UPPSALA
UNIVERSITET

DCTD: Deep Conditional Target Densities for Accurate Regression

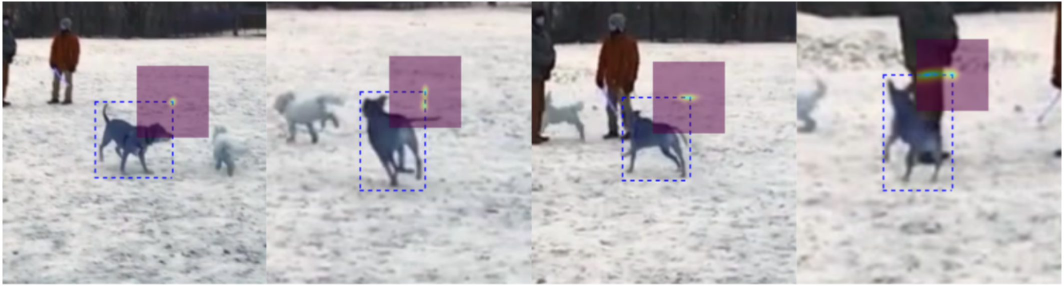
Fredrik K. Gustafsson
Uppsala University

SysCon μ-seminar
Uppsala, November 1, 2019

DCTD: Deep Conditional Target Densities for Accurate Regression

Fredrik K. Gustafsson*, Martin Danelljan* (ETH Zurich), Goutam Bhat (ETH Zurich), Thomas B. Schön

We propose Deep Conditional Target Densities (DCTD), a novel and general regression method with a clear probabilistic interpretation. DCTD models the conditional target density $p(y|x)$ *by using a neural network to directly predict the un-normalized density from the input-target pair* (x, y). *This model of* $p(y|x)$ *is trained by minimizing the associated negative log-likelihood, approximated using Monte Carlo sampling. Notably, our method achieves a* 1.9% *AP improvement over Faster-RCNN for object detection on COCO, and sets a new state-of-the-art on visual tracking when applied for bounding box regression.*



Outline

1. Background: regression using deep neural networks
   1.1 Direct regression
   1.2 Probabilistic regression
   1.3 Confidence-based regression
2. Deep Conditional Target Densities (DCTD) for accurate regression
   2.1 Training
   2.2 Prediction
3. Experiments
   3.1 Age estimation, head-pose estimation, object detection
   3.2 Generic visual object tracking

Supervised regression: learn to predict a continuous target value $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

UPPSALA UNIVERSITET

Supervised regression: learn to predict a continuous target value $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

Deep neural network (DNN): a function $f_\theta : \mathcal{U} \to \mathcal{O}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $u \in \mathcal{U}$ to an output $f_\theta(u) \in \mathcal{O}$.

1.1 Direct regression

Direct regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

Direct regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\operatorname{argmin}}\, J(\theta').$$

UPPSALA UNIVERSITET

Direct regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\operatorname{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Direct regression: train a DNN $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\text{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Minimizing $J(\theta)$ then corresponds to minimizing the *negative log-likelihood* $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$, for a specific model $p(y|x;\theta)$ of the conditional target density.

Direct regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\operatorname{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Minimizing $J(\theta)$ then corresponds to minimizing the *negative log-likelihood* $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$, for a specific model $p(y|x;\theta)$ of the conditional target density.

For example, the L^2 loss corresponds to a fixed-variance Gaussian model (1D case):

$$p(y|x;\theta) = \mathcal{N}(y; f_\theta(x), \sigma^2).$$

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

For example, a general 1D Gaussian model can be realized as:

$$p(y|x;\theta) = \mathcal{N}\big(y;\mu_\theta(x),\, \sigma_\theta^2(x)\big), \quad f_\theta(x) = [\, \mu_\theta(x) \quad \log\sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2.$$

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

For example, a general 1D Gaussian model can be realized as:

$$p(y|x;\theta) = \mathcal{N}\big(y;\mu_\theta(x), \sigma_\theta^2(x)\big), \quad f_\theta(x) = [\,\mu_\theta(x) \quad \log \sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2.$$

The negative log-likelihood $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$ then corresponds to the loss:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{(y_i - \mu_\theta(x_i))^2}{\sigma_\theta^2(x_i)} + \log \sigma_\theta^2(x_i).$$

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-based regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity over y to predict the target:

$$y^\star = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y)$$

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-based regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity over y to predict the target:

$$y^\star = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

UPPSALA UNIVERSITET

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-based regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity over y to predict the target:

$$y^\star = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

Commonly employed for image-coordinate regression, e.g. human pose estimation [11], where the DNN predicts a 2D confidence heatmap over image-coordinates y.

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-based regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity over y to predict the target:

$$y^\star = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

Commonly employed for image-coordinate regression, e.g. human pose estimation [11], where the DNN predicts a 2D confidence heatmap over image-coordinates y. Recently, the approach was also employed by IoU-Net [4] for bounding box regression in object detection, which in turn was utilized by the ATOM [3] visual tracker.

Outline

2. Deep Conditional Target Densities (DCTD) for accurate regression

UPPSALA UNIVERSITET

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation.

