Nonparametric Conditional Random Fields for Image Labelling

Jeremy Jancsary, Sebastian Nowozin, and Carsten Rother
Microsoft Research Cambridge

NIPS Workshop on
Modern Nonparametric Methods in Machine Learning

Friday, December 7, 2012
Structured Prediction

How to model:

\[ p(y \mid x; w) \]

correct labels
„ground truth“

observed input

Structured Prediction
Structured Prediction

How to model:

$p(y | x; w)$

correct labels
„ground truth“

observed input

Structured Prediction
Regression Tree Fields

Non-parametric

$$f(x_F) \leq c$$

Gaussian conditional random fields

Proposed Model
Factor Energy

\[
E(y_F \mid x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F)
\]

Proposed Model
The proposed model uses factor energy to compute the energy of a configuration $\mathbf{y}_F$ given the observed input image $\mathbf{x}_F$. The energy function is defined as:

$$
E(\mathbf{y}_F \mid \mathbf{x}_F) = \frac{1}{2} \mathbf{y}_F^T \mathbf{Q}(\mathbf{x}_F) \mathbf{y}_F - \mathbf{y}^T \mathbf{L}(\mathbf{x}_F) \mathbf{b}(\mathbf{x}_F)
$$

where $\mathbf{Q}(\mathbf{x}_F)$ is the covariance matrix, $\mathbf{L}(\mathbf{x}_F)$ and $\mathbf{b}(\mathbf{x}_F)$ are functions of the input image. The model is quadratic in the factors, which allows for efficient optimization.

Factor Energy

Proposed Model
Factor Energy

\[ E(y_F \mid x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F) \]

Proposed Model
The energy of a particular factor $F_t$ is given by:

$$E(\mathbf{y}_F | \mathbf{x}_F) = \frac{1}{2} \mathbf{y}_F^T \mathbf{Q}(\mathbf{x}_F) \mathbf{y}_F - \mathbf{y}^T \mathbf{L}(\mathbf{x}_F) \mathbf{b}(\mathbf{x}_F)$$
Factor Energy

$$E(y_F | x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F)$$

Proposed Model
The energy of a particular factor is given by:

\[ E(y_F | x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F) \]

where \( Q(x_F) \) is a positive-definite matrix and \( L(x_F) \) and \( b(x_F) \) are functions of the observed image and the factor instantiation, respectively.
Factor Energy

\[ E(y_F \mid x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F) \]

Proposed Model
Factor Energy

\[ E(y_F | x_F) = \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F) \]

Proposed Model
The energy of a particular factor \( y_{F} \) as maps to the parameters \( \mathbf{Q}(x_{F}) \) arises as sums of per-factor contributions. Each local energy term working on a subset of pixels is \( \mathbf{E}(x_{F} | y_{F}) = \frac{1}{2} \mathbf{y}_{F}^{T} \mathbf{Q}(x_{F}) \mathbf{y}_{F} - \mathbf{y}^{T} \mathbf{L}(x_{F}) \mathbf{b}(x_{F}) \).
Factor Energy

\[\begin{align*}
E(y_F \mid x_F) &= \frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F)
\end{align*}\]

Proposed Model
Proposed Model

Global Energy

\[ w = \{Q, L\} \]

\[
\begin{align*}
\frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F)
\end{align*}
\]

\[
E(y \mid x; w) = \sum_F E(y_F \mid x_F; w)
\]

\[
\text{factor contribution}
\]

Proposed Model
Global Energy

\[ w = \{ Q, L \} \]

\[
\begin{align*}
\frac{1}{2} y_F^T Q(x_F) y_F - y^T L(x_F) b(x_F)
\end{align*}
\]

Remember:

\[
E(y \mid x; w) = \frac{1}{2} y^T Q(x; w) y - y^T (Q(x_F) y_F - L(x_F) b(x_F))
\]

Proposed Model
Global Energy

\[ w = \{ Q, L \} \]

Remember:

\[ \frac{1}{2} y_F^T Q(x_F) y_F - y_F^T L(x_F) b(x_F) \]

\[ p(y \mid x; w) \propto \exp[-E(y \mid x; w)] \]

Proposed Model
Predicting

Given input image:

\[ \hat{y}(x) = \arg\max_y p(y \mid x) = \text{[???]} = \text{[???]} \]
Given input image:

\[ \hat{y}(x) = \arg\max_y p(y \mid x) = \mu = \]

Predicting
Given input image:

\[ \hat{y}(x) = \arg\max_y p(y \mid x) = \mu = [Q(x; w)]^{-1}l(x; w) \]
Given input image:

\[ \hat{y}(x) = \arg\max_y p(y \mid x) = \mu = [Q(x; w)]^{-1} l(x; w) \]

Predicting
Given input image:

\[ \hat{y}(x) = \arg\max_y p(y \mid x) = \mu = \left[ Q(x; w) \right]^{-1} l(x; w) \]
Jointly choose structure of trees and parameters at leaves to minimize empirical risk:

\[
\frac{1}{N} \sum_{i}^{N} \ell(\hat{y}(x^{(i)}; w), y^{(i)}) \approx \mathbb{E}_{p(x,y)} [\ell(\hat{y}(x; w), y)]
\]
Jointly choose structure of trees and parameters at leaves to minimize empirical risk:

\[
\frac{1}{N} \sum_{i=1}^{N} \ell(\hat{y}(x^{(i)}; w), y^{(i)}) \approx \mathbb{E}_{p(x,y)} [\ell(\hat{y}(x; w), y)]
\]

➢ Efficient greedy algorithm
(please come to poster for details)
Image Denoising Results

<table>
<thead>
<tr>
<th>Truth</th>
<th>Input</th>
<th>EPLL (state of the art)</th>
<th>Ours (best)</th>
<th>Ours (fast)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Average PSNR on test set

<table>
<thead>
<tr>
<th>Time required to generate reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>38s</td>
</tr>
<tr>
<td>1,275s</td>
</tr>
<tr>
<td>1.6s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$(\sigma = 40)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.17db</td>
</tr>
<tr>
<td>26.43db</td>
</tr>
<tr>
<td>26.24db</td>
</tr>
</tbody>
</table>
### Image Denoising Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Time Required</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth</td>
<td>38s</td>
<td>26.17db</td>
</tr>
<tr>
<td>Input</td>
<td>1,275s</td>
<td>26.43db</td>
</tr>
<tr>
<td>EPLL (state of the art)</td>
<td></td>
<td>26.24db</td>
</tr>
<tr>
<td>Ours (best)</td>
<td>1.6s</td>
<td></td>
</tr>
<tr>
<td>Ours (fast)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\((\sigma = 40)\)

Average PSNR on test set:

- Ours (best): 26.43db
- Ours (fast): 26.24db
### Image Denoising Results

<table>
<thead>
<tr>
<th>Truth</th>
<th>Input</th>
<th>EPLL (state of the art)</th>
<th>Ours (best)</th>
<th>Ours (fast)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Truth" /></td>
<td><img src="image2" alt="Input" /></td>
<td><img src="image3" alt="EPLL" /></td>
<td><img src="image4" alt="Ours (best)" /></td>
<td><img src="image5" alt="Ours (fast)" /></td>
</tr>
</tbody>
</table>

- **Time required to generate reconstruction**
  - 38s
  - 1,275s
  - 1.6s

- **Average PSNR on test set**
  - 26.17db
  - 26.43db
  - 26.24db

- **(σ = 40)**