People Counting by Huber Loss Regression

Jacopo Cavazza[1,2] and Vittorio Murino[1,3]

[1]Pattern Analysis & Computer Vision, Istituto Italiano di Tecnologia, Genova, Italy
[2]Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni, University of Genova, Italy
[3]Dipartimento di Informatica, University of Verona, Italy
firstname.lastname@iit.it

Abstract

In this paper, we address the problem of people counting, proposing a novel approach for regression, based on a new closed-form solution for the Huber loss. Numerically, in a multi-kernel setting, our algorithm adapts to the data in order to iteratively reduce the regression error. We provide a strong theoretical foundation of the method and we validate it in an wide comparison with several techniques for people counting on MALL, UCSD and PETS 2009 benchmark datasets. Globally, we get outstanding results, scoring second and third best lowest errors on MALL and setting the new state-of-the-art on UCSD and PETS 2009.

1. Introduction

The analysis of images is a board research area and automatic scene understanding has a strong impact on social scenarios, e.g. in surveillance/security, event detection and the like. In this context, *people counting*, aims at estimating the number of persons from images in a scene. The applications are various, such as monitoring crowd dynamics in public events or customer profiling. Due to big amount of video-surveillance data, the human control on gatherings is unpractical. On the other hand, automatic people counting from closed-circuit cameras is challenging because of low resolution videos, inter-person occlusions, perspective distortion and light variations [15, 13]. State-of-the-art methods are mainly based on regression, thus learning a map between low level features, computed on single images, and people count. As logical, they avoid explicit pedestrian detection or clustering [6, 21, 23, 14]. In this paper, we face people counting proposing a new algorithm which makes use the Huber loss to boost the reduction of the regression error. Concretely, our contribution is three-fold.

- (i) We solve an open problem in optimization, deducing the analytic closed-form solution for the Huber loss applied in a manifold regularization objective functional.
- (ii) From this theoretical results, we propose HLR, a new algorithmic framework for the Huber loss regression problem. The method has strong theoretical background (convergence and error-reduction statements are proved). In addition to encode different types of features with separated kernels, HLR is able to actively select the data, in order to boost the reduction of the regression error.
- (iii) We validate our framework in several comparative experiments on MALL, UCSD and PETS 2009 benchmark datasets (Figure 1), involving the most effective methods from the literature using different training/testing splits. If on MALL we scored second and third best performance, we set the new state-of-the-art on UCSD and PETS 2009.

Figure 1. An example of frames from the MALL (left), UCSD (center) and PETS 2009 (right) benchmark datasets.

This is the outline of the paper. In Section 2, we briefly overview the literature on people counting and regression. In Section 3, we proposed our new closed-form solution for the Huber loss in a classical manifold regularization framework. Section 4 explains the HLR algorithm, with relative theoretical foundation. Section 5 reports an extensive experimental session with comparisons. Finally, we discuss the results and draw the conclusions in Section 6.

2. Related work

A consolidated taxonomy of people counting approaches identifies three main paradigms [19]: *counting by detection*, *counting by clustering* and *counting by regression*. In counting by detection, a classifier is trained to learn a model for a single person [15], as expected, this type of approaches is sensible to occlusions and deformable part models have been introduced to overcome this issue [16]. Counting by clustering is based on the extraction of coherent motion pattern from the crowd (*e.g.* with a KLT tracker [25]). Finally, in counting by regression, one estimates the number of people from image features without intermediate steps. Usually, a region of interest is detected and the effects

of perspective distortion are removed [21]. The works of [9] and [21] proposed a linear-affine relation between edge pixel number and people count, [23] exploited the mutual occurrences of gray levels, [6] focused edges and texture-based statistics. Several methods have been exploited, like Bayesian regression models [5, 4] or ridge regression [8, 7]. A group of recent papers [30, 20] tried to perform manifold learning to exploit geometric inner configuration of feature space, reducing the amount of needed annotations in a semi-supervised framework.

Regression analysis predicts *target/response* output variables y from *independent* input instances x. In classical machine learning approaches, such relation is learnt from (training) examples. Many ad-hoc techniques have been developed (linear and least-squares regression [11] or Gaussian process regression [26]) and classical algorithms have been successfully applied to regression tasks (neural networks [17] or K-nearest neighbors [27]).

