Criteria Sliders: Learning Continuous Database Criteria via Interactive Ranking

James Tompkin* Brown University, USA
http://www.jamestompkin.com/
Kwang In Kim* University of Bath, UK
k.kim@bath.ac.uk
Hanspeter Pfister Harvard University, USA
https://vcg.seas.harvard.edu/
Christian Theobalt MPI for Informatics, Germany
http://gvv.mpi-inf.mpg.de/
* Equal contribution.

Abstract

Large databases are often organized by hand-labeled metadata—or *criteria*—which are expensive to collect. We can use unsupervised learning to model database variation, but these models are often high dimensional, complex to parameterize, or require expert knowledge. We learn low-dimensional continuous criteria via *interactive ranking*, so that the novice user need only describe the relative ordering of examples. This is formed as semi-supervised label propagation in which we maximize the information gained from a limited number of examples. Further, we actively suggest data points to the user to rank in a more informative way than existing work. Our efficient approach allows users to interactively organize thousands of data points along 1D and 2D continuous sliders. We experiment with databases of imagery and geometry to demonstrate that our tool is useful for quickly assessing and organizing the content of large databases.

1 Introduction

Computer vision helps automatically model visual databases with high-dimensional 2D image and 3D shape features. From these, we parameterize *criteria* for easy database organization by embedding the high-dimensional representations into low-dimensional spaces, e.g., for face expression, we ascribe 'amount of smile' from the high-dimensional features. This task is complicated, because most desired criteria do not map to individual features, and instead map to complicated paths lying on a manifold in the high-dimensional feature space. As such, this parameterization is accomplished either by an expert, or by supervised learning from laboriously-collected labeled examples [8, 24].

When surveying a visual database, even an expert would need time to assess a database of a few thousand items for parameterization. This is especially time consuming for new databases, e.g., scraping the Internet and recovering unknown contents. Even after parameterization, organization tools are still restricted to only the expert-defined criteria, with no easy way for users to define new criteria for their interests, especially if they are abstract or cross typical boundaries. The ability to *interactively* parameterize a database is needed: to quickly describe database variation without prior knowledge or labels; to intuitively discover low-dimensional criteria from high-dimensional models.

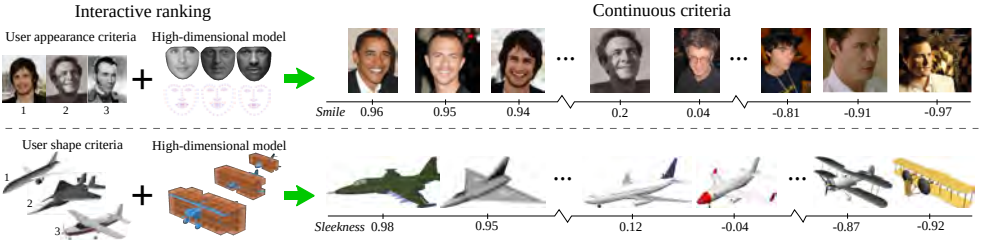

Figure 1: Users generate continuous criteria *(right)* interactively by ranking examples *(left)*. *Top:* A database of face images is ordered on a 'smiling expression' 1D parameterization. *Bottom:* A database of aircraft geometry is ordered by the user criteria of 'sleekness'.

We present an interactive system to generate continuous criteria from high-dimensional models. Interactive labeling requires efficient computation, and ideally extracts the most information from the fewest user-provided labels. Thus, to exploit the rich structure of unlabeled data points, we adapt a state-of-the-art semi-supervised learning algorithm for interactive use. To best exploit limited user interaction, our system actively suggests data points to label such that the information gain is maximized. Further, we propose a new sparse active learning strategy that moves towards interactive label suggestion for large-scale databases.

To simplify parameterization, we ask the user to *rank* examples. This removes much of the burden when describing potentially abstract criteria on continuous scales. For instance, deciding that item 1 is more than, equal to, or less than item 2 is easier than quantifying that item 1 is 0.2 criteria units away from item 2. From this ranking, our semi-supervised approach generates continuous criteria, which become *sliders* in our user interface. For 2D criteria, users provide example embeddings directly such that the underlying relative locations are automatically meterized.

Our problem is not conventional ranking for data retrieval applications, where the goal is to classify all data instances that *match* the given query. Instead, we wish to regress *all* database items into continuous 1D and 2D criteria. We present two motivating scenarios: 1) For criteria which have no well-defined answer, our system helps users define their opinion. 2) For new visual databases with no metadata, where the alternative may be to laboriously hand collect labels, our system helps to quickly assess and organize the variation within the database.

