Discriminative Sparse Neighbor Approximation for Imbalanced Learning

Chen Huang, Chen Change Loy, *Member, IEEE,* and Xiaoou Tang, *Fellow, IEEE*

***Abstract*—Data imbalance is common in many vision tasks where one or more classes are rare. Without addressing this issue conventional methods tend to be biased toward the majority class with poor predictive accuracy for the minority class. These methods further deteriorate on small, imbalanced data that has a large degree of class overlap. In this study, we propose a novel discriminative sparse neighbor approximation (DSNA) method to ameliorate the effect of class-imbalance during prediction. Specifically, given a test sample, we first traverse it through a cost-sensitive decision forest to collect a good subset of training examples in its local neighborhood. Then we generate from this subset several class-discriminating but overlapping clusters and model each as an affine subspace. From these subspaces, the proposed DSNA iteratively seeks an optimal approximation of the test sample and outputs an unbiased prediction. We show that our method not only effectively mitigates the imbalance issue, but also allows the prediction to extrapolate to unseen data. The latter capability is crucial for achieving accurate prediction on small dataset with limited samples. The proposed imbalanced learning method can be applied to both classification and regression tasks at a wide range of imbalance levels. It significantly outperforms the state-of-the-art methods that do not possess an imbalance handling mechanism, and is found to perform comparably or even better than recent deep learning methods by using hand-crafted features only.**

***Index Terms*—Imbalanced learning, decision forest, discriminative sparse neighbor approximation, data extrapolation.**

I. Introduction

DATA imbalance exists in many vision tasks ranging from low-level edge detection [1] to high-level facial age estimation [2] and head pose estimation [3]. For instance, in age estimation, there are often many more images of the youth than the old on the widely used FG-NET [2] and MORPH [4] datasets. In edge detection, various image edge structures [5] obey a power-law distribution, as shown in Figure 1. Without handling this imbalance issue conventional vision algorithms have a strong learning bias towards the majority class with poor predictive accuracy for the minority class, usually of equal or more interest (e.g., rare edges may convey the most important semantic information about natural images).

The insufficient learning for the minority class is due to the complete lack of representation by a limited number of or even no examples, especially in the presence of small datasets. For instance, FG-NET age dataset has 1002 images in total with only 8 images over 60 years old. Certain age classes of 60+ ages have no images at all. This reveals a bigger challenge on unseen data extrapolation from the few minority class samples that usually have high variability. Even worse, the small imbalanced datasets can be accompanied by the class overlap problem. We plot the PCA and Fisher embeddings of FG-NET in Figure 2 to illustrate this problem. From the figure, it is evident that training a robust classifier or regressor capable of handling old ages is indeed a hard problem: (i) the corresponding minority class (blue star) contains insufficient samples for learning, (ii) these samples have high degree of variability which is hard to model, (iii) there is a severe class overlap between the rare samples and those from majority classes, further compounding the learning difficulty. Consequently, if we look into the local neighborhood of a minority class sample, it is very likely to be dominated by the majority class samples. Its weak local boundary would

Chen Huang is with the Robotics Institute of Carnegie Mellon University, Pittsburgh, USA (e-mail: yach23@gmail.com). Chen Change Loy and Xiaoou Tang are with the Department of Information Engineering, The Chinese University of Hong Kong (e-mail: {ccloy,xtang}@ie.cuhk.edu.hk).

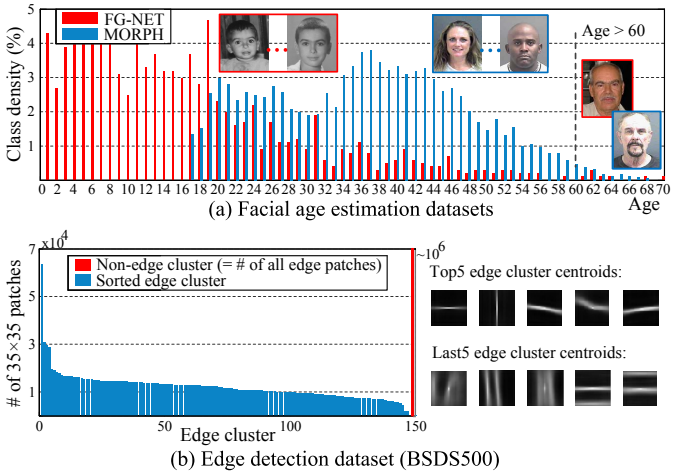


Fig. 1. Data imbalance in (a) age estimation and (b) edge detection. In existing age datasets, there are usually more images of the youth than the old (>60 years old). While in image edge datasets, the observed edge patches typically obey an imbalanced power-law distribution. Moreover, the numbers of collected edge and non-edge patches are usually equal, which leads to a severe imbalance between each edge class and the non-edge class.

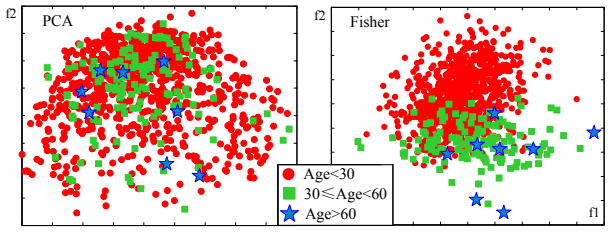


Fig. 2. 2D distributions of age data after PCA and LDA on FG-NET dataset. They show the dataset is not only small and class-imbalanced, but also has the class overlap issue. Such issues just compound the learning difficulty.

bias the prediction towards the majority class.

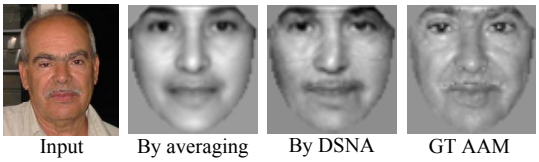
There are three common approaches to counter the negative impact of data imbalance: resampling [6], [7], cost-sensitive learning [8]–[10] and ensemble learning [11], [12]. Resampling approaches aim to make class priors equal by under-sampling the majority class or over-sampling the minority class (or both [6]). These methods can easily eliminate valuable information or introduce noise respectively. Cost-sensitive learning is often reported to outperform random re-sampling by adjusting misclassification costs, however the true costs are often unknown. An effective technique for further improvement is to resort to ensemble learning [13]. Chen *et al.* [11] combined bagging and weighted decision trees to generate a re-weighted version of random forest. We show in our experiments that the aforementioned strategies fall short in handling complex imbalanced data. Beyond empirical performance, the above approaches have two common drawbacks: 1) They are designed for either classification [6]–[8], [10]–[12] or regression [9] without a universal solution to both. 2) They have a limited ability to account for unseen appearances or extrapolate novel labels on the observed space. This is critical in the typical case of small imbalanced datasets where the minority class is under-represented by an excessively reduced number of or even no samples/labels.

In this paper we address the problems of data imbalance *and* unseen data extrapolation using a data-driven approach. The approach can be applied to *both* classification and regression scenarios. The key idea of our approach is intuitive – given a test sample, we first locate for it a 'safe' local neighborhood. This local neighborhood is formed by training samples, which are carefully mined so as to provide a relatively large coverage of minority class samples in the full space. But overall, this subspace is tight and is less probable to be invaded by imposter samples[1]. We show that this 'safe' local neighborhood can be constructed via a cost-sensitive decision forest. It is worth noting that the local neighborhood may still be overwhelmed by majority classes especially when the minority ones are absolutely rare. Thus prediction by simple voting or averaging within it could easily smooth out the minority class samples. To this end, we further partition the local neighborhood into several discriminative but soft clusters. This process provides purer clusters eliminating the undesired class domination.

