information gain criterion
- Pairwise and order-$k$ factors: treat as $\mathcal{L} \times \mathcal{L}$ classification problem ($\mathcal{L}^k$)
- Although trees are trained independently, overcounting is avoided by optimizing the weights jointly

Training summary

1. For each factor type, train a decision tree using information gain
2. Initialize tree weights to zero
3. Maximize the pseudolikelihood (using L-BFGS)
Learning the Decision Trees

How to learn the decision tree?

- Ideal world: learn entire model jointly
- Here: learn decision trees using common information gain criterion
- Pairwise and order-\(k\) factors: treat as \(\mathcal{L} \times \mathcal{L}\) classification problem (\(\mathcal{L}^k\))
- Although trees are trained independently, overcounting is avoided by optimizing the weights jointly

Training summary

1. For each factor type, train a decision tree using information gain
2. Initialize tree weights to zero
3. Maximize the pseudolikelihood (using L-BFGS)
Experiment: Body-part Recognition, Results

Figure: Test recognition results. MRF (top) and DTF (bottom).
### Experiment: Body-part Recognition, Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Shotton et al.</th>
<th>unary</th>
<th>+1</th>
<th>+1,20</th>
<th>+1,5,20</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>avg-acc</td>
<td>34.4</td>
<td>36.15</td>
<td>37.82</td>
<td>38.00</td>
<td>39.30</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>6h34</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(30h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>6.3M</td>
<td>6.2M</td>
<td>6.2M</td>
<td>6.3M</td>
</tr>
<tr>
<td>DTF</td>
<td>avg-acc</td>
<td>-</td>
<td>-</td>
<td>39.59</td>
<td>40.26</td>
<td>41.42</td>
</tr>
<tr>
<td></td>
<td>runtime</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>(40h)*</td>
</tr>
<tr>
<td></td>
<td>weights</td>
<td>-</td>
<td>-</td>
<td>6.8M</td>
<td>7.8M</td>
<td>8.8M</td>
</tr>
</tbody>
</table>

**Table:** Body-part recognition results: mean per-class accuracy, training time on a single 8-core machine, and number of model parameters.
Experiment: Body-part Recognition, Results

Figure: Learned horizontal interactions: Left: mean silhouette reaching the 32 leaf nodes in the learned tree. One leaf (marked red) and corresponding effective 32 × 32 weight matrix. Visualizing the most attractive (blue) and most repulsive (red) weights. Right: superimposing label-label interactions on test images, (a) matching the pattern, (b) no match, interaction is inactive.
Experiment: Body-part Recognition, Results

Figure: **Learned horizontal interactions**: Left: mean silhouette reaching the 32 leaf nodes in the learned tree. One leaf (marked red) and corresponding effective $32 \times 32$ weight matrix. Visualizing the most attractive (blue) and most repulsive (red) weights. Right: superimposing label-label interactions on test images, (a) matching the pattern, (b) no match, interaction is inactive.
Experiment: Body-part Recognition, Results

**Figure:** Learned horizontal interactions: Left: mean silhouette reaching the 32 leaf nodes in the learned tree. One leaf (marked red) and corresponding effective $32 \times 32$ weight matrix. Visualizing the most attractive (blue) and most repulsive (red) weights. Right: superimposing label-label interactions on test images, (a) matching the pattern, (b) no match, interaction is inactive.
Experiment: Snakes

- Simplest tasks with conditional label-label structure
- Snake: 10 labels from head (black) to tail (white)
- Image contains perfect instructions
  - red = “go up”
  - yellow = “go right”
  - green = “go down”
  - blue = “go left”
- Myopic decisions are impossible (weak local evidence)
- Training: 200 small images
- Testing: 100 small images
- Features: relative pixel color tests
Experiment: Snakes

- Simplest tasks with conditional label-label structure
- Snake: 10 labels from head (black) to tail (white)
- Image contains perfect instructions
  - red = “go up”
  - yellow = “go right”
  - green = “go down”
  - blue = “go left”
- Myopic decisions are impossible (weak local evidence)
- Training: 200 small images
- Testing: 100 small images
- Features: relative pixel color tests
## Experiment: Snakes, Results

<table>
<thead>
<tr>
<th></th>
<th>RF</th>
<th>Unary</th>
<th>MRF</th>
<th>DTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>90.3</td>
<td>90.9</td>
<td>91.9</td>
<td><strong>99.4</strong></td>
</tr>
<tr>
<td>Accuracy (tail)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Accuracy (mid)</td>
<td>28</td>
<td>28</td>
<td>38</td>
<td>95</td>
</tr>
</tbody>
</table>

**Table:** Test set accuracies for the snake data set.
Experiment: Snakes, Results

Figure: Predictions on a novel test instance.
Experiment: Snakes, Conclusion

Here,

- Strong pairwise interactions help when having weak local evidence,
- Pairwise interactions are strong because they condition on the image,
- 200 training images are enough
Factor type in DTFs

Every factor type has one

- **scope**: relative set of variables it acts on,
- **decision tree**: tree with split functions,
- **weight parameters**: in each node

Energy is the sum along path of traversed nodes

\[
E_{tF}(y_F, x_F, w_{tF}) = \sum_{q \in \text{Path}(x_F)} w_{tF}(q, y_F)
\]
Efficient Training

Minimize in $\mathbf{w}$ the regularized negative log-pseudolikelihood,

$$\ell_{npl}(\mathbf{w}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \ell_i(\mathbf{w}) - \frac{1}{|\mathcal{V}|} \sum_{t} \log p_t(w_t),$$

with

$$\ell_i(\mathbf{w}) = -\log p(y_i|y_{\mathcal{V}\setminus\{i\}}, \mathbf{x}, \mathbf{w})$$

and

$\mathcal{V}$ : set of pixels in all images.
Efficient Training

\[ \ell_{npl}(w) = \frac{1}{|V|} \sum_{i \in V} \ell_i(w) - \frac{1}{|V|} \sum_t \log p_t(w_t), \]
Efficient Training

\[ \ell_{npl}(w) = \mathbb{E}_{i \sim \mathcal{U}(\mathcal{V})} [\ell_i(w)] - \frac{1}{|\mathcal{V}|} \sum_t \log p_t(w_t), \]
Efficient Training

\[ \ell_{npl}(w) = \mathbb{E}_{i \sim \mathcal{U}(\mathcal{V})} [\ell_i(w)] - \frac{1}{|\mathcal{V}|} \sum_t \log p_t(w_t), \]

- Composite objective: expectation + simple function
- Approximate expectation, deterministic, for \(\mathcal{V}' \subset \mathcal{V}\),

\[ \ell_{npl}(w) \approx \frac{1}{|\mathcal{V}'|} \sum_{i \in \mathcal{V}'} \ell_i(w) - \frac{1}{|\mathcal{V}|} \sum_t \log p_t(w_t). \]

- → MPLE enables subsampling on variable level