We show the generality of our approach across databases of images of paintings and faces, and of geometries of human bodies and man-made objects (Fig. 1). We contribute:

1. The scenario of interactively defining continuous criteria from high-dimensional models, with a prototype implementation (please see our supplemental material and video).
2. A maximally-informative efficient semi-supervised active label suggestion algorithm.

2 Related Work

Re-ranking. This related problem takes the results of context- or text-based search and refines the query and/or retrieved result with user interaction. Personalized faceted search exploits relevant meta-data and suggests new keywords to refine the current search [19]. User behaviour is modeled probabilistically and tuned to maximize the expected *utility* of the facet. A rich body of literature shows the importance of re-ranking in search [19]; however, existing algorithms in this context focus on *maximizing search efficiency* rather than organizing databases along criteria.

Jain and Varma assume click behaviour relates to the interest query, and use a click count model to predict relevant rankings [17]. Zha et al. propose a similar approach for visual query suggestion [36]. The COPE system interactively refines search queries by users stating

whether the results match their information need, which then weights image features for future searches [2]. These approaches re-rank imagery based on user confirmation, and so they typically focus on the discrete problem of whether the retrieved results are a match [19]. Our goal is to learn criteria outright from sparse user labels and a high-dimensional model.

Rank learning. Existing rank learning algorithms typically use supervised learning from labeled data points (e.g., rank support vector machines; RankSVMs), whereas in our interactive setting, the user starts with no labels. As such, we exploit information in the unlabeled data with semi-supervised learning [6, 38], which has been used in rank aggregation [7] though not in our rank propagation case. Semi-supervised learning approaches often rely heavily on graph Laplacian regularization, in which data points are connected to their k-nearest neighbors with edges weighted by the input similarity (see supplemental for an introductory explanation). As a first-order regularizer, this has been shown to be unsuitable for learning continuous functions on high-dimensional manifolds [22], and so we build upon an approach which overcomes this limitation [18]. Szummer and Yilmaz [30] apply the graph Laplacian to the orthogonal problem of learning a preference criteria, and our learning approach could improve this application.

For data retrieval, Parikh and Grauman [24] learn discrete ranking functions via RankSVM from existing user labels. This restricts exploration to known criteria, whereas we discover criteria interactively. Murray et al. [21] presented a database for visual analysis that is characterized by abstract 'aesthetic' features, while Caicedo et al. [5] exploited user preference for image enhancement. Reinert et al. [27] use interaction to visually arrange a small image database into an aesthetic overview, which is orthogonal to the efficient exploration that we pursue.

Our interactive criteria definition on high-dimensional data does not compare directly to existing supervised criteria learning systems. CueFlick [1, 12] learns on binary labels, and WhittleSearch [24] attempts to re-rank data along existing criteria rather than generate criteria from scratch. We improve upon WhittleSearch's underlying RankSVM techniques when adapted to our scenario (Sec. 3.2).

Active learning. Chen et al. [8] apply active learning to remove inconsistency from existing crowdsourced labels. Their non-interactive approach is approximate in information gain, while our interactive approach is exact given model assumptions. Shen and Lin [28] essentially use RankSVM for bipartite ranking, with active learning based on single point and pair closeness. This is very similar to baseline predictive variance, which may accidentally pick uninformative outliers. Our new measure is unbiased by outliers. Our active learning approach is complementary to human-in-the-loop active learning approaches, e.g., Branson et al. [4]. Their task is to select *features* for a given data point which minimize class conditional distribution uncertainty (e.g., 20 questions game). Our problem is to suggest *data points* to label. Fogarty et al. [12] learn image retrieval criteria from binary labels, e.g., outdoor vs. indoor, by iterative refinement of distance measures between data points. Their active label suggestion was extended by Amershi et al. [1] by adopting a Gaussian process model on distance measures. We ask users to provide rank labels, and increase performance over Amershi et al. (Sec. 3.2).

Concept embedding. Our approach can be interpreted as using interaction to embed data into a high-level concept space. Existing work in this area focuses on category- or cluster-level supervision. Wilber et al. [35] receive triplet constants from users ((i,j,k): object i should be closer to object j than it is to k) to learn pair-wise similarity kernels that are used in t-SNE-type embedding [10]. We ask users to provide rank labels and emphasize continuous parameterization.

3 Semi-supervised criteria learning

From a large database of images or geometry, we compute offline a high-dimensional vector of 2D and 3D features (Sec. 4). We assume that some combination of these are sufficient to describe the desired user criteria, with our problem being to learn the criteria from a very small number of samples. We use semi-supervised learning to compensate for the lack of labeled data points by exploiting the *rich structural information* contained within unlabeled data points. Given this preprocess, the user produces a ranking (with equality) for a subset of the database items with our interface (supplemental video). The labels for these ranked examples are assigned an internal continuous representation ranging from -1.0 to 1.0.