We propose a new Discriminative Sparse Neighbor Approximation (DSNA) method that allows robust prediction from our formed clusters. The clusters are modelled as affine subspaces to account for unseen appearances in a similar spirit of [14]. The core of DSNA is a new cost function and a joint optimization approach to iteratively determine the best affine subspace that best approximates the test sample with the help of associated sparse neighbors. From the found neighbors and their approximating coefficients, we can transfer their labels to achieve a robust prediction under the class-imbalanced scenario. Figure 3 illustrates the effectiveness of DSNA in an age estimation example.

In summary, the main contributions of this paper are:

[1]An imposter sample is defined as the one from a different class w.r.t. the test sample.



Input GT age 69 | By averaging Est 46 | By DSNA Est **58** | GT AAM representation

Fig. 3. A visualization of age estimation result when neither the testing appearance nor age label is observed during training. Averaging provides a crude way of estimating the face appearance (AAM, Active Appearance Model) from the nearest neighbors in the training set. The proposed discriminative sparse neighbor approximation (DSNA) provides a more robust estimation, thus an age value closer to the Ground Truth (GT).

- A new discriminative sparse neighbor approximation (DSNA) method is proposed for unbiased predictions with preserved discriminative and extrapolative ability given class-imbalanced data.
- To facilitate robust predictions via DSNA, we formulate an effective way of constructing a safe local neighorhood through a cost-sensitive decision forest framework.
- The proposed method is applied to the vision tasks of age estimation (regression), head pose estimation (regression) and edge detection (classification) with varying degree of data imbalance and amount of data. It advances the state-of-the-art, sometimes considerably, across all tasks especially on highly imbalanced ones. It comes at only modest extra computational burden, showing its potential as a fast and general framework for imbalanced learning. Our results are particularly impressive when favorably compared to deep learning methods [15]–[22] as our method uses no deep features, but introduces the imbalance handling mechanism absent in these deep models.

The rest of the paper is organized as follows. Section II briefly reviews related work on imbalanced learning and the considered vision tasks. Section III details the major components of the proposed method. Section IV presents the results on imbalanced vision and generic datasets, as well as the runtime analysis. Section V concludes the paper.

II. Related Work

Much effort for imbalanced learning in the machine learning community has been devoted to resampling approaches [7] that randomly under-sample the majority class or over-sample the minority. Other smart resampling techniques are also available (please refer to [7] for a comprehensive survey). Generally under-sampling may remove valuable information and over-sampling easily introduces noise with overfitting risks. Additionally, random over-sampling does not increase information by only replication, so it does not solve the fundamental "lack of data" issue. SMOTE [6], on the other hand, creates new examples by interpolating neighboring minority class instances. However, it is error-prone to interpolate noisy or borderline examples. Therefore under-sampling is often preferred to over-sampling [8], but is not suitable for small datasets (e.g., FG-NET) because of the information loss.

Cost-sensitive learning [8]–[10] as an alternative is closely related to resampling. Instead of manipulating samples at the data level, it adjusts misclassification costs at the algorithmic

level and imposes heavier penalty on misclassifying the minority class. For example, Li and Lin [9] proposed RED-SVM to use the label-sensitive costs in the ordinal regression problem. Zadrozny *et al.* [10] combined cost sensitivity with ensemble approaches to further improve classification accuracy. Chen *et al.* [11] formed an ensemble of cost-sensitive decision trees by weighting the Gini criterion during the node splitting and final tree aggregation. We similarly grow cost-sensitive but more generalized and principled random trees, and propose a discriminative and extrapolative "aggregation" scheme that proves necessary for complex imbalanced data.

The above methods in [10], [11] already show the effectiveness of classifier ensemble in the context of data imbalance [11], [12]. Bagging and Boosting are the most popular ensemble strategies [13]. Generally, Boosting (e.g., [12], [23]) can easily embed the cost sensitivities in example weights according to the misclassification costs. Li *et al.* [23] further combined boosting with the training of an extreme learning machine. But boosting is vulnerable to noise and is more prone to overfitting, which can be better addressed by Bagging [13]. Our method based on the improved random forest is essentially a Bagging method, thus shares its advantages.

Age estimation: There are three main groups of age estimation methods: classification [2], [24], [25], regression [9], [26]–[29], and ranking [4], [30], [31] methods. OHRank [4], [31] surpasses previous classification- and regression-based methods by utilizing ordering information and cost sensitivities. However, the imbalance issue is neglected especially when designing ordered classifiers at the youngest and oldest ages. Some recent works focus on advanced feature extraction [24], [31], [32], including applying convolutional neural network (CNN) [15], [16] to automatically learn deep features instead of using hand-crafted ones. Unfortunately strong biases are still observed on imbalanced datasets, and we provide here an explicit solution to imbalanced learning with better results, using no deep features. Only three papers [29], [33], [34], as far as we know, consider data imbalance and sparseness when estimating ages. lsRCA [29] simply balances the number of nearest neighbors from each class to compute the similarity matrix for LPP (Locality Preserving Projection). In [33], [34], the imbalance is only mitigated by leveraging adjacent labels in implicit ways, respectively via modeling cumulative attribute space and label distribution. We will show the advantages of our explicit imbalanced learning mechanism and the extrapolative mechanism for possible missing data/labels.

Head pose estimation: Methods for head pose estimation from 2D images can be categorized into two main groups: classification [35] and regression [36]–[40], with regression being more attractive for its continuous output. We refer readers to [41] for a comprehensive survey. Random forest is a popular choice for pose estimation in both classification [35] and regression [39] settings. It is also applied to depth images [42]. To our knowledge, the inherent imbalance in pose data [3] is seldom addressed again. Note on many pose datasets such as Pointing'04, the sparse data sampling (with typical pose intervals of $10^\circ+$) makes learning even more difficult.

Edge detection: State-of-the-art edge detection methods [1], [5], [43]–[48] mostly use engineered gradient features to classify edge pixels/patches. Recent CNN-based methods [17]–[22] achieve top results by learning deep features. Due to the large variety of edge structures, it is usually hard to learn an ideal binary classifier to separate edges as one class from the non-edge class. Therefore some methods first cluster edge patches into compact subclasses (e.g., [5], [20]), and cast edge detection as a multi-way classification problem (i.e., to predict whether an input patch belongs to each edge subclass or the non-edge class) so as to implicitly solve the binary task. And the numbers of "positive" and "negative" patches are commonly set equal to facilitate the binary goal. However, this results in a severe imbalance between each edge subclass and the dominant negative one (see Figure 1(b)), which is barely addressed properly by the above methods. Consequently, biased predictions tend to occur with low edge recall or damage of fine edge structures in those rare subclasses.

Another limitation of existing methods is that they cannot well predict the unseen edge structures from a novel class. For example, Sketch Tokens [5] only predict from a pre-defined set of edge classes based on random forest. Structured Edge (SE) detector [44] can model more subtle edge variations in a structured forest framework without the finite-class assumption, but still can only infer the edge structures observed during training. Although this problem is ameliorated by merging predicted structures while testing, it is in sharp contrast to our explicit DSNA method that empowers random forests to extrapolate.

III. Methodology

The proposed discriminative sparse neighbor approximation (DSNA) aims to provide unbiased predictions given a class-imbalanced dataset. More precisely, given a training set $\mathcal{D} = \{s_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^D$ is the feature vector of sample s_i and y_i the label, our problem can be formulated as learning a function $F(\boldsymbol{x}) \rightarrow y$ to make unbiased predictions from severely imbalanced datasets. The label $y \in \mathcal{C}$ refers to the class index (e.g., edge class) for classification or a numeric value (e.g., age and pose angle) for regression.

For a query $\mathbf{q}$, the key steps of DSNA are: 1) to draw a well localized neighborhood of training data that is less probable to be invaded by imposter samples, 2) then follow the "divide and conquer" idea to perform a class-discriminative local clustering to obtain clusters (modeled as affine subspaces) without undesired class domination, 3) and finally choose the best cluster to make unbiased predictions.