3. Manifold regression with differentiable loss

Let $y \in \mathcal{Y} \subseteq \mathbb{R}$ a scalar target variable and $x \in \mathcal{X} \subseteq \mathbb{R}^d$ an independent input; more precisely, x will encode a feature vector. For any $\alpha = 1, \dots, m$, let $\kappa^\alpha \colon \mathcal{X} \times \mathcal{X} \to [0, +\infty)$ a symmetric and positive semi-definite function (Mercer kernel [29]). Let us define $K(x, x') = \operatorname{diag}(\kappa^1(x, x'), \dots, \kappa^m(x, x')) \in \mathbb{R}^{m \times m}$, where $x, x' \in \mathcal{X}$. Consider $\mathcal{S}_0$ the set of all linear combination of functions $f(x) = \sum_{i=1}^n K(x, x_i) u_i$ for $x_1, \dots, x_n \in \mathcal{X}$ and $u_1, \dots, u_n \in \mathbb{R}^m$; define the norm $\|f\|_K^2 = \sum_{i,j=1}^n u_i K(x_i, x_j) u_j$. The reproducing kernel Hilbert space $\mathcal{S}_K$ (related to K) is $\mathcal{S}_0$ added with all Cauchy sequences limits converging with respect to $\|\cdot\|_K$ [3]. Consider a training set $\mathbf{z}$ made of ℓ pairs $(x_1, y_1), \dots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$ and u additional inputs $x_{\ell+1}, \dots, x_{\ell+u} \in \mathcal{X}$. This formulation is very general: $u = 0$ leads to a fully supervised setting in which all $x_i \in \mathcal{X}$ are provided with targets $y_i \in \mathcal{Y}$. If $u > 0$, we exploit $x_{\ell+1}, \dots, x_{\ell+u}$ to infer feature space geometrical information. Learning unsupervised concepts is both familiar to human brain and useful to improve algorithms generalization [2]. Now, consider the objective functional

$$J_{\lambda,\gamma}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i - c^\top f(x_i)) + \lambda \|f\|_K^2 + \gamma \|f\|_M^2. \quad (1)$$

It consists in three terms. $\frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i - c^\top f(x_i))$ is the empirical risk, depending on the loss V, where $c \in \mathbb{R}^m$ defines our regression map $c^\top f \colon \mathcal{X} \to \mathbb{R}$. $\|f\|_K^2$ is a Tichonov regularizer, scaled by $\lambda > 0$. Parameter $\gamma \geq 0$ weighs the manifold regularization [2] term $\|f\|_M^2 = \sum_{i,j=1}^{u+\ell} f(x_i)^\top M_{ij} f(x_j)$ where $M_{ij} \in \mathbb{R}^{m \times m}$ is a symmetric and positive definite matrix. Thanks to Representer Theorem [24], we have a unique solution $f^\star = \arg\min_{f \in \mathcal{S}_K} J_{\lambda,\gamma}(f)$ given by the formula $f^\star(x) = \sum_{j=1}^{u+\ell} K(x, x_j) w_j$ for some $w_1, \dots, w_{u+\ell} \in \mathbb{R}^m$. Coefficients $\mathbf{w} = [w_1, \dots, w_{u+\ell}]$ define explicitly $f^\star$; moreover optimizing $J_{\lambda,\gamma}$ equals to minimizing $\mathcal{J}_{\lambda,\gamma}(\mathbf{w})$ defined as

$$\frac{1}{\ell} \sum_{i=1}^{\ell} V\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j\right) + \lambda \sum_{j,k=1}^{u+\ell} w_j K(x_j, x_k) w_k + \gamma \sum_{i,j=1}^{u+\ell} \sum_{h,k=1}^{u+\ell} w_h^\top K(x_h, x_i) M_{ij} K(x_j, x_k) w_k. \quad (2)$$

This can be done setting to 0 the derivatives of $\mathcal{J}_{\lambda,\gamma}$ with respect to w_p^η, *i.e.* the η-component of w_p. It provides

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = = V'\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j\right) c \quad (3)$$

for $i = 1, \dots, \ell$; and, when $i = \ell + 1, \dots, u + \ell$,

$$\lambda w_i + \gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = 0. \quad (4)$$

In (3)-(4), fix $V = H_\xi$, the Huber loss [12], defined $H_\xi(y) = y^2/2$ if $|y| \leq \xi$ and $H_\xi(y) = \xi|y| - \xi^2/2$ otherwise. Then,

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = = \begin{cases} -\mu c & \text{if } i \in L_+[\mathbf{z}, \mathbf{w}, \xi] \\ \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j\right) c & \text{if } i \in L_0[\mathbf{z}, \mathbf{w}, \xi] \\ +\mu c & \text{if } i \in L_-[\mathbf{z}, \mathbf{w}, \xi] \\ 0 & i = \ell + 1, \dots, u + \ell, \end{cases} \quad (5)$$

where, for the training set $\mathbf{z}$, any $\mathbf{w}$ and $\xi > 0$,

$$L_+[\mathbf{z}, \mathbf{w}, \xi] = \left\{ i \leq \ell \colon \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j \geq y_i + \xi \right\}, \quad (6)$$

$$L_0[\mathbf{z}, \mathbf{w}, \xi] = \left\{ i \leq \ell \colon \left| \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j - y_i \right| < \xi \right\}, \quad (7)$$

$$L_-[\mathbf{z}, \mathbf{w}, \xi] = \left\{ i \leq \ell \colon \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j \leq y_i - \xi \right\}. \quad (8)$$