Formally, for the set of data points $\mathcal{X} = \{X_1,...,X_u\}$, plus the corresponding *labels* for the first l data points, $\mathcal{Y} = \{Y_1,...,Y_l\} \subset \mathbb{R}$, where $l \ll u$, the goal of semi-supervised learning is to infer or propagate to the labels of the remaining $u-l$ data points in $\mathcal{X}$. We adopt the standard energy minimization approach [6, 37, 39]:

$$E(\mathbf{f}) = (\mathbf{f}-\mathbf{y})^\top L(\mathbf{f}-\mathbf{y}) + \lambda \mathbf{f}^\top H \mathbf{f}, \tag{1}$$

where L is a diagonal label indicator matrix and $\mathbf{y}$ is a vector of continuous label values: $L_{[i,i]} = 1$ and $\mathbf{y}_i = Y_i$ if i-th data point is labeled, and $L_{[i,i]} = 0$ and $\mathbf{y}_i = 0$, otherwise. H is the *regularization matrix* which quantifies how to *smooth* the propagated labels $\mathbf{f}$ within their local context. The regularization hyper-parameter λ balances between the smoothness of $\mathbf{f}$ and the deviation from the labels $\mathbf{y}$. In general, for a symmetric non-negative definite matrix H, the energy functional E is convex with respect to $\mathbf{f}$. The solution $\mathbf{f}^*$ is then explicitly given as:

$$\mathbf{f}^* = (L+\lambda H)^{-1} L\mathbf{y}. \tag{2}$$

Our approach is based on the *local Gaussian* (LG) regularizer for H [18] as it is designed to regularize continuous outputs. Our supplemental material discusses why we use this regularizer over the well-established graph Laplacian regularizer.

3.1 Interactive ranking and active label suggestion

In interactive ranking, the user iteratively provides training labels until they are happy with the learned criteria. Naïvely using the LG regularizer is computationally expensive for large databases since, at each iteration, obtaining the propagated labels $\mathbf{f}$ requires minimizing E in Eq. 1, which is order $O(u^3)$ complexity and requires solving a linear system of size $u \times u$. Further, we wish to aid the user by suggesting data points to label. At iteration t, we wish to estimate criteria uncertainty per data point from existing labels, and present the user with more informative samples to label at time $t+1$.

Following the analogy between regularized empirical risk minimization and maximum a posteriori (MAP) estimation [26], and by adopting Bayesian optimization [29], we reformulate $-1 \times E$ (Eq. 1) as (the logarithm of) a product of the prior $p(\mathbf{f})$ and a Gaussian noise model $p(\mathbf{y}|\mathbf{f})$. This leads us to assess the minimizer of E as the mean of the predictive distribution (the *posterior*):

$$-\log p(\mathbf{f}|\mathbf{y}) = (\mathbf{m}-\mathbf{f})^\top C^{-1} (\mathbf{m}-\mathbf{f}) + Z, \tag{3}$$

with mean $\mathbf{m} = CL\mathbf{y}$, covariance matrix $C = (L+\lambda H)^{-1}$, and the normalization constant Z. This perspective informs a *predictive uncertainty* for each data point: The i-th diagonal component C_{ii} of the covariance matrix C contains information on the uncertainty of the prediction on label $\mathbf{f}_i$, which is typically low when X_i is labeled and is high otherwise.

One simple and well-established strategy to exploit these modeled uncertainties for active label selection is to predict at each iteration t the point X_i which has the largest uncertainty. However, Figure 4 shows that naïvely choosing data points with maximum uncertainty leads

to poor performance with a higher error rate than random selection, as isolated outlier data points—which are not broadly informative—receive high variances and are chosen.

Instead, we construct the candidate data points that minimize the predictive variance over the *entire* set of data points. At each time step t, we choose data points with the highest average *information gain* $\mathcal{I}$, defined as:

$$\mathcal{I}(X_i) = \sum_{j=1,\ldots,u} [C(t-1) - C(t)^i]_{jj}. \tag{4}$$

The matrix $C(t)^i$ is constructed by adding 1 to the i-th diagonal element of $C(t-1)^{-1}$ and inverting it (i.e., i-th data point is regarded as labeled; see Eq. 1).