The overall pipeline is shown in Figure 4. The pipeline begins with a cost-sensitive random decision Forest (CS-RF), which takes care of generating an initial good local neighborhood at leaf nodes, in order to reduce unnecessary distractions from unrelated samples. We retrieve all the leaf samples for a test instance, aiming to gain as more coverage of relevant minority samples as possible. The DSNA component starts with discriminative local clustering, and performs a sparse approximation to iteratively output unbiased predictions. To enable extrapolative prediction for unseen appearances, we model the found clusters as affine subspaces in order to extrapolate from them.

In the following, we first present the DSNA approach which is the key of this paper. We make the assumption that local

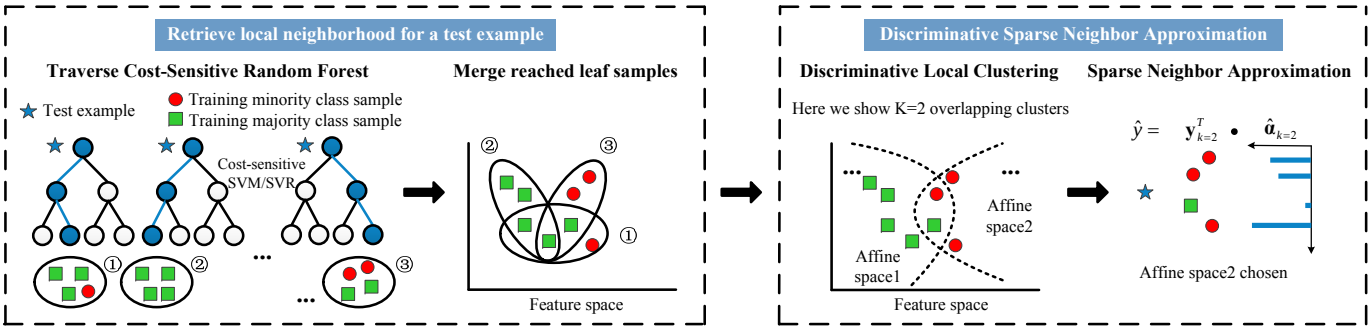


Fig. 4. The overall pipeline of our CS-RF-induced DSNA method.

neighborhood of training data is already available. We then describe the use of cost-sensitive random decision forest to obtain such local neighborhood.

A. Discriminative Sparse Neighbor Approximation

Discriminative local clustering - The first step of DSNA is to perform discriminative clustering within the local data neighborhood of a test sample. Suppose we have an initially retrieved local data neighborhood at hand, which can be noisy and class-imbalanced. This local neighborhood can be represented as

$$\mathcal{R} = \{s_i\}_{i=1}^{M}, \quad \mathcal{R} \subset \mathcal{D}, \quad M < N. \tag{1}$$

Intuitively, the samples in $\mathcal{R}$ are close to the test sample based on some notions of metric or non-metric distance. Our objective is to separate the samples in $\mathcal{R}$ based on their different class labels so as to pave the way for unbiased prediction of the test sample. We shall choose a clustering technique that possesses two desirable properties to achieve this goal: 1) It should generate discriminative clusters from one of which unbiased predictions can be made. 2) The found clusters should have adequate descriptiveness to account for unseen data patterns.

We achieve the aforementioned goal through a simple yet effective extension of K-means. It differs from the standard K-means in two respects. First, the inter-point distance $\widetilde{d}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is label-aware:

$$\widetilde{d}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} d(\boldsymbol{x}_i, \boldsymbol{x}_j) * \mathbf{1}(y_i \neq y_j) & \text{for classification,} \\ d(\boldsymbol{x}_i, \boldsymbol{x}_j) * g(|y_i - y_j|) & \text{for regression,} \end{cases} \tag{2}$$

where $d(\cdot,\cdot)$ is the Euclidean distance, $\mathbf{1}(\cdot)$ is an indicator function, $g(y) = \tau y/(max\{y\} - y + \text{eps})$ is a reciprocal increasing function with τ the trade-off parameter, and eps a small positive number to prevent overflow. The label-aware distance makes clustering discriminative by preferring the "same-class" data-pairs over those from different classes. In the extreme case, under classification scenarios for example, it forms clusters $\{\mathcal{L}_k\}_{k=1}^{K}$ each purely from one class even when the cluster members differ remarkably in appearances, which is suitable for classification.

Considering it is highly possible that the "pure" clusters in small imbalanced problems have limited samples, especially those mostly with the minority class samples, such clustering is actually not desirable for data/label extrapolation purposes. Hence, we allow cluster overlap by relaxing the cluster assignment of sample $\boldsymbol{x}_i$. Instead of assigning it solely to the nearest cluster centroid, we choose more than one centroids with distances slightly larger than the minimum distance in each K-means optimization iteration. This results in overlapping clusters each containing some "inter-class" samples. Such samples have complementary appearances to those "same-class" ones for enriching cluster representations.

Sparse neighbor approximation - The previous step generates K overlapping clusters $\{\mathcal{L}_k\}_{k=1}^{K}$ with their feature matrices $\{\boldsymbol{L}_k\}_{k=1}^{K}$ and labels $\{\boldsymbol{y}_k\}_{k=1}^{K}$. Our problem becomes how to discriminatively predict the label of a query $\boldsymbol{q}$ and extrapolate to its possibly unseen appearance simultaneously.

To address this problem, we model each cluster by an affine hull model $\mathcal{A}_k$ [14] that is able to account for unseen data of different modes, and then choose the best prediction returned by them. Every single $\mathcal{A}_k$ covers all possible affine combinations of its belonging samples and can be parameterized as:

$$\mathcal{A}_k = \{\boldsymbol{x} = \boldsymbol{\mu}_k + \boldsymbol{U}_k \boldsymbol{v}_k, k = 1, \ldots, K\}, \tag{3}$$

where $\boldsymbol{\mu}_k = \sum_{\boldsymbol{x}_i \in \mathcal{L}_k} \boldsymbol{x}_i / |\mathcal{L}_k|$ is the centroid, $\boldsymbol{U}_k$ is the orthonormal basis obtained from the SVD of centered $\boldsymbol{L}_k$, and $\boldsymbol{v}_k$ is the coefficient vector.

Note that to predict the class label of query $\mathbf{q}$, we still need to know which cluster the query should be assigned to, and the cluster index k in Eq. 3 remains unknown. To this end, we formulate a joint optimization problem for simultaneuously finding the belonging cluster of the query and its affine hull approximation:

$$\begin{aligned} &\min_{k, \boldsymbol{v}_k, \boldsymbol{\alpha}_k} \|\boldsymbol{\mu}_k + \boldsymbol{U}_k \boldsymbol{v}_k - \boldsymbol{L}_k \boldsymbol{\alpha}_k\|_2 + \lambda \|\boldsymbol{\alpha}_k\|_1 + \gamma \|\boldsymbol{\alpha}_k - \overline{\boldsymbol{\alpha}}_k\|_1, \\ &s.t. \ \|\boldsymbol{q} - (\boldsymbol{\mu}_k + \boldsymbol{U}_k \boldsymbol{v}_k)\|_2 \leq \varepsilon, \end{aligned} \tag{4}$$

where $\varepsilon \geq 0$, and λ and γ are regularization parameters.

We explain the objective function as follows:

First term - This term approximates $\boldsymbol{q}$ over the k^{th} cluster using the cluster's affine subspace as well as the feature matrix of associated member samples $\boldsymbol{L}_k$. This term is motivated by affine hull models [14] but differs significantly in the following aspects:

i) the affine space is class-aware. In particular, the affine space is learned from our class-discriminating cluster. A class-aware

sparsity constraint is further imposed to promote discrimination (*Third term*).
ii) the affine space approximation benefits from the enriched descriptiveness of overlapping clusters.