Let us interpret numerically the sets (6), (7) and (8). $\{L_-[\mathbf{z}, \mathbf{w}, \xi], L_0[\mathbf{z}, \mathbf{w}, \xi], L_+[\mathbf{z}, \mathbf{w}, \xi]\}$ provides a partition of $\{1, \dots, \ell\}$. Define $\varepsilon_i = \left| \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j - y_i \right|$, the in-sample absolute error involving the i-th labeled element $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ of training set $\mathbf{z}$, and call $\varepsilon_\infty = \max_{i \leq \ell} \varepsilon_i$ the maximum. Thus, $L_0[\mathbf{z}, \mathbf{w}, \xi] = \{i \leq \ell \colon \varepsilon_i < \xi\}$ is the set collecting the indexes of instances which are more effective in strictly reducing absolute errors ε_i below threshold ξ. Instead, in $L_+[\mathbf{z}, \mathbf{w}, \xi]$ and $L_-[\mathbf{z}, \mathbf{w}, \xi]$, target variables are over and under-estimated respectively. Furthermore, when $L_0[\mathbf{z}, \mathbf{w}, \xi] = \{1, \dots, \ell\}$, since $\varepsilon_i < \xi$ for any i, we obtain $\varepsilon_\infty < \xi$. So, if $L_+[\mathbf{z}, \mathbf{w}, \xi]$ and $L_-[\mathbf{z}, \mathbf{w}, \xi]$ are empty, ξ is an upper bound for the absolute error of the whole traning set.

4. The Huber loss regression algorithm

In this section, we will focus on a computational approach to solve equations (5). The main issue pertains to the right-hand part of the labeled systems, since the conditions $L_+[\mathbf{z}, \mathbf{w}, \xi]$, $L_0[\mathbf{x}, \mathbf{w}, \xi]$, and $L_-[\mathbf{z}, \mathbf{w}, \xi]$ involve the final solution. Thus, for Huber loss function, equations (3)–(4) implicitly gives $w_1, \ldots, w_{u+\ell}$. To overcome this drawback, we propose an iterative algorithm to solve the Huber loss regression (HLR) problem. Let preliminary fix the notation.

- $\mathbf{w} \in \mathbb{R}^{m(u+\ell)}$ concatenates $w_1, \ldots, w_{u+\ell} \in \mathbb{R}^m$.
- Vector $\mathbf{y} = [y_1, \ldots, y_\ell]^\top$ and feature matrix $\mathbf{x} \in \mathbb{R}^{(u+\ell)\times d}$ encode target and input variables.
- $\mathbf{M}$ is the block matrix collecting M_{ij} for any i, j.
- Consider $\mathbf{K}[\mathbf{x}]$, the Gram matrix defined by $K_{ij}[\mathbf{x}] = K(x_i, x_j)$ for $i, j = 1, \ldots, u+\ell$.
- For $c = [c^1, \ldots, c^m]^\top$, denoting I_n the $n \times n$ identity matrix and $\otimes$ the Kronecker tensor product, let $\mathbf{C} = I_{u+\ell} \otimes c^\top$ and $\mathbf{C}^\star = I_{u+\ell} \otimes c$.
- Let $\mathbf{Q}[\mathbf{z}, \mathbf{w}, \xi] \in \mathbb{R}^{m(u+\ell)\times m(u+\ell)}$ the block matrix, $Q_{ij}[\mathbf{z}, \mathbf{w}, \xi] = \mathbb{1}(i \in L_0[\mathbf{z}, \mathbf{w}, \xi]) K_{ij}[\mathbf{x}]$ for any i, j.
- $\mathbf{b}[\mathbf{z}, \mathbf{w}, \xi] = [b_1[\mathbf{z}, \mathbf{w}, \xi], \ldots, b_{u+\ell}[\mathbf{z}, \mathbf{w}, \xi]]^\top$, where

$$b_i[\mathbf{z}, \mathbf{w}, \xi] = \begin{cases} -\xi & \text{if } i \in L_+[\mathbf{z}, \mathbf{w}, \xi] \\ y_i & \text{if } i \in L_0[\mathbf{z}, \mathbf{w}, \xi] \\ +\xi & \text{if } i \in L_-[\mathbf{z}, \mathbf{w}, \xi] \\ 0 & \text{if } i = \ell+1, \ldots, u+\ell. \end{cases} \tag{9}$$

Since we can not find explicitly the sets L_+, L_- and L_0 in equation (5) if we have not already the solution $\mathbf{w}$, the main idea is an iterative scheme in which L_0, L_+ and L_- are firstly computed for an initial value $\mathbf{w}^{\text{old}}$, which is updated to $\mathbf{w}^{\text{new}}$, once linear system

$$\begin{aligned} &\left(\mathbf{C}^\star \mathbf{C}\mathbf{Q}[\mathbf{z}, \mathbf{w}^{\text{old}}, \xi] + 2\ell\lambda \mathbf{M}\mathbf{K}[\mathbf{x}]\right) \mathbf{w}^{\text{new}} + \\ &+ 2\ell\gamma \mathbf{w}^{\text{new}} = \mathbf{C}^\star \mathbf{b}[\mathbf{z}, \mathbf{w}^{\text{old}}, \xi], \end{aligned} \tag{10}$$

is solved for a certain $\xi > 0$. Denoting with $\mathbf{y}_0$ the result of appending a $u \times 1$ zeros vector to $\mathbf{y}$, we suggest to initialize $\mathbf{w}$ as $\mathbf{w}^{(0)}$ provided by

$$\left(\mathbf{C}^\star \mathbf{C} + \ell\gamma \mathbf{M}\right) \mathbf{K}[\mathbf{x}] \mathbf{w}^{(0)} + \ell\lambda \mathbf{w}^{(0)} = \mathbf{C}^\star \mathbf{y}_0, \tag{11}$$

which is exactly the formulas (5) in the case $\xi = +\infty$.