Naïvely estimating the information gain for all data points requires quadratic computational complexity: One has to estimate the minimizer of $E(\mathbf{f})$ (Eq. 1), which is $O(u^3)$ for each data point. However, in our iterative label suggestion scenario, $\mathcal{I}$ can be efficiently computed in linear time: Assuming that $C(t-1)$ is given from the previous step, calculating $\mathrm{diag}[C(t)^i]$ does not require inverting the matrix $L(t)+\lambda H$: Using *Sherman–Morrison–Woodbury* formula, $C(t)^i$ can be efficiently calculated from $C(t-1)$:

$$\mathrm{diag}[C(t)^i] = \mathrm{diag}[C(t-1)] - \frac{\mathrm{squ}[C(t-1)_{[:,i]}]}{1+\mathrm{diag}[C(t-1)]_i}, \tag{5}$$

where $A_{[:,j]}$ denotes a vector formed from the j-th column of the matrix A, $\mathrm{diag}[A]$ constructs a vector from the diagonal components of matrix A, and $\mathrm{squ}[B]$ is a vector obtained by taking element-wise squares of the vector B. Accordingly, after explicitly calculating $C(0)$, subsequent updates in $C(t)$ and $C(t)^i$ can be performed efficiently for each iteration t.

For large-scale problems where explicitly calculating the covariance matrix $C = (L+\lambda H)^{-1}$ is infeasible, we adopt a sparse eigen-decomposition-based approximation of $L+\lambda H$:

$$C^{-1} = L+\lambda H = E^F \Lambda^F E^{F\top} \Leftrightarrow C \approx \overline{C} = E\Lambda^{-1}E^\top, \tag{6}$$

where matrix E^F stores eigenvectors column-wise, and Λ^F is a diagonal matrix of the corresponding eigenvalues. E and Λ store the first r eigenvectors and eigenvalues, respectively, assuming that they are arranged in increasing eigenvalues. In this case, the corresponding information gain is given as:

$$[\overline{C}(t-1) - \overline{C}(t)^i]_{kk} = \left(\frac{E_{[k,:]}\Lambda^{-1}E^\top_{[i,:]}E_{[i,:]}\Lambda^{-1}E^\top_{[k,:]}}{1+H^{-1}_{ii}} \right). \tag{7}$$

At step t, adding a label to the $i(t)$-th data point leads to a new covariance estimate:

$$[EF_iE^\top]_{kk} = E_{[k,:]}\Lambda^{-1}[k,:]^\top + \sum_{i=1,\ldots,t} E_{[k,:]}\mathbf{r}_i\mathbf{r}_i^\top E^\top_{[k,:]}, \tag{8}$$

with $\mathbf{r}_i = E_{[i,:]}\Lambda^{-1}/(1+\overline{C}^{-1}_{ii})$. Please see our supplemental material for more details.

3.2 Evaluation

1D criteria learning. We evaluate our interactive LG adaptation on user rank data (evaluation on objective measures can be found in Kim et al. [18]). We compare against three techniques: RankSVM [24, 28], and 'forced binary' versions of RankSVM and our approach where labels are set either to 1 or -1 to simulate a simpler yes/no interaction. We compare over increasing numbers of randomly-chosen labels, with the remaining data points used as unlabeled examples (10 trials, averaged). Our approach improves performance over both baselines once the number of labels surpasses 20 (Fig. 2). While our method is designed to estimate continuous sliders, in supplemental material we compare our regularizer in the data retrieval setting.