Second term - This term constrains the loose affine approximation by imposing sparsity among the cluster samples. Thus a large drift is avoided when extrapolating $\boldsymbol{q}$ on the affine subspace, because we constrain the affine subspace to be near to the observed samples using feature matrix $\boldsymbol{L}_k$.

Third term - This term regularizes the coefficient vector $\boldsymbol{\alpha}_k$ to make it focus more on the "same-class" nearest neighbors, $\mathcal{N}_k = \{\boldsymbol{x}_i \in \mathcal{L}_k : \widetilde{d}(\boldsymbol{x}_i, \boldsymbol{q}) \leq \varepsilon_k\}$, which are found by using the class-aware distances in Eq. 2. From our experiments, we empirically found that this term is useful to provide stable predictions. Formally, the $\overline{\boldsymbol{\alpha}}_k$ is estimated as:

$$\overline{\boldsymbol{\alpha}}_k = \sum_{\boldsymbol{x}_i \in \mathcal{N}_k} w_i \boldsymbol{\alpha}_i,\ w_i \propto \exp(-\widetilde{d}(\boldsymbol{x}_i, \boldsymbol{q})/h), \tag{5}$$

where h is the decay parameter, and $\boldsymbol{\alpha}_i$ is the representation coefficient of the i^{th} neighbor with the i^{th} element equal to one and the rest zero.

Eq. 4 can be solved by alternatively seeking the best affine approximation $\min_{k,\boldsymbol{v}_k} \|\boldsymbol{q} - (\boldsymbol{\mu}_k + \boldsymbol{U}_k \boldsymbol{v}_k)\|_2$ and the sparse neighbor approximation with two l_1-norms:

$$\min_{\boldsymbol{\alpha}_k} \|\boldsymbol{q} - \boldsymbol{L}_k \boldsymbol{\alpha}_k\|_2 + \lambda \|\boldsymbol{\alpha}_k\|_1 + \gamma \|\boldsymbol{\alpha}_k - \overline{\boldsymbol{\alpha}}_k\|_1, \tag{6}$$

which can be efficiently solved by using the Augmented Lagrange Multiplier (ALM) method [49].

With the converged $\hat{\boldsymbol{\alpha}}_k$, the label for $\boldsymbol{q}$ is finally predicted as $\hat{y} = \boldsymbol{y}_k^T \hat{\boldsymbol{\alpha}}_k$ for regression or by majority voting for classification (in this case we determine the nonzero entries of thresholded $\hat{\boldsymbol{\alpha}}_k$, and vote among the corresponding $\boldsymbol{y}_k$). The initial label in the iterative process is the mean or majority vote of $\boldsymbol{y}_k$ in the best-fit cluster.

B. Cost-Sensitive Random Decision Forest

Returning to the initial step of finding a 'safe' local neighborhood, we choose random decision forest for its efficiency and robustness. We first traverse a test example through every trained decision tree and retrieve the respective training samples $\mathcal{R}_t$ stored at the leaf node. Traditional random forest calculates either a class distribution for classification or a local mean for regression from each $\mathcal{R}_t$, and aggregates them as the final prediction. We face two fundamental problems by doing so: in the case of absolute rarity, each $\mathcal{R}_t$ will still predominantly consist of majority classes that make simple aggregation biased to them; or $\mathcal{R}_t$ may form pure but small disjuncts [7] of minority class samples leading to overfit.

Therefore, we instead merge all the retrieved leaf sample sets $\{\mathcal{R}_t\}$ into a single one $\mathcal{R} = \cup_t \mathcal{R}_t$, and treat $\mathcal{R}$ as our initial local neighborhood in Eq. 1. Then DSNA is applied for prediction as described in Section III-A. Such simple merging helps our data-driven method to gain as more coverage of relevant minority samples as possible. This can be easily satisfied thanks to the diversities between different trees. In fact, random forest has the proved upper bound of generalization error given by [13]:

$$\epsilon \leq \rho(1 - m^2)/m^2, \tag{7}$$

where m is the strength of individual trees and ρ is the correlation between decision trees. Hence in order to maintain the low correlation and diversity among trees, we just keep the Bagging nature and feature randomness at internal nodes as in standard random forest.

To make the merged neighborhood less distracted by imposter samples, we focus on improving the strength m of each tree in the context of data imbalance by making the tree cost-sensitive. We have explored different cost-sensitive schemes, such as the re-weighting of nodes as in [11] and boosting of trees with class costs, but seen marginal effects. We finally came to a modified node splitting rule that can not only take into account the imbalanced distribution, but also can work seamlessly for both classification and regression.

Specifically, we first follow the standard Bagging procedure to grow an ensemble of random trees. Each tree recursively divides the input space into disjoint partitions in a coarse-to-fine manner. The key is to design good splitting functions. For a node j with local samples $\mathcal{S}_j$, a binary function $\phi_j : \mathbb{R}^{D'} \rightarrow \{0, 1\}$ is trained on randomly sampled features ($D' = \sqrt{D}$) and splits into $\mathcal{S}_j^l$ and $\mathcal{S}_j^r$ to maximize the information gain:

$$\mathcal{I}(\mathcal{S}_j, \phi_j) = H(\mathcal{S}_j) - \left(\frac{|\mathcal{S}_j^l|}{|\mathcal{S}_j|} H(\mathcal{S}_j^l) + \frac{|\mathcal{S}_j^r|}{|\mathcal{S}_j|} H(\mathcal{S}_j^r) \right), \tag{8}$$

where $H(\cdot)$ denotes the class entropy. For regression, information gain can be replaced by the label variance as $H(\mathcal{S}) = \sum_y (y - \mu)^2 / |\mathcal{S}|$ where $\mu = \sum_y y / |\mathcal{S}|$. Training stops when a maximum depth is reached or if information gain or local sample size $|\mathcal{S}_j|$ falls below a fixed threshold.

The standard node splitting function ϕ_j is not necessarily optimal with respect to imbalanced data. To alleviate this problem, in both classification and regression scenarios, we incorporate a cost function $f(\cdot) \geq 0$ into ϕ_j that penalizes more heavily on the minority class.

In classification trees, we first apply the widely used K-means technique [39], [44] to cluster $\mathcal{S}_j$ into $\{\mathcal{S}_j^k\}_{k=1}^2$, and then the splitting function ϕ_j that best preserves the two clusters is determined by a cost-sensitive version of linear SVM:

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_2 + C \sum_{k=1}^{2} f(p_k) \sum_{\boldsymbol{x}_i \in \{\mathcal{S}_j^k\}} \left(max(0, 1 - z_i \boldsymbol{w}^T \boldsymbol{x}_i) \right)^2, \tag{9}$$

where $p_k = |\mathcal{S}_j^k| / |\mathcal{S}_j|$ denotes the cluster proportion, $\boldsymbol{w}$ is the weight vector, C is a regularization parameter, and $z_i = 1$ if $\boldsymbol{x}_i \in \mathcal{S}_j^1$ and -1 otherwise. Each sample is finally sent to either $\mathcal{S}_j^l$ or $\mathcal{S}_j^r$ by $sgn(\boldsymbol{w}^T \boldsymbol{x}_i)$. The resulting splitting function is thus *learned* in a cost-sensitive manner instead of being chosen from some predefined splitting rules. Note the cost here is defined as a function of the cluster distribution rather than the targeted class distribution, but they will correlate well at the deeper tree depth with much purer nodes where Eq. 9 can better play its role.