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

2. Deep Conditional Target Densities (DCTD) for accurate regression

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

With DCTD, we aim to combine the benefits of these two approaches.

2. Deep Conditional Target Densities (DCTD) for accurate regression

UPPSALA UNIVERSITET

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

With DCTD, we aim to combine the benefits of these two approaches.

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

With DCTD, we aim to combine the benefits of these two approaches.

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

2. Deep Conditional Target Densities (DCTD) for accurate regression

UPPSALA UNIVERSITET

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

With DCTD, we aim to combine the benefits of these two approaches.

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

UPPSALA UNIVERSITET

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

UPPSALA UNIVERSITET

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Training thus requires the evaluation of $Z(x,\theta)$, we employ importance sampling:

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Training thus requires the evaluation of $Z(x,\theta)$, we employ importance sampling:

$$\begin{aligned}
-\log p(y_i|x_i;\theta) &= \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i,y_i) \\
&= \log\left(\int \frac{e^{f_\theta(x_i,y)}}{q(y)} q(y) dy\right) - f_\theta(x_i,y_i) \\
&\approx \log\left(\frac{1}{M}\sum_{k=1}^{M} \frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).
\end{aligned}$$

UPPSALA UNIVERSITET

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M} \frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M}\frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

We use a proposal distribution $q(y) = q(y|y_i) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(y; y_i, \sigma_l^2)$ that depends on the ground truth target y_i.

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M}\frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

We use a proposal distribution $q(y) = q(y|y_i) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(y;y_i,\sigma_l^2)$ that depends on the ground truth target y_i. The final minimization objective $J(\theta)$ is thus given by:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i,y^{(i,m)})}}{q(y^{(i,m)}|y_i)}\right) - f_\theta(x_i,y_i), \quad \{y^{(i,m)}\}_{m=1}^{M} \sim q(y|y_i).$$

The DCTD model $p(y|x;\theta) = e^{f_\theta(x,y)}/Z(x,\theta)$ is highly flexible and can learn complex target densities directly from data, including multi-modal and asymmetric densities.

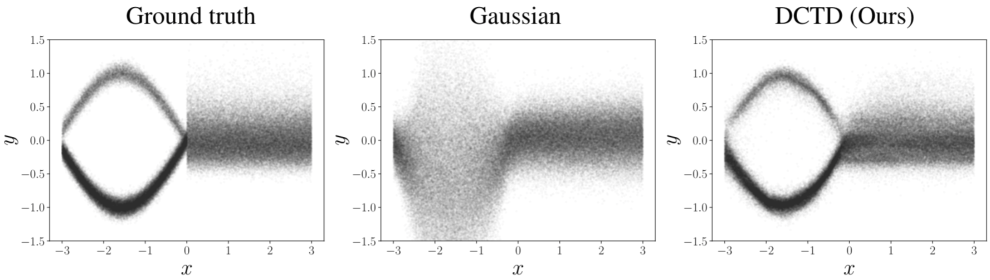


Figure 2: An illustrative 1D regression problem. The training data $\{(x_i, y_i)\}_{i=1}^{2000}$ is generated by the ground truth conditional target density $p(y|x)$.

UPPSALA UNIVERSITET

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

Given an input $x^\star$ at test time, we predict the target $y^\star$ by maximizing $p(y|x^\star;\theta)$:

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

Deep Conditional Target Densities (DCTD): train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

Given an input $x^\star$ at test time, we predict the target $y^\star$ by maximizing $p(y|x^\star;\theta)$:

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

By designing the DNN f_θ to be differentiable w.r.t. targets y, the gradient $\nabla_y f_\theta(x^\star, y)$ can be efficiently evaluated using auto-differentiation. We can thus perform *gradient ascent* to find a local maximum of $f_\theta(x^\star, y)$, starting from an initial estimate $\hat{y}$.

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

$$y^\star = \underset{y}{\text{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\text{argmax}}\, f_\theta(x^\star, y).$$

Deep Conditional Target Densities (DCTD):

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,y)} dy.$$

$$y^\star = \underset{y}{\text{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\text{argmax}}\, f_\theta(x^\star, y).$$

Algorithm 1 Prediction via optimization-based refinement

Input: $x^\star$, $\hat{y}$, T, λ, Ω_1, Ω_2.

1: $y \leftarrow \hat{y}$.
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad$ $\texttt{PrevValue} \leftarrow f_\theta(x^\star, y)$.
4: $\quad$ $y \leftarrow y + \lambda \nabla_y f_\theta(x^\star, y)$.
5: $\quad$ $\texttt{NewValue} \leftarrow f_\theta(x^\star, y)$.
6: $\quad$ **if** $|\texttt{PrevValue} - \texttt{NewValue}| < \Omega_1$ **or** $(\texttt{NewValue} - \texttt{PrevValue}) < \Omega_2$ **then**
7: $\quad\quad$ **Return** y.
8: **Return** y.

Outline

We evaluate DCTD on four diverse computer vision regression tasks: **age estimation**, **head-pose estimation**, **object detection** and **generic visual object tracking**.