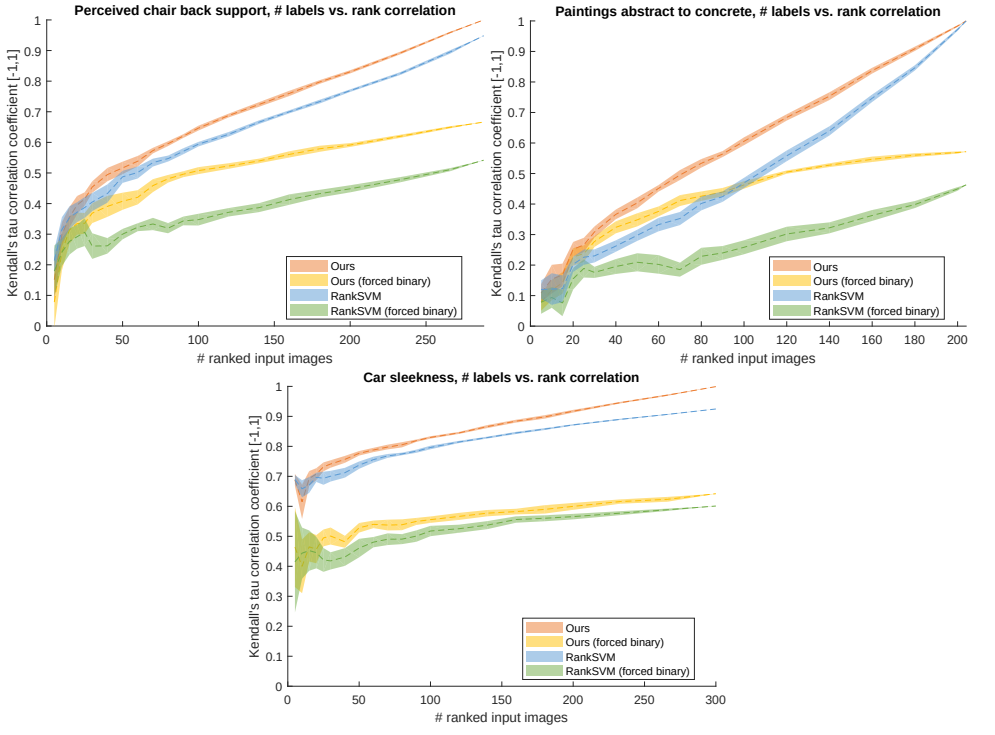

Figure 2: 1D criteria learning with varying numbers of labels over 10 random trials (dotted line is the mean; faded region is the standard error set at 95% confidence level). For fairness, each technique underwent hyper-parameter optimization to achieve the best result. Our approach is more successful once the number of labels rises above ≈20. *Top left:* Criteria 'perceived chair back support' on 288 chair geometries (Fig. 5). *Top right:* Criteria 'abstract to concrete' on 201 painting images (Fig. 7). *Bottom left:* Criteria 'sleekness' on 300 car geometries (Fig. 6)

2D criteria learning. For expert users, we can relax the ranking interface convenience and allow 2D criteria via direct positioning. Training labels are positioned in a 2D unit domain, effectively defining a geometric slider space. The labels are 25 faces selected randomly from 2,000 face images of a single person [32], describing horizontal and vertical rotations (Fig. 3). Objectively assigning pose angle is difficult, and so the labels are perceptual approximations. Compared to RankSVM, our embedding better reproduces user intention with a more even spread over the output space. Since RankSVM does not enable users to define a 'geometry' in parameter space, coordinating more than one parameter in this way can be challenging.

Active label suggestion. We compare our performance to 1) random label selection, 2) predictive uncertainty, and 3) Amershi et al. [1] adapted to our semi-supervised setting (Fig. 4). Over 10 trials on the CAESAR database (Sec. 4), we randomly selected a set $\mathcal{A}$ of 2,000 data points, with two intial labels $\mathcal{B} \subset \mathcal{A}$, and train on $\mathcal{B}$. Then, we calculate information gain $\mathcal{I}$ (Eq. 4) for each data point in $\mathcal{A} \backslash \mathcal{B}$. The best data point is assigned as a label, and we iterated until $|\mathcal{B}| = 50$.

Predictive variance only resulted in suggesting outliers, which led to worse results than random selection. The adapted algorithm of Amershi et al. improves upon random selection, while our new algorithm shows further improvement, especially when the number of labels is low. The computational complexity of Amershi et al. and ours are equal. We also demonstrate

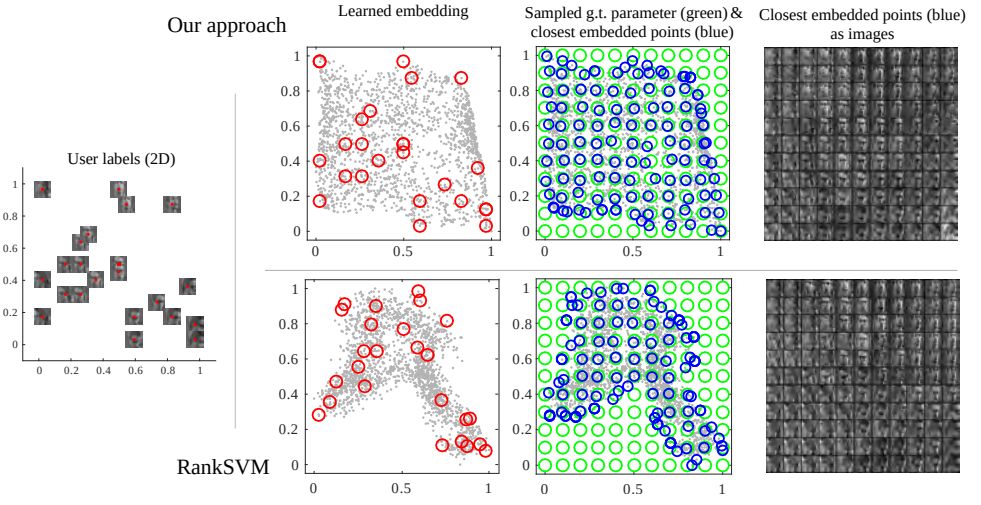

Figure 3: A user labels 25 face images by rotation angles (far left) to learn a criteria over 2,000 images. *Top:* Our approach. *Bottom:* RankSVM. *Left to right:* 1) learned embedding, showing labeled points as red circles, with our approach better maintaining the spatial layout. 2) Sampling ground truth labeled points uniformly (green circles) to see where their embedded points lie (blue circles), with our approach better reproducing the original uniform sampling. 3) Corresponding images to the blue circles in 2). Please zoom for detail.