Algorithm 1 : CS-RF-Induced DSNA

Input: Training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, trained CS-RF, query $\boldsymbol{q}$.
Initialization: to predict y of $\boldsymbol{q}$
- Merge for $\boldsymbol{q}$ all its reached leaf samples to $\mathcal{R}$.
- Via **Discriminative Local Clustering**, obtain clusters $\{\mathcal{L}_k\}_{k=1}^K$, features $\{\boldsymbol{L}_k\}_{k=1}^K$ and labels $\{\boldsymbol{y}_k\}_{k=1}^K$.
- Set $y^{(0)}$ as the mean of $\boldsymbol{y}_k$ for regression or its majority vote for classification in $\mathcal{A}_k$ that best approximates $\boldsymbol{q}$.

Outer Loop: Iterate on $t = 1, \ldots, T$ until convergence
- Update $\{k^{(t-1)}, \boldsymbol{v}_k^{(t-1)}\}$ by Eq. 3 as the ones that best approximate $\boldsymbol{q}$.
- Update the sparse coefficient estimate $\overline{\boldsymbol{\alpha}}_k^{(t-1)}$ by Eq. 5.
- Update $\boldsymbol{\alpha}_k^{(t-1)}$ via **Sparse Neighbor Approximation** by minimizing Eq. 6.
- Predict label $y^{(t)} = \boldsymbol{y}_k^T \boldsymbol{\alpha}_k^{(t-1)}$ or by majority voting among $\boldsymbol{y}_k$ with nonzero coefficients.

Output: Converged label $\hat{y}$.

In regression trees, we perform a cost-sensitive regression at each node $\mathcal{S}_j$ using a weighted linear SVR:

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_2 + C \sum_{y \in \mathcal{C}} f(p_y) \sum_{\substack{y_i = y \\ \boldsymbol{x}_i \in \mathcal{S}_j}} \left(max(0, |y_i - \boldsymbol{w}^T \boldsymbol{x}_i| - \varepsilon)\right)^2, \tag{10}$$

where $\varepsilon \geq 0$, and we directly penalize the true label distribution $\{p_y = |\{y_i = y, \boldsymbol{x}_i \in \mathcal{S}_j\}|/|\mathcal{S}_j|\}$ as costs. The node then branches left if the numeric prediction $\{\boldsymbol{w}^T \boldsymbol{x}_i\}$ is smaller than the local mean of labels $\sum_{\boldsymbol{x}_i \in \mathcal{S}_j} y_i/|\mathcal{S}_j|$, otherwise branches right.

In practice, we use the cost transformation technique in [4] to solve the above weighted SVM/SVR. The cost function $f(\cdot)$ is defined by a reciprocal decreasing function as $f(p) = (1-p)/p$. Obviously, $f(p)$ gives larger weights to the minority classes which proves effective to improve their prediction accuracies without losing the overall performance in our experiments. In addition, we use the inverse class frequencies to reweight the information gain (Eq. 8, as in [11]) to select the best D' random features in both classification and regression trees. The result is a CS-RF framework able to carve reasonably good local neighborhoods for both the majority and minority classes.

C. Convergence and Complexity

Our full algorithm is detailed in Algorithm 1. Similar to the affine hull (AH) method [14], Algorithm 1 can converge to a global solution. Compared with AH's non-asymptotic convergence rate of $O(1/t^2)$, our DSNA method converges even faster as shown in Figure 5. Typically DSNA converges within 10 iterations with lower objective values thanks to the introduced class discrimination as a guidance. In our experiments, we will visualize some converged examples with accurate predictions in different vision tasks.

IV. Experiments

We validate the effectiveness of our CS-RF-induced DSNA method in three vision tasks at various imbalance levels: the age estimation and head pose estimation tasks (by regression) and the edge detection task (by classification).

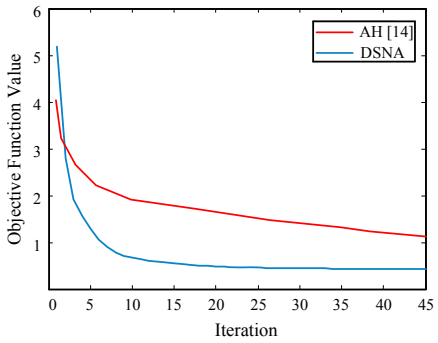


Fig. 5. Comparison of the convergence of unsupervised AH [14] and our DSNA given a query in the age estimation task.

A. Experimental Settings

Dataset settings: For age estimation, the FG-NET [2] and MORPH [4] datasets are used. FG-NET contains 1002 facial images of 82 subjects with ages in a range from 0 to 69. Algorithms are evaluated by the leave-one-person-out protocol. MORPH contains about 55000 images of more than 13000 subjects with ages between 16 and 77. We randomly split it into three disjoint subsets S1, S2 and S3 as in [16]. Algorithms repeat 1) training on S1, testing on S2+S3 and 2) training on S2, testing on S1+S3 with the average result reported. Both datasets are highly imbalanced (see Figure 1(a)) and class-overlapped. FG-NET further suffers from the issue of small data. For both, we use AAM [50] as the feature extractor, and Mean Absolute Error (MAE) as the evaluation metric.

For head pose estimation, poses should intrinsically admit an imbalanced distribution with much more near-frontal instances than the profile ones. Unfortunately, we are unable to obtain such datasets with ground truth labels (e.g., "Face Pose" dataset [3]) for experiments. We instead adopt the popular Pointing'04 dataset that exhibits some imbalance in pitch angles. The dataset contains images from 15 subjects each with two series of 93 pose images. The pose is discretized into 9 pitch angles $\{\pm 90°, \pm 60°, \pm 30°, \pm 15°, 0°\}$ and 13 yaw angles $\{\pm 90°, \pm 75°, \pm 60°, \pm 45°, \pm 30°, \pm 15°, 0°\}$. However, when the pitch angles are $\{\pm 90°\}$, the yaw is always $\{0°\}$ (so $7 \times 13 + 2 = 93$ poses in total), leading to an imbalance ratio of 1:13 between $\{\pm 90°\}$ pitch angles and other pitch angles. We further test when pitch angles are randomly removed to form a Gaussian-like distribution to mimic the real-world imbalanced distribution. As in [38], [39], evaluation of MAE is performed with 5-fold cross-validation using HOG features.

For edge detection, we use the BSDS500 [1] and NYUD (v2) [51] datasets, the latter for testing cross-dataset generalization. BSDS500 contains 200 training, 100 validation and 200 testing images. NYUD contains 1449 pairs of RGB and depth images. We follow [46] to use 60%/40% training/testing split (1/3 training data for validation) with the images reduced to 320×240 pixels. For cross-dataset testing, we only use RGB images on both datasets. We combine our method in classification mode with the structured edge detector [44] since it induces classification forest like us but operates on edge

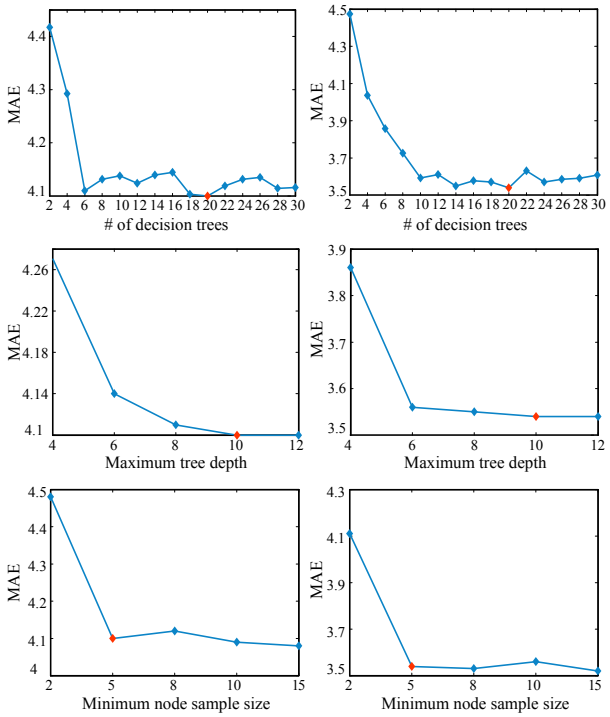


Fig. 6. Parameter sweeps for age estimation (left column) and head pose estimation (right column). Each row (from top to bottom) considers the parameter of tree number, maximum tree depth and minimum node sample size, respectively. The chosen parameter value is marked in red.

patches instead of pixels, which proves efficient in practice. We use the same multiple low-level features extracted from 32×32 image patches and apply non-maximal suppression prior to evaluation as in [44]. Edge detection accuracy is evaluated by the fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP) [1], which are very suitable to assess the performance of such an imbalanced problem.