The pseudocode for our method is described in Algorithm 1. Here, the apex (τ) highlights the updating of any variable to the τ-th iteration, $\tau = 0, 1, \ldots, T$. In order to provide an effective convergence of the scheme, consider

$$\xi^{(\tau)} = \max_{i \le \ell} \left| \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j^{(\tau)} - y_i \right|, \tag{12}$$

Algorithm 1: HLR algorithm pseudocode

Input: $\mathbf{x}$ feature design matrix, $\mathbf{y}$ labels vector, $\mathbf{M}$ block matrix of operators $\{M_{ij}\}$, $\lambda > 0$ Tichonov regularizing parameter, $\gamma \ge 0$ manifold regularization parameter, $\Delta\xi$ updating rate, T maximum number of iterations.
Output: Coefficients vector $\mathbf{w}^\star \in \mathbb{R}^{m(u+\ell)}$.
begin
1. Find $\mathbf{w}^{(0)}$ solving equation (10).
2. $\mathbf{z}^{(0)} := \mathbf{z}$ and compute $\xi^{(0)}$ as in (12).
   for $\tau = 1, \ldots, T$ **do**
3. $\xi^{(\tau-1/2)} := \xi^{(\tau-1)} - \Delta\xi$.
4. Solve (11) in the unknown $\mathbf{w}^{\text{new}} := \mathbf{w}^{(\tau-1/2)}$, with $\mathbf{w}^{\text{old}} := \mathbf{w}^{(\tau-1)}$, $\mathbf{z} := \mathbf{z}^{(\tau-1)}$ and $\xi := \xi^{(\tau-1/2)}$.
5. Compute the sets L_0, L_+ and L_- on $[\mathbf{z}^{(\tau-1)}, \mathbf{w}^{(\tau-1/2)}, \xi^{(\tau-1/2)}]$.
   if $L_0[\mathbf{z}^{(\tau-1)}, \mathbf{w}^{(\tau-1/2)}, \xi^{(\tau-1/2)}]$ *is empty* **then**
6. **return** $\mathbf{w}^\star := \mathbf{w}^{(\tau-1)}$.
   else
7. $\mathbf{z}^{(\tau)} := \{(x_i, y_i) : i \in L_0\} \cup \{x_j : j \in L_+ \cup L_- \text{ or } j = \ell+1, \ldots, u+\ell\}$.
8. Obtain $\mathbf{w}^{(\tau)}$ permuting $\mathbf{w}^{(\tau-1)}$: first the elements $w_j^{(\tau-1)}$ with $j \in L_0$, then $j \in L_+ \cup L_- \cup \{\ell+1, \ldots, \ell+u\}$.
9. Compute $\xi^{(\tau)}$ using relation (12).
   return $\mathbf{w}^\star := \mathbf{w}^{(\tau)}$

the maximum of in-sample absolute error at iteration τ. In Algorithm 1, it is crucial to introduce the intermediate update $\xi^{(\tau-1/2)}$, systematically reducing of $\Delta\xi > 0$ the error $\xi^{(\tau-1)}$ resulting from the previous iteration. This passage is the key-point lemma for the following propositions, which consist in the theoretical foundation of HLR.

Proposition 1. *For any $\tau = 0, 1, \ldots, T$, the coefficients $\mathbf{w}^{(\tau)}$ satisfy system (5) with $\xi = \xi^{(\tau)}$.*

Proof. It is enough to show that, for any $\tau = 0, 1, \ldots, T$, we get $L_0[\mathbf{z}^{(\tau)}, \mathbf{w}^{(\tau)}, \xi^{(\tau)}] = \{1, \ldots, \ell\}$. Let's go by induction. For $\tau = 0$, as afore mentioned, $\mathbf{w}^{(0)}$ solves (5) for $\xi = +\infty$ and $\xi^{(0)}$ is a posteriori computed. So $L_0[\mathbf{z}, \mathbf{w}^{(0)}, \xi^{(0)}] = \{1, \ldots, \ell\}$ tautologically. Let's assume $L_0[\mathbf{z}^{(\tau-1)}, \mathbf{w}^{(\tau-1)}, \xi^{(\tau-1)}] = \{1, \ldots, \ell\}$ and show that the same relation holds also at iteration τ. Once computed $\mathbf{w}^{(\tau-1/2)}$, we gather the response variable only for x_i which $\left|\sum_{j=1}^{u+\ell} c^\top K_{ij}[\mathbf{x}] w_j^{(\tau-1/2)} - y_i\right| \le \xi^{(\tau-1/2)}$. So, we have $\left|\sum_{j=1}^{u+\ell} c^\top K_{ij}[\mathbf{x}] w_j^{(\tau)} - y_i\right| \le \xi^{(\tau-1/2)}$ because we only have sorted $\mathbf{z}^{(\tau-1)}$ and $\mathbf{w}^{(\tau-1)}$ in order to have $\mathbf{z}^{(\tau)}$ and $\mathbf{w}^{(\tau)}$ preserving such property. Since $\xi^{(\tau)}$ is the maximum of a finite set of elements all bounded by $\xi^{(\tau-1/2)}$, we can conclude

$$\xi^{(\tau)} = \max_{i=1,\ldots,\ell} \left| \sum_{j=1}^{u+\ell} c^\top K_{ij}[\mathbf{x}] w_j^{(\tau)} - y_i \right| \le \xi^{(\tau-1/2)}. \tag{13}$$