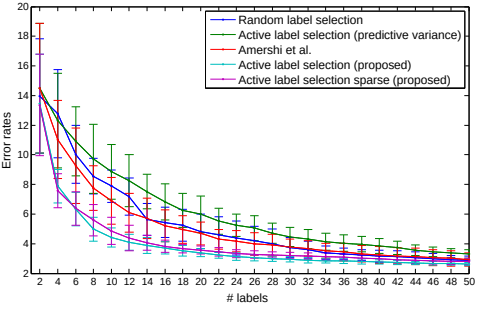

Figure 4: Active label selection performance (automatically picking the top label suggestion) as mean absolute error vs. ground truth for learning *body weight* as a criteria on the CAESAR database. # labels is subsamped by 2 for display. Error bar lengths correspond to twice the standard deviations; please note the smaller bars of our proposed full and sparse methods. Please zoom for detail.

that our proposed sparse eigen-decomposition-based approximation produces comparable results to the full information-gain-based method (Eqs. 4 and 5). We used only the first $r = 100$ eigenvectors. Calculating the selected eigenvectors of a sparse matrix does not require explicitly generating a dense matrix of size $u \times u$, as the eigen-decomposition can be efficiently updated after an initial pre-interaction computation (Eq. 8).

Computation complexity. This depends on the number of data points u, the number of nearest neighbors k in building the LG regularizer [13], the rank of the sparse approximation r (Eq. 6), and the number of non-zeros entries in the resulting regularization matrix H. This lies in-between $O(uk)$ and $O(uk^2)$, depending on the well-behavedness of neighborhoods ($O(uk^2)$ is random neighbors). H is built once per database as a preprocess. The dimensionality of the data model affects only the construction of the regularizer. At interaction time, complexity depends only on the number of data points. For the incremental step, we randomly select 1,000 candidate data points among u points, and choose the one which maximizes the information gain (Eq. 7).

On CAESAR, with $u = 4{,}258$ and $k = 20$, the preprocess takes 14 seconds. Solving the system took 0.5 seconds for standard batch *LG* approach. For active label suggestion, in

non-incremental batch LG learning, estimating the predictive variance for each of the 4,258 data points requires re-training the entire system per data point, which takes ≈ 35 minutes. In contrast, our algorithm enables suggesting the best data point across all examples in 1.5 seconds. All timings were on an Intel Xeon 3GHz CPU in MATLAB.

Let us consider a larger 60,000 item database—we will use MNIST purely for its size. With $k = 10$, the preprocess takes ≈ 10 minutes, with 5 seconds per solve. This database size requires our sparse eigen-decomposition-based approach to suggest labels, and this takes about 2 seconds per label. Generally, only ≈ 5 labels are needed for suggestion, and so this still allows interaction (if slower). Naïve information gain calculation (Eq. 4) took ≈ 4 hours, while (dense) incremental selection (Eq. 5) is infeasible due to the prohibitively large memory requirement.

50k+ item databases. As discussed, our system is no longer interactive in these situations. However, often we can subsample a large database down to a few thousand items which capture the major variation within. One way to accomplish this would be to use our active learning suggestion as a preprocess: as it only relies on whether X_i is regarded as labeled and not on how it was labeled in the rank, we can accept the top suggestion and repeat until the desired n-sized subset is reached. Then, the user can rank this smaller subset to define their criteria. Once defined, we can use the subset as labels to propagate the criteria to the larger database.

Parameters. As default parameters, we set regularization weight λ to 10^{-6} and nearest neighbors k to 20. The estimated dimensionality of the underlying high-dimensional manifold m in feature vector space varies between 5–20 with database size. Parameters can vary per database. A hyper-parameter search against a test set is at odds with many motivating applications, e.g., initially assessing a just-collected database. However, in some applications the regularizer need only be computed once per database with parameters set by a 'database curator', after which end users may interactively define any number of criteria.

Rank usefulness. With 28 participants, we evaluated how Kendall's Tau rank correlation coefficient (KT) relates to perceived rank usefulness. With our system, we generated example rankings of 50 items at different KT values, along with an ideal ranking. We asked users to rate the usefulness of our rank given an ideal rank, both on a 7-point scale and absolutely (useful/not useful). Participants assessed 24 rankings split evenly across three object databases (Sec. 4). Assuming our scale data to be continuous, the relationship of usefulness (y) to KT (x) was linear as $y = 4.8x + 1.3$, where 70% of participants claimed a produced rank was useful at $KT = 0.62$. We can cross-reference this with our performance measures per database and critera (Fig. 2) to discover the number of labels required for useful criteria: ≈75 for chairs, ≈100 for art, and ≈10 for cars.