Parameters: For age and head pose estimation, we empirically combine 20 cost-sensitive trees in our regression forest, and terminate splitting when the maximum depth 10 is reached or if the node sample size is smaller than 5. Figure 6 shows the robustness of these parameters across tasks. Evaluations are done by varying one parameter at a time, with others fixed. The chosen parameter value is marked in red. For edge detection, we use the same parameter setting with [44].

Cross-validation is used to determine the trade-off parameter C for cost-sensitive SVM/SVR (Eq. 9 and 10), τ for biased distance (Eq. 2), λ and γ in Eq. 4. We select K for discriminative local clustering from 2 to 4.

B. Evaluation of the CS-RF and DSNA

We start with evaluating our key components of CS-RF and DSNA. CS-RF concerns about generating good local neighborhood, while DSNA makes unbiased and extrapolative prediction and is the major contribution of this paper.

Figure 7 visualizes the advantage of DSNA over simple averaged prediction in the three considered tasks. Clearly, given an appropriate local neighborhood, e.g., by CS-RF, DSNA can localize the correct mode (cluster) in it for the difficult minority class samples. As a result, DSNA makes much more unbiased predictions than simple averaging. More

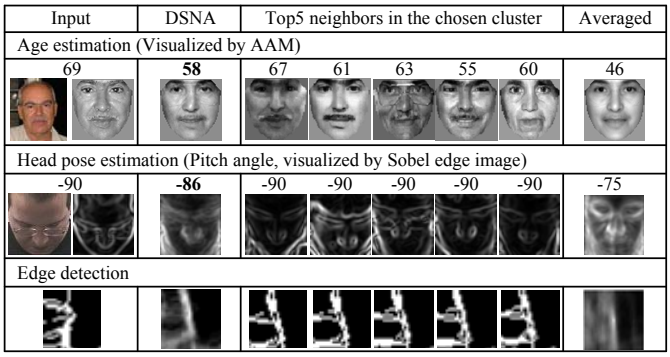


Fig. 7. Visualizations of both the DSNA converged result and simple averaged result among the retrieved samples by CS-RF. Results are obtained for the minority class testing samples in all the three tasks.

TABLE I
ABLATION TEST FOR CS-RF AND DSNA IN AGE ESTIMATION (MAE ON FG-NET), HEAD POSE ESTIMATION (AVG. MAE ON POINTING'04) AND EDGE DETECTION (ODS ON BSDS500, HIGHER IS BETTER).

Methods	RF	RF+ SMOTE [6]	RED-SVM [9]	WRF [11]	CS-RF	CS-RF+ AH [14]	CS-RF+ DSNA
Age	5.28	5.39	5.24	–	**4.81**	4.89	**4.10**
Pose	6.41	6.65	6.53	–	**4.02**	4.28	**3.54**
Edge	0.75	0.75	–	0.75	**0.76**	0.76	**0.78**

significantly, for age estimation on the small FG-NET dataset, although there are very few elderly samples, our DSNA still extrapolates well from the limited data.

Table I quantifies the benefits of both CS-RF and DSNA against other competitive schemes in vision tasks. Note all the RF variants in the left and middle columns—RF+SMOTE, WRF (Weighted RF) and CS-RF simply average tree predictions as in standard RF. They do not consider data extrapolation as AH (Affine Hull) [14] and DSNA do. We make the following observations: 1) The over-sampling method SMOTE shows no benefits over Bagging in standard RF since it can introduce undesirable noise (e.g., in age and pose cases). 2) Cost-sensitive learning, in the middle column, helps for these imbalanced tasks, and our CS-RF consistently outperforms RED-SVM and WRF. This suggests that simple weighting schemes in RED-SVM and WRF are not adequate in complex imbalanced tasks. In contrast, our CS-RF can be seen as an ensemble of cost-sensitive experts organized in hierarchical trees, and has higher capability and robustness. Another advantage is that CS-RF provides a unified cost-embedded solution to both regression and classification. 3) The supervised DSNA combined with CS-RF leads to large improvements, whereas the unsupervised AH shows no improvements or even worse results. This emphasizes the importance of using supervisory information. DSNA uses such information intelligently by extrapolating from several discriminatively trained AH models with a class-aware constraint (Eq. 4).

C. Comparison with State-of-the-Arts

Age estimation: We compare with the state-of-the-arts on FG-NET and MORPH datasets in Table II. Our CS-RF+DSNA outperforms most methods by a large margin, and reduces the MAEs of the runner-up lsRCA and MSCNN on the

TABLE II
COMPARISONS OF AGE ESTIMATION RESULTS (MAE) ON FG-NET AND MORPH DATASETS.

FG-NET							MORPH	
RUN [30]	RED-SVM [9]	MTWGP [28]	BIF [32]	CPNN [34]	CSOHR [31]	CA-SVR [33]	KPLS [26]	KCCA [27]
5.33	5.24	4.83	4.77	4.76	4.70	4.67	4.04	3.98
Choi *et al.* [25]	MidFea-NS [15]	Han *et al.* [52]	RealAdaBoost [24]	OHRank [4]	lsRCA [29]	CS-RF+DSNA	MSCNN [16]	CS-RF+DSNA
4.66	4.62	4.60	4.49	4.48	4.38	**4.10**	3.63	**3.54**

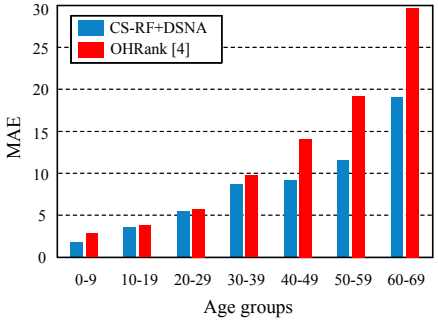


Fig. 8. MAEs at different age groups on the FG-NET dataset.

TABLE III
COMPARISON OF POSE ESTIMATION MAES[°] ON POINTING'04.

Method	Yaw	Pitch	Avg.
KPLS [38]	6.56	6.61	6.59
SLDML [40]	6.31	6.71	6.51
Fenzi *et al.* [37]	5.94	6.73	6.34
GLLiM [36]	5.62	6.68	6.15
KRF [39]	5.29	2.51	3.90
CS-RF+DSNA	**5.04**	**2.03**	**3.54**

two datasets by 6.4% and 2.5% respectively. The larger improvement on FG-NET is impressive because the dataset is very small and has missing class labels (old ages). This validates our competence in synthesizing novel labels on small imbalanced datasets. Note the mere cost-sensitive methods RED-SVM, CSOHR and OHRank all show their inferiority on this imbalanced dataset, necessitating the ability of extrapolation. The advanced features—Bio-Inspired Features (BIF), generalized BIF with scattering transform in [31] and feature selection by RealAdaBoost [24] also do not reach top in this task. In contrast, our method, using the AAM features only, even outperforms the deep feature-based MidFea-NS and MSCNN due to the handling of data imbalance. Compared with the indirect imbalance-handling methods, CPNN, CA-SVR and lsRCA, ours performs much better by introducing explicit mechanisms that are discriminative and extrapolative.