From the previous relation we immediately obtain $L_0[\mathbf{z}^{(\tau)}, \mathbf{w}^{(\tau)}, \xi^{(\tau)}] = \{1, \ldots, \ell\}$. □

Proposition 2. *The sequence $\xi^{(0)}, \xi^{(1)}, \ldots, \xi^{(T)}$ is monotonically strictly decreasing.*

Proof. In order to prove monotonicity, we fix $\tau = 1, \dots, T$ and show $\xi^{(\tau)} < \xi^{(\tau-1)}$. We directly use relation (13), $\xi^{(\tau)} \leq \xi^{(\tau-1/2)}$. By definition, $\xi^{(\tau-1/2)} = \xi^{(\tau-1)} - \Delta\xi$. Since $\Delta\xi > 0$, then $\xi^{(\tau)} \leq \xi^{(\tau-1)} - \Delta\xi < \xi^{(\tau-1)}$ that is the thesis. □

Differently from the current literature (see [22]), in which approximated solution are found for other Huber loss based optimization frameworks, Proposition 1 proves that *at each iteration, equation* (5) *is solved for a different value of* ξ. Proposition 2 certifies the error reduction inside the training set and the consequent improvement of performance. A crucial aspect of HLR is that automatically performs an active selection of which instances $x_1, \dots, x_\ell$ have to be paired with the corresponding responses $y_1, \dots, y_\ell$, in order to reduce the error. Precisely, we discard those targets y_i which violate

$$\left| \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j^{(\tau-1/2)} - y_i \right| < \xi^{(\tau-1/2)}. \qquad (14)$$

If i-th training istance does not respect (14) and $(x_i, y_i) \in \mathbf{z}^{(\tau-1)}$, then $x_i \in \mathbf{z}^{(\tau)}$ and y_i is removed. Overall, HLR automatically adapts ξ to maximize the amount of data exploitable to improve regression model. The responses inconsistent with this improvement are discarded and corresponding inputs are only used to infer geometrical information about feature space. Such adaptive behavior is a unique characteristic of HLR, which is thus able to automatically analyze data, extracting as much information as possible to improve regression. As a final theoretical remark, once $w_1^\star, \dots, w_{u+\ell}^\star$ are computed, $h(v) = \sum_{j=1}^{u+\ell} c^\top K(v, x_j) w_j^\star$ is the prediction for the new test sample $v \in \mathbb{R}^d$.

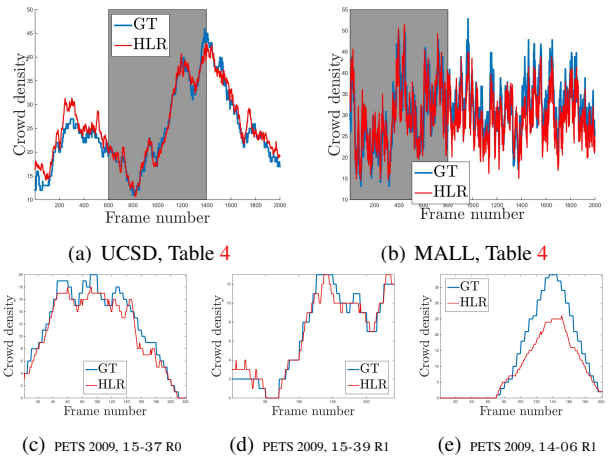
5. HLR validation on people counting task

Three benchmark datasets have been used to test the performance of our Huber loss regression method in people counting in different scenarios. They are MALL [8], UCSD [6] and PETS 2009 [4]. We adopted the training/testing splits usually adopted from the literature. Indeed, on MALL, the first 800 frames are used for training, while the remaining for testing [7]. Experiments on UCSD are usually trained on frames $601 \div 1400$ [8]; for PETS 2009, please refer to Table 1.

Test	Region(s)	Train
13-57 (221)	R0,R1,R2	13-59,14-03 (1308)
13-59 (241)	R0,R1,R2	13-57,14-03 (1268)
14-06 (201)	R1,R2	13-57,13-59,14-03 (1750)
14-17 (91)	R1	13-57,13-59,14-03 (1750)

Table 1. For each PETS 2009 sequence, the region(s) of interest and the training/testing sets are shown (also in brackets the number of frames involved). We are reproducing the setting of [4] for the sake of comparison.

For the sake of a fair comparison, we mimic the same conditions of the other methods in the literature, employing precomputed features encoding the foreground area, pedestrian edges and texture [6]. The ground truth annotations of the number of people in each frame and all the data were retrieved from public repositories on the Internet[1]. In our framework, and each category of features is thus encoded with a separate polynomial kernel (thus $m = 3$). We fixed $c = [1/3, 1/3, 1/3]^\top$ and M_{ij} is the sum of between-view operator from [24] and normalized graph laplacian related to $K_{ij}[\mathbf{x}]$. We have a small set of parameters to be tuned: the number of iterations T, the regularizing parameters λ, γ, and the rate $\Delta\xi$. Actually, a good rule of the thumb is $T = 3$, $\lambda = 10^{-4}$ and $\gamma = 10^{-5}$; $\Delta\xi = 0.01$; in general, only few variations from this setting are required (Table 6).