Human ranking time. Ranking time is database and criteria dependent as harder decisions take longer. The relationship between rank length and rank time is not linear—the more labels ranked, the longer it takes to rank. Our current prototype interactive system requires ≈2 minutes for 20 labels, and ≈8–10 minutes for 50 labels. An efficient design of an interface specific to our use case is a more serious human-computer interaction problem than our paper scope includes.

Human variation. Some criteria are inherently ambiguous. To investigate human variance in criteria description, we asked seven users to rank 201 paintings along the criteria 'abstract to concrete' (Sec. 4; see supplemental for details). Participants performed this task by hand, not using our system, and it took on average 158 minutes per participant. We compute KT between each pair of user rankings. The mean coefficient is (coincidentally also) 0.619, with standard deviation of 0.043: there is only moderate agreement between participants. This shows the difficulty of the task and the need for an interactive ranking system which can more easily create personal criteria. However, at least for this task, it provides us evidence that a system should aim to achieve this level of performance to reach this 'agreement level' among participants.

Figure 5: Result at 20 labels. *Top:* Rank preference (≈2 minutes). *Bottom:* Rank propagated to 288 data points (raster order). While there are occasional outliers, we maintain the trend of chairs with high backs, then chairs with arms, and then lounge chairs, in this highly variable database.

Figure 6: 300 items from the cars 3D object database, organized by user criteria 'sleekness', via our approach, with the 20 labels provided at the top.

4 Databases and Example Applications

Object geometry. We use data from Hua et al. [16]: 5,000 chair, 3,000 airplane, and 1,700 car geometries are downloaded from Trimble 3D Warehouse, then oriented and scaled to align object features, then locally deformed to better align shared elements, e.g., seat heights for chairs. Once normalized, we voxelize these spaces and extract, for each voxel, the shortest distance to the nearest mesh point to create a distance field which captures the shape variation of each example. Figures 1, 5, and 6 show examples. *Related applications*: Shape exploration [16] and synthesis [23] scenarios, especially for large Web collections.

Human geometry. The CAESAR database contains 4,258 human 3D scans, along with ground-truth caliper body measurements. We fit a statistical model to each of the scans [25], with which we describe body variations from an abstract 20-dimensional linear shape basis (see supplemental video). *Related applications*: Separating the semantically-coupled axes found by dimensionality reduction, e.g., for body [15], face [33], or cloth [14] statistical shape models.

Painting images. We collected 19,808 paintings covering periods until 1930 (`www.zeno.org/kunst`). As a feature vector, we follow Gatys et al. [13] and use the pre-trained VGG 19-layer convolutional neural network to estimate a set of style matrices, based on computing Gram matrices from the neural response in the first five layers. Figure 7 depicts a subset of the database after 201 labels were provided for the criteria 'abstract to concrete'. *Related applications*: Arranging cultural heritage or historical artifacts by style/genre [9, 34].

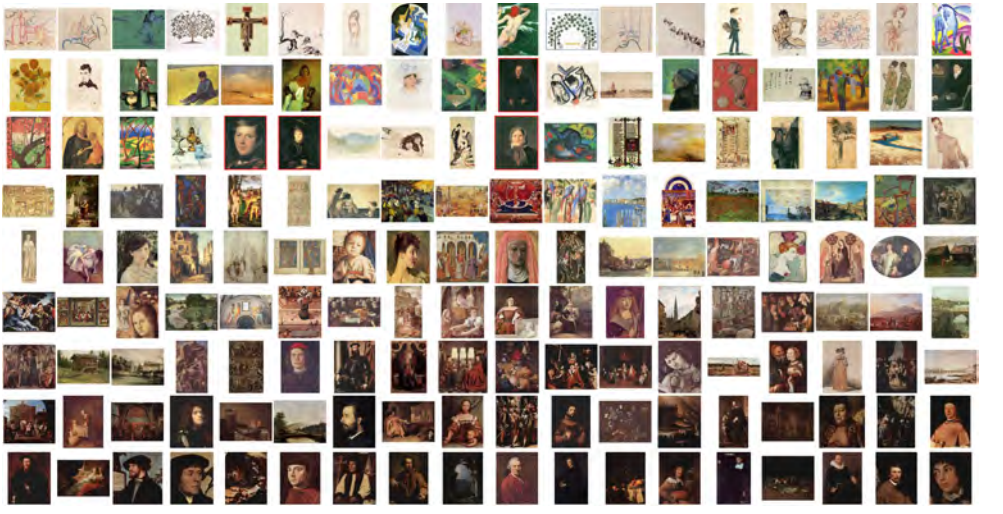

Figure 7: A 162-point subset of 19,808 paintings ordered by the criteria 'abstract to concrete', sampled evenly along the output interval $[-1.2, 1.2]$. Raster order. The images generally increase in realism, with outliers marked in red. While the trend appears dominated by brightness, upon closer inspection (please zoom) the detail is apparent: darker abstract portraits are nearer the top, and brighter concrete landscapes nearer the bottom.