Figure 8 shows the MAE per decade for a detailed analysis of the different errors and difficulties in the entire imbalanced distribution. From the comparison with OHRank, the benefit of our method becomes prominent at old ages with very limited samples. We attribute this benefit to the imbalanced learning and extrapolation abilities of the proposed DSNA. On the other hand, we do not lose accuracy (even better) for those majority or normal ages, which is desirable.

Head pose estimation: Table III compares our method with the regression-based prior arts KPLS, SLDML, Fenzi *et al.*, GLLiM and KRF on Pointing'04 dataset. As mentioned in

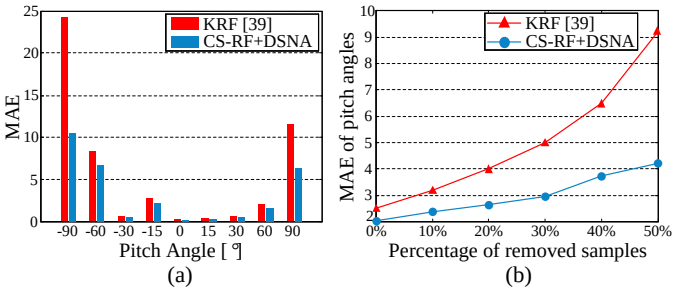


Fig. 9. Comparisons of imbalanced pitch angle estimation on Pointing'04. (a) MAEs at different pitch angles. (b) Average pitch MAEs with different percentages of samples removed.

Section II, the sparse sampling of pose angle compounds the learning difficulty, especially for the imbalanced pitch angles. Our method performs best for both pose angles, with a large margin for imbalanced pitch. Figure 9(a) compares our results at individual pitch angles with those of KRF, the state-of-the-art regression forest-based method. Again due to the proposed cost-sensitive RF and extrapolative DSNA, our method can better handle data imbalance at the rare $\pm 90°$ poses. Specifically, we obtain an MAE of 8 degrees for those $\pm 90°$ poses with only 24 training samples (hundreds of samples for other poses), which is 55.6% lower than that of KRF. For those normal poses besides $\pm 90°$, we still have MAEs lower than or comparable to KRF. We finally show the performance in Figure 9(b) when pitch samples are randomly removed to form a Gaussian-like distribution to mimic real-world distributions. Our performance degrades more gracefully with the increase of removed data, showing a strong ability to handle small imbalanced data.

Edge detection: We refer to our combined method with structured edge (SE) detector [44] as CS-SE+DSNA. Table IV summarizes an extensive comparison with state-of-the-art methods on BSDS500. It is observed that CS-SE+DSNA outperforms all "shallow" methods (top cell) across all evaluation metrics, and also performs better than most deep models (bottom cell). CS-SE+DSNA is even comparable to the top HED method that has 16 deep layers, and also to the latest deep learning methods [53], [54]. Such results are very impressive because our method only uses hand-designed features but can reach top with the built-in imbalance handling mechanism.

This advantage also holds over those similar random forest-based methods—Sketch Tokens, SE and OEF. The major reason again lies in our capability of correctly classifying imbalanced edge patches and generalizing to novel edge structures. Figure 10(a) compares our CS-SE+DSNA with three random forest-based methods, including DeepContour that applies random forest on top of deeply learned features.

TABLE IV
COMPARISON OF EDGE DETECTION RESULTS ON THE BSDS500 DATASET.

Method	ODS	OIS	AP
ISCRA [47]	0.72	0.75	0.46
gPb-owt-ucm [1]	0.73	0.76	0.73
Sketch Tokens [5]	0.73	0.75	0.78
SCG [46]	0.74	0.76	0.77
PMI+sPb [45]	0.74	0.77	0.78
SE [44]	0.75	0.77	0.80
OEF [48]	0.75	0.77	0.82
SE+multi-ucm [43]	0.75	0.78	0.76
DeepNet [17]	0.74	0.76	0.76
N^4-Fields [18]	0.75	0.77	0.78
DeepEdge [19]	0.75	0.77	0.81
DeepContour [20]	0.76	0.78	0.80
Unsupervised patch [54]	0.77	0.78	0.82
HFL [21]	0.77	0.79	0.80
LMLE-kNN [53]	**0.78**	0.79	**0.83**
HED [22]	**0.78**	**0.80**	**0.83**
CS-SE+DSNA	0.77	0.79	0.81

TABLE V
COMPARISON OF EDGE DETECTION (TOP) AND CROSS-DATASET GENERALIZATION (BOTTOM) RESULTS ON THE NYU DATASET USING ONLY RGB IMAGES. TRAIN/TEST INDICATES THE TRAINING/TESING DATASET USED.

Method	ODS	OIS	AP
gPb [1] (NYU/NYU)	0.51	0.52	0.37
SCG [46] (NYU/NYU)	0.55	0.57	0.46
SE [44] (NYU/NYU)	0.60	0.61	0.56
CS-SE+DSNA (NYU/NYU)	**0.62**	**0.63**	**0.60**
SE [44] (BSDS/NYU)	0.55	0.57	0.46
DeepContour [20] (BSDS/NYU)	0.55	0.57	0.49
CS-SE+DSNA (BSDS/NYU)	**0.57**	**0.58**	**0.51**

Clearly, CS-SE+DSNA is able to produce cleaner results with preserved edge structures. In other words, it is capable of predicting the minority edges without jeopardizing the majority non-edges that make edge maps clean.

To further validate the extrapolative ability of our method, we perform the cross-dataset generalization test in comparison to other competing methods. The NYU/NYU results are used as baselines, see Table V. In both cases of NYU/NYU and BSDS/NYU testing, we find favorable performance, demonstrating a superior capability of generalization. Figure 10(b) shows the visual results.

D. Generalization and Runtime Analysis

We here conduct another experiment on the non-image-typed KEEL dataset repository [55] to show the generality of our method. Specifically, we consider three classification datasets Iris0, Glass6, and Glass2 with various imbalance levels (see their imbalance ratios in Table VI), and two regression datasets Abalone and Treasury that also have highly skewed target variables. For all datasets, we adopt 5-fold cross-validation to calculate the AUC (Area Under the ROC Curve) metric for imbalanced classification, and report the standard Mean Squared Error (MSE) for regression. Table VI shows that our CS-RF+DSNA method can cope well with data imbalance on these generic datasets, and always performs better than the representative baselines SVM and WRF [11].

TABLE VI
CLASSIFICATION (AUC, HIGHER IS BETTER) AND REGRESSION (MSE, LOWER IS BETTER) RESULTS ON GENERIC KEEL DATASETS. IMBALANCE RATIO (IR) IS THE RATIO OF THE NUMBER OF MAJORITY CLASS INSTANCES TO THE NUMBER OF MINORITY CLASS INSTANCES.

	Classification			Regression	
Dataset	Iris0	Glass6	Glass2	Abalone	Treasury
IR	2.00	6.38	10.39	-	-
SVM	0.9900	0.8752	0.5000	2.8517	0.0637
WRF [11]	0.9800	0.9117	0.7282	2.6014	0.0594
CS-RF+DSNA	**0.9929**	**0.9338**	**0.7943**	**2.3521**	**0.0385**

TABLE VII
RUNTIME (PER IMAGE) VS. PERFORMANCE (AGE MAE ON FG-NET, AVG. POSE MAE ON POINTING'04, EDGE ODS ON BSDS500).

	Age		Pose		Edge		
Methods	MSCNN [16]	Ours	KRF [39]	Ours	SE [44]	HED [22]	Ours
Performance	3.63	**3.54**	3.90	**3.54**	0.75	**0.78**	0.77
Runtime	200ms	**23ms**	**7.7ms**	19.4ms	**400ms**	12s	550ms

Table VII presents the runtime analysis for our method in comparison to the top performing methods in each vision domain. The runtime is tested on an Intel Core i7 4.0GHz CPU. During training, it takes a similar amount of time to generate our CS-RF and existing RF methods like KRF [39] and SE [44]. While during testing, our DSNA only introduces a modest computational overhead as compared to RF methods, but leads to better performance. Our speed advantages over deep models MSCNN [16] and HED [22] are obviously large, and the performance is comparable, which makes our method a more desirable choice in such imbalanced problems.