(a) UCSD, Table 4 (b) MALL, Table 4

(c) PETS 2009, 15-37 R0 (d) PETS 2009, 15-39 R1 (e) PETS 2009, 14-06 R1

Figure 2. Qualitative results. Blue: ground truth; red: HLR prediction. In 2(a) and 2(b), the gray area highlights the training set.

Method	UCSD			MALL		
	MAE	MSE	MRE	MAE	MSE	MRE
LSSVR[10]	2.20(4)	7.69(4)	**0.107(3)**	3.51(4)	18.20(5)	0.108(4)
KRR[1]	**2.16(3)**	**7.45(3)**	**0.107(3)**	3.51(4)	18.18(4)	0.108(4)
RFR[18]	2.42(8)	8.47(8)	0.116(8)	3.91(9)	21.50(8)	0.121(9)
GPR[6]	2.24(5)	7.97(6)	0.112(7)	3.72(7)	20.10(7)	0.115(7)
RR[28]	2.25(6)	7.82(5)	0.110(6)	3.59(6)	19.00(6)	0.110(6)
CA-RR[7]	**2.07(2)**	**6.86(2)**	**0.102(2)**	**3.43(3)**	**17.70 (3)**	**0.105(3)**
MLR[31]	2.60(9)	10.10(9)	0.125(9)	3.90(8)	23.90(9)	0.120(8)
MORR[8]	2.29 (7)	8.08(7)	0.109(5)	**3.15(1)**	**15.70(1)**	**0.099(1)**
HLR	**1.99 (1)**	**6.00(1)**	**0.093(1)**	**3.36(2)**	**16.42(2)**	**0.104(2)**

Table 4. MAE, MSE and MRE comparison of HLR with the results showed in [8] and [7] involving least square support vector regression LSSVR, kernel ridge regression KRR, random forest regression RFR, Gaussian process regression GPR, ridge regression RR with its cumulative attribute CA-RR variant, multiple localized regressors MLR and multiple output regression MORR applied to people counting.

Figure 2 shows some qualitative results, overlapping the profile of the ground truth people number with our predicted evaluations. For the quantitative performance evaluation, we used three classical error measures. Let $y_1, \dots, y_n$ the target variables in the testing set and $\widehat{y}_1, \dots, \widehat{y}_n$ the corresponding predicted outputs. We will define the mean absolute error as $\text{MAE} = \frac{1}{n} \sum_i |y_i - \widehat{y}_i|$, the mean squared error

[1] http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html for MALL; http://visal.cs.cityu.edu.hk/downloads/ for UCSD and PETS 2009.

Method	UCSD		MALL	
	MAE	MRE	MAE	MRE
GPR	**1.46(2)**	**6.23(2)**	**2.58(1)**	**8.34(1)**
Lin	**1.56(3)**	**6.48(3)**	**2.58(1)**	**8.52(2)**
Knn(1)	2.89(6)	10.75(8)	3.45(8)	11.22(8)
Knn(2)	2.77(5)	10.20(5)	3.05(6)	9.94(6)
Knn(4)	2.72(4)	9.63(4)	2.89(4)	9.28(5)
Knn(8)	2.90(7)	10.20(5)	2.92(5)	9.20(4)
Knn(16)	3.12(8)	10.56(7)	3.25(7)	9.96(7)
Knn(32)	3.76(9)	12.37(9)	4.19(9)	12.51(9)
NN(4)	8.13(11)	33.08(11)	26.06(13)	87.83(13)
NN(8)	9.15(12)	43.08(12)	13.02(11)	43.40(11)
NN(16)	4.36(10)	19.26(10)	16.70(12)	57.61(12)
NN(32)	11.70(13)	54.86(13)	12.18(10)	40.32(10)
HLR	**1.23(1)**	**5.43 (1)**	**2.63 (3)**	**8.74 (3)**

Table 2. Beside Knn and NN, the respective value of K and the number of neurons in the hidden layer are reported. Since talking of errors, lower is better; the ranking in in brakets and winning values are in bold. Following [27], all the frames were divided into 5 equally spaced subset in which training subsets were fixed.