3. Experiments

We evaluate DCTD on four diverse computer vision regression tasks: **age estimation**, **head-pose estimation**, **object detection** and **generic visual object tracking**.

DCTD outperforms the confidence-based IoU-Net [4] method for bounding box regression in direct comparisons, both when applied in object detection on the COCO dataset [6], and in the state-of-the-art ATOM [3] visual tracker.

We evaluate DCTD on four diverse computer vision regression tasks: **age estimation**, **head-pose estimation**, **object detection** and **generic visual object tracking**.

DCTD outperforms the confidence-based IoU-Net [4] method for bounding box regression in direct comparisons, both when applied in object detection on the COCO dataset [6], and in the state-of-the-art ATOM [3] visual tracker.

(IoU-Net trains a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict the IoU overlap between a bounding box y and the corresponding ground truth y_i. For training, boxes are sampled around y_i and the difference between predicted and true IoU is minimized. For prediction, an initial estimate $\hat{y}$ is refined using gradient-based maximization of the predicted IoU.)

Age estimation: refinement using DCTD consistently improves MAE (lower is better) for the age predictions outputted by a number of baselines.

+DCTD	Cao et al. [2]	Direct	Gaussian	Laplace	Softmax (CE, L^2)	Softmax (CE, L^2, Var)
	5.47 ± 0.01	4.81 ± 0.02	4.79 ± 0.06	4.85 ± 0.04	4.78 ± 0.05	4.81 ± 0.03
✓	-	**4.65** $\pm$ 0.02	4.66 ± 0.04	4.81 ± 0.04	**4.65** $\pm$ 0.04	4.69 ± 0.03

Head-pose estimation: refinement using DCTD consistently improves the average MAE for Yaw, Pitch and Roll for the predicted pose outputted by our baselines.

+DCTD	Yang et al. [12]	Direct	Gaussian	Laplace	Softmax (CE, L^2)	Softmax (CE, L^2, Var)
	3.60	3.09 ± 0.07	3.12 ± 0.08	3.21 ± 0.06	3.04 ± 0.08	3.15 ± 0.07
✓	-	3.07 ± 0.07	3.11 ± 0.07	3.19 ± 0.06	$\mathbf{3.01} \pm 0.07$	3.11 ± 0.06

Object detection: when applied to refine the Faster-RCNN detections on COCO [6], DCTD both improves the original detections and outperforms the IoU-Net refinement.

Formulation Approach	Direct Faster-RCNN [10]	Gaussian	Laplace	Confidence IoU-Net [4]	Confidence IoU-Net†	**DCTD**
AP (%)	37.2	36.7	37.1	38.3	38.2	**39.1**
AP_{50}(%)	**59.2**	58.7	59.1	58.3	58.4	58.5
AP_{75}(%)	40.3	39.6	40.2	41.4	41.4	**41.8**

Generic visual object tracking: given *any* target object defined by a bounding box in the first frame of a video, estimate its bounding box in all subsequent video frames.

3.2 Generic visual object tracking

Generic visual object tracking: given *any* target object defined by a bounding box in the first frame of a video, estimate its bounding box in all subsequent video frames.

ATOM [3] trains a classifier *online* to first roughly localize the target object in a new frame. Its bounding box is then estimated by using an IoU-Net, trained *offline*, to refine this initial estimate.

Video: https://youtu.be/UP_eLvwskzU

UPPSALA UNIVERSITET

Results: when applied to refine the initial estimate provided by the classifier in ATOM, DCTD outperforms the original method (which uses IoU-Net for refinement). DCTD also outperforms other state-of-the-art trackers.

Dataset	Metric	SiamFC [1]	MDNet [9]	DaSiamRPN [13]	SiamRPN++ [5]	ATOM [3]	ATOM†	**DCTD**
TrackingNet [7]	Precision (%)	53.3	56.5	59.1	**69.4**	64.8	66.7	68.9
	Norm. Prec. (%)	66.6	70.5	73.3	**80.0**	77.1	78.3	79.5
	Success (%)	57.1	60.6	63.8	73.3	70.3	72.1	**73.7**
UAV123 [8]	$OP_{0.50}$ (%)	-	-	73.6	75*	78.9	79.6	**80.1**
	$OP_{0.75}$ (%)	-	-	41.1	56*	55.7	56.0	**59.8**
	AUC (%)	-	52.8	58.4	61.3	65.0	65.0	**66.5**

Qualitative results for DCTD:

https://youtu.be/AAnr0g38UeA

https://youtu.be/JyhgUYpwQ5c

References

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV workshop*, 2016.

[2] W. Cao, V. Mirjalili, and S. Raschka. Rank-consistent ordinal regression for neural networks. *arXiv preprint arXiv:1901.07884*, 2019.

[3] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4660–4669, 2019.

[4] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[5] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[7] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

[8] M. Müller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.

[9] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[10] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[11] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[12] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang. FSA-Net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1087–1096, 2019.

[13] Z. Zhu, Q. Wang, L. Bo, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.

UPPSALA UNIVERSITET

Contact

Fredrik K. Gustafsson, Uppsala University

fredrik.gustafsson@it.uu.se

www.fregu856.com