V. CONCLUSION

We propose in this paper a principled method to handle data imbalance and make unbiased predictions with preserved discriminative and extrapolative ability. The predictions are made by discriminative sparse neighbor approximation, within the local data neighborhood retrieved by a cost-sensitive decision forest. The proposed method proves effective in diverse vision tasks at various imbalance levels, and substantially outperforms the state-of-the-arts including some deep learning methods that ignore the imbalance issue. We show its great potential as an efficient and general purpose solution for imbalanced learning. Future works include making the framework deeper by using cascaded forests with multi-level predictions, to explore the extent to which we can achieve by simulating deep architectures.

ACKNOWLEDGMENT

This work is supported by the General Research Fund sponsored by the Research Grants Council of the Hong Kong SAR (CUHK 416713, 14241716, 14224316).

REFERENCES

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[2] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2234–2240, 2007.

Fig. 10. Edge detection results on the (a) BSDS500 dataset and (b) NYU dataset with BSDS trained model.

[3] J. Aghajanian and S. Prince, "Face pose estimation in uncontrolled environments," in *Proc. British Mach. Vis. Conf.*, 2009.

[4] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011.

[5] J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[7] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[8] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. Int. Conf. Mach. Learn. Workshops*, 2003.

[9] L. Li and H. Lin, "Ordinal regression by extended binary classification," in *Neural Inf. Process. Syst.*, 2006.

[10] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proc. IEEE Int. Conf. Data Mining*, 2003.

[11] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," University of California, Berkeley, Tech. Rep. 666, 2004.

[12] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2000.

[13] L. Breiman, "Random forests," *J. Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[14] Y. Hu, A. Mian, and R. Owens, "Sparse approximated nearest points for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011.

[15] S. Kong, Z. Jiang, and Q. Yang, "Learning mid-level features and modeling neuron selectivity for image classification," *arXiv preprint*, vol. arXiv:1401.5535, 2014.

[16] D. Yi, Z. Lei, and S. Z. Li, "Age estimation by multi-scale convolutional network," in *Proc. Asian Conf. Comput. Vis.*, 2014.

[17] J. J. Kivinen, C. K. I. Williams, and N. Heess, "Visual boundary prediction: A deep neural prediction network and quality dissection," in *Proc. Int. Conf. Artif. Intell. Stats.*, 2014.

[18] Y. Ganin and V. S. Lempitsky., "N4-fields: Neural network nearest neighbor fields for image transforms," in *Proc. Asian Conf. Comput. Vis.*, 2014.

[19] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[20] W. Shen, X. Wang, Y. Wang, and X. Bai, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[21] G. Bertasius, J. Shi, and L. Torresani, "High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[22] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[23] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted ELM for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[24] H. Ren and Z.-N. Li, "Age estimation based on complexity-aware features," in *Proc. Asian Conf. Comput. Vis.*, 2014.

[25] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim, "Age estimation using a hierarchical classifier based on global and local facial features," *Pattern Recognit.*, vol. 44, no. 6, pp. 1262–1281, 2011.

[26] G. Guo and G. Mu, "Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011.

[27] ——, "Joint estimation of age, gender and ethnicity: CCA vs. PLS," in *Proc. IEEE Int. Conf. Autom. Face and Gesture Recognit.*, 2013.

[28] Y. Zhang and D.-Y. Yeung, "Multi-task warped gaussian process for personalized age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010.

[29] W.-L. Chao, J.-Z. Liu, and J.-J. Ding, "Facial age estimation based on label-sensitive learning and age-oriented regression," *Pattern Recognit.*, vol. 46, no. 3, pp. 628–641, 2013.

[30] S. Yan, H. Wang, T. Huang, Q. Yang, and X. Tang, "Ranking with uncertain labels," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2007.

[31] K.-Y. Chang and C.-S. Chen, "A learning framework for age rank estimation based on face images with scattering transform," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 785–798, 2015.

[32] G. Guo, G. Mu, Y. Fu, and T. Huang, "Human age estimation using bio-inspired features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009.

[33] K. Chen, S. Gong, T. Xiang, and C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.

[34] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2401–2412, 2013.

[35] C. Huang, X. Ding, and C. Fang, "Head pose estimation based on random forests for multiclass classification," in *Proc. Int. Conf. Pattern Recognit.*, 2010.

[36] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *Proc. IEEE Int. Conf. Image Process.*, 2015.

[37] M. Fenzi, L. Leal-Taixe, B. Rosenhahn, and J. Ostermann, "Class generative models based on feature regression for pose estimation of object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.

[38] M. Haj, J. Gonzalez, and L. Davis, "On partial least squares in head pose estimation: How to simultaneously deal with misalignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012.

[39] K. Hara and R. Chellappa, "Growing regression forests by classification: Applications to object pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[40] Y. Liu, Q. Wang, Y. Jiang, and Y. Lei, "Supervised locality discriminant manifold learning for head pose estimation," *Knowl. Based Syst.*, vol. 66, pp. 126–135, 2014.

[41] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[42] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool, "Random forests for real time 3D face analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 437–458, 2013.

[43] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.

[44] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

[45] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson, "Crisp boundary detection using pointwise mutual information," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[46] X. Ren and L. Bo, "Discriminatively Trained Sparse Code Gradients for Contour Detection," in *Neural Inf. Process. Syst.*, 2012.

[47] Z. Ren and G. Shakhnarovich, "Image segmentation by cascaded region agglomeration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.

[48] S. Hallman and C. C. Fowlkes, "Oriented edge forests for boundary detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[49] D. Bertsekas, A. Nedic, and A. Ozdaglar, "Convex analysis and optimization," *Athena Scientific*, 2003.

[50] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

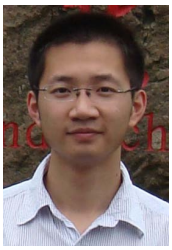
[51] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012.

[52] H. Han, C. Otto, and A. Jain, "Age estimation from face images: Human vs. machine performance," in *Proc. Int. Conf. Biometrics*, 2013.

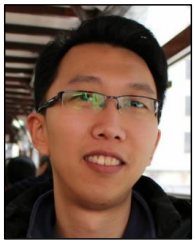
[53] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[54] C. Huang, C. C. Loy, and X. Tang, "Unsupervised learning of discriminative attributes and visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[55] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult.-Valued Log. S.*, vol. 17, no. 2-3, pp. 255–287, 2011.



Chen Huang received the Ph.D. degree in Electronic Engineering from Tsinghua University, Beijing, China, in 2014. He is currently a postdoctoral fellow in the Robotics Institute of Carnegie Mellon University. His research interests include machine learning and computer vision, with focus on deep learning and face analysis.



Chen Change Loy received the PhD degree in Computer Science from the Queen Mary University of London in 2010. He is currently a Research Assistant Professor in the Department of Information Engineering, Chinese University of Hong Kong. Previously he was a postdoctoral researcher at Queen Mary University of London and Vision Semantics Ltd. His research interests include computer vision and pattern recognition, with focus on face analysis, deep learning, and visual surveillance. He serves as an Associate Editor of IET Computer Vision Journal and a Guest Editor of Computer Vision and Image Understanding. He is currently a member of IEEE.

Xiaoou Tang received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a Professor and the Chairman of the Department of Information Engineering. He worked as the group manager of the Visual Computing Group at the Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing. Dr. Tang received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and has served as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and International Journal of Computer Vision (IJCV). He is a Fellow of IEEE.