Sequence	Method	total		right-moving		left-moving	
		MAE	MSE	MAE	MSE	MAE	MSE
13-57 R0	GPR	2.308(2)	8.362(2)	0.249(2)	0.339(2)	2.475(2)	8.955(2)
	HLR	**2.290(1)**	**8.118(1)**	**0.204(1)**	**0.204(1)**	**2.385(1)**	**8.719(1)**
13-57 R1	GPR	1.697(2)	5.000(2)	0.100(2)	0.100(2)	1.643(2)	4.720(2)
	HLR	**1.330(1)**	**3.005(1)**	**0.059(1)**	**0.059(1)**	**1.290(1)**	2.919(**1**)
13-57 R2	GPR	1.072(2)	1.796(2)	0.235(2)	0.317(2)	0.842(2)	1.484(2)
	HLR	**0.819(1)**	**1.253(1)**	**0.081(1)**	**0.081(1)**	**0.756(1)**	**1.190(1)**
13-59 R0	GPR	1.647(2)	4.087(2)	1.668(2)	4.158(2)	0.154(2)	0.154(2)
	HLR	**1.560(1)**	**3.320(1)**	**1.639(1)**	**3.589(1)**	**0.137(1)**	**0.137(1)**
13-59 R1	GPR	0.685(2)	1.116(2)	0.589(2)	0.871(2)	**0.095(1)**	**0.095(1)**
	HLR	**0.622(1)**	**0.855(1)**	**0.481(1)**	**0.689(1)**	0.166(2)	0.166(2)
13-59 R2	GPR	1.282(2)	**2.577(1)**	1.291(2)	2.436(2)	**0.066(1)**	**0.066(1)**
	HLR	**1.253(1)**	2.747(2)	**1.195(1)**	**2.274(1)**	0.141(2)	0.141(2)
14-06 R1	GPR	4.328(2)	44.159(2)	4.338(2)	44.159(2)	0.005(2)	0.005(2)
	HLR	**4.299(1)**	**43.383(1)**	**4.299(1)**	**43.383(1)**	**0.000(1)**	**0.000(1)**
14-06 R2	GPR	3.139(2)	26.035(2)	3.144(2)	26.129(2)	0.020(2)	0.020(2)
	HLR	**2.995(1)**	**23.970(1)**	**3.015(1)**	**24.289(1)**	**0.020(1)**	**0.020(1)**
14-17 R1	GPR	0.604(2)	1.220(2)	0.604(2)	**1.198(1)**	**0.000(1)**	**0.000(1)**
	HLR	**0.593(1)**	**1.209(1)**	**0.593(1)**	1.209(2)	**0.000(1)**	**0.000(1)**

Table 3. Comparison with [4] and the associated ranking. In some cases we have 0.000 error values: this occurs when in the current splits no people are moving in the direction currently analyzed and the regression method correctly predicts all 0-response values.

as $\text{MSE} = \frac{1}{n}\sum_i (y_i - \widehat{y}_i)^2$, and the mean relative error as $\text{MRE} = \frac{1}{n}\sum_i \frac{|y_i - \widehat{y}_i|}{y_i}$. MRE is a very reliable metrics since it quantifies the differences between the exact and the predicted values, taking into account their characteristic size. Anyway, it diverges when true values approach zero (this justifies some grows in MRE tables); in this cases MAE and MSE are numerically more stable.

Experiment	MALL	UCSD	PETS2009
Table 2	84.00%	89.00%	
Table 4	99.25%	99.50%	
Table 3 13-57 R0			99.62%
Table 3 13-57 R1			99.54%
Table 3 13-57 R2			99.69%
Table 3 13-59 R0			99.21%
Table 3 13-59 R1			99.68%
Table 3 13-59 R2			99.53%
Table 3 14-06 R1			99.77%
Table 3 14-06 R2			99.77%
Table 3 14-17 R1			99.83%

Table 5. Percentages of inputs x which our algorithm actively kept paired with corresponding target y. For example, for MALL dataset in the conditions of Table 4, 99.25% means that, from the starting 800 input-target pairs, $\ell = 794$ are preserved by HLR and only $u = 6$ inputs are isolated. Notice that, when training and testing splits change from Table 4 to Table 2, the percentages decrease of about 5% (MALL) and 10% (UCSD). Percentages related to PETS 2009 are high since we test on a sequence which is different from the ones used in training.

A recent paper [27] provides an interesting evaluation and survey about the most useful features and algorithms in people counting. In addition to theoretical dissertations, many experiments have been reported. Using same classes of features, in Table 2 we integrate Huber Loss Regression (HLR) in the comparison between Gaussian Process Regression (GPR), regularized linear regression (Lin), K-nearest neighbors (Knn) and neural networks (NN) methods. Using same training/testing splits as in [27], we scored top three error on MALL, while we set state-of-the art in UCSD considerably reducing MRE. We also tested UCSD and MALL datasets in the same conditions as [8] and [7] (see Table 4). Moving to PETS 2009, we reproduce the same experimental conditions as in [4]. Motion segmentation allows to divide the right-moving pedestrians from the others moving in the opposite direction. Total people count has been obtained summing the partial results. Table 3 shows a comparison of Huber loss vs. Gaussian Process Regression (GPR) in this setting reported in Table 1. Performances are sometimes substantially improved, see sequence $13-57$, regions R1 and R2. Finally, it is interesting to check whether or not multiple kernels approach improves performance, *i.e.* instead of $m = 3$, what happens if all the features are grouped together in a unique kernel ($m = 1$)? On MALL dataset in the same condition as in Table 4, if we set $m = 1$, MAE raises from 3.36 to 3.90, MSE from 16.42 to 21.42, and MRE from 0.104 to 0.192. Globally, the semantical information extracted pooling together all the features is poorer than our class-specific encoding, where separated kernels preserve intrinsic differences.

6. Discussion and conclusions

In this work, we deduced a closed-form solution for the Huber loss in a manifold regularization framework. Methodologically, we presented HLR, a new algorithm with remarkable properties. First, self-tuning of Huber loss threshold ξ. Second, the active learning approach in which the algorithm adapts itself to data, while improving model learning (Table 5). Third, the efficiency of HLR whose computational cost is $O((T+1)m^2(u+\ell)^2))$. Thus, to reproduce results from Table 4, our system[2] took 6.5 seconds for training and 0.36 seconds for testing (on MALL) and 5.6 for training and 0.5 for testing (on UCSD). In addition, the default choice for the parameters T, λ, γ and $\Delta\xi$ needs a few tuning (Table 6).

In the people counting application, HLR is extremely performing in comparison with other approaches. From a

[2]The method is implemented in MATLAB, on a Intel(R) Xeon(R) CPU X5650 @2.67 GHz ×2 cores, 12 GB RAM.

	Table 2		Table 4		Table 3
	UCSD	MALL	UCSD	MALL	PETS 2009
T	0	3	4	3	3
λ	10^{-4}	10^{-4}	10^{-10}	10^{-5}	10^{-5}
γ	10^{-5}	10^{-5}	10^{-11}	10^{-6}	10^{-6}
$\Delta\xi$	0.10	0.15	0.05	0.10	0.10

Table 6. Parameters T, λ, γ and $\Delta\xi$ used in 2, Tables 4 and 3. $T = 0$ means that the initialization (10) only has been performed.

qualitative point of view, our predicted values show a trend very similar to the ground truth data (see Fig. 2). Examining numerical results, in the comparison with [4], HLR sets the new state-of-the art on PETS 2009 in 46 experiments over 54 (loosing 6 comparisons with GPR and having a tie in the remaining two cases). From Table 4 and 2, on MALL we record a remarkable second and third overall placement, with very small deviations from the winner method and exploiting about 10% less of the data (Table 5). Finally, in both experiments on UCSD dataset, HLR sets the state-of-the-art in people counting.

Globally, HLR turns out to be very effective for people counting in all the different tested scenarios. Future perspective are essentially focused on applying Huber loss function to classification problems.

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. In *CVPR*, 2007. 4

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006. 2

[3] C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and mercer theorem. *Anal. Appl. (Singap.)*, 4:377 – 408, 2006. 2

[4] A. Chan, M. Morrow, and N. Vasconcelos. Analysis of crowded scenes using holistic properties. In *CVPRw*, 2009. 2, 4, 5, 6

[5] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *TIP*, 21(4):2160–2177, 2012. 2

[6] A. B. Chan, J. S. Zhang, and L. N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *CVPR*, 2008. 1, 2, 4

[7] K. Chen, S. Gong, T. Xiang, and C. Change Loy. Cumulative attribute space for age and crowd density estimation. In *CVPR*, 2013. 2, 4, 5

[8] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012. 2, 4, 5

[9] A. C. Davies, J. H. Yin, and S. A. Velastin. Crowd monitoring using image processing. *Electron Commun Eng*, 7:37–47, 1995. 2

[10] T. V. Gestel, J. A. K. Suykens, B. D. Moor, and J. Vandewalle. Automatic relevance determination for least squares support vector machines classifiers. In *ESANN*, 2001. 4

[11] F. Harrell. *Regression Modeling Strategies: Applications to Linear Models, Logistic Regression, and Survival Analysis*. Graduate Texts in Mathematics. Springer, 2001. 2

[12] P. J. Huber. Robust estimation of a location parameter. *Ann. of Math. Stat.*, 35(1), 1964. 2

[13] T. K. Kim and R. Cipolla. Multiple classifier boosting for perceptual co-clustering of images and visual features. In *Adv. Neural Inf. Process. Syst.*, pages 841–848, 2009. 1

[14] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, 2006. 1

[15] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005. 1

[16] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *ICPR*, 2008. 1

[17] N.-N. Li, J. Song, R.-Y. Zhou, and J.-H. Gu. A people-counting system based on bp neural network. In *Conference on FSKD*, 2007. 2

[18] A. Liaw and M. Wiener. Classification and regression by randomforest. *R Journal*, 2(3):18–22, 2002. 4

[19] C. Loy, K. Chen, S. Gong, and T. Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, volume 11 of *The International Series in Video Computing*, pages 347–382. 2013. 1

[20] C. C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *ICCV*, 2013. 2

[21] R. Ma, L. Li, W. Huang, and Q. Tian. On pixel count based crowd density estimation for visual surveillance. In *CCIS*, 2004. 1, 2

[22] O. Mangasarian and D. R. Musicant. Robust linear and support vector regression. *TPAMI*, 22, 2000. 4

[23] A. Marana, S. Velastin, L. Costa, and R. Lotufo. Estimation of crowd density using image processing. In *Image Processing for Security Applications*, 1997. 1, 2

[24] H. Q. Minh, L. Bazzani, and V. Murino. A unifying framework for vector-valued manifold regularization and multi-view learning. In *ICML*, 2013. 2, 4

[25] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, 2006. 1

[26] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. 2

[27] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. An evaluation of crowd counting methods, features and regression models. *CVIU*, 130:1–17, 2015. 2, 5

[28] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998. 4

[29] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. 2002. 2

[30] B. Tan, J. Zhang, and L. Wang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 44:2297 – 2304, 2011. 2

[31] X. Wu, G. Liang, K. K. Lee, and Y. Xu. Crowd density estimation using texture analysis and learning. In *ROBIO*, 2006. 4