Face images. We fit an active appearance model to the Labeled Face Parts in the Wild image database, and extract shape and appearance PCA coefficients for features [3, 31]. Figure 1 shows 10 faces ranked by smile expression to recover a slider for 417 faces. *Related applications*: Face appearance ranking [20], such as personalized attractiveness scales [11], or in policing to aid suspect identification by a witness ranking database examples by 'more/less like him/her'.

5 Conclusion

Interactive database exploration is a difficult problem, and a lack of labels requires efficient analysis via semi-supervised learning. To achieve this, we introduced an incremental version of the LG regularizer, which has superior performance to RankSVM. This enables us to guide the user via a new active label suggestion method, which estimates the information gain of labeling a particular data point. Our approach is sparse to enable LG active learning for large databases. We show an interactive system which uses rank labels to quickly and easily create criteria sliders, and demonstrate its use across databases of images and geometry.

Acknowledgments. We thank Qi-Xing Huang, Leonid Pishchulin, Thomas Helten, and all of our study participants, particularly Atsunobu Kotani, Frances Chen, Gary Chien, Numair Khan, and Eleanor Tursman. Kwang In Kim thanks EPSRC EP/M00533X/2; James Tompkin and Hanspeter Pfister thank the DARPA Memex program.

References

[1] S. Amershi, J. Fogarty, A. Kapoor, and D. S. Tan. Effective end-user interaction with machine learning. In *Proc. AAAI*, 2011.

[2] B. Balcer, M. Halvey, J. M. Jose, and S. A. Brewster. COPE: interactive image retrieval using conversational recommendation. In *Proc. BCS-HCI*, pages 1–10, 2012.

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proc. IEEE CVPR*, 2011.

[4] S. Branson, G. Horn, C. Wah, P. Perona, and S. Belongie. The ignorant led by the blind: a hybrid human-machine vision system for fine-grained categorization. *IJCV*, 108(1-2): 3–29, 2014.

[5] J. C. Caicedo, A. Kapoor, and S. B. Kang. Collaborative personalization of image enhancement. In *Proc. IEEE CVPR*, pages 249–256, 2011.

[6] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[7] S. Chen, F. Wang, Y. Song, and C. Zhang. Semi-supervised ranking aggregation. In *Proc. ACM CIKM*, pages 1427–1428, 2008.

[8] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proc. ACM WSDM*, pages 193–202, 2013.

[9] M. Culjak. Classification of art paintings by genre. In *Proc. MIPRO*, pages 1634–1639, 2011.

[10] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, pages 2579–2605, 2008.

[11] Y. Eisenthal, G. Dror, and E. Ruppin. Facial attractiveness: beauty and the machine. *Neural Computation*, 18(1):119–142, 2006.

[12] J. Fogarty, D. Tan, A. Kapoor, and S. Winder. CueFlik: interactive concept learning in image search. In *Proc. CHI*, pages 29–38, 2008.

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE CVPR*, June 2016.

[14] P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. J. Black. Drape: Dressing any person. *ACM TOG (Proc. SIGGRAPH)*, 31(4):35:1–35:10, July 2012.

[15] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 28(2):337–346, 2009.

[16] Q.-X. Huang, H. Su, and L. Guibas. Fine-grained semi-supervised labeling of large shape collections. *ACM TOG*, 32(6):190:1–10, 2013.

[17] V. Jain and M. Varma. Learning to re-rank: query-dependent image re-ranking using click data. In *Proc. WWW*, pages 277–286, 2011.

[18] K. I. Kim, J. Tompkin, H. Pfister, and C. Theobalt. Local high-order regularization on data manifolds. In *Proc. IEEE CVPR*, pages 5473–5481, 2015.

[19] J. Koren, Y. Zhang, and X. Liu. Personalized interactive faceted search. In *Proc. WWW*, pages 477–486, 2008.

[20] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-driven enhancement of facial attractiveness. *ACM TOG (Proc. SIGGRAPH)*, 27(3):38:1–38:9, 2008.

[21] N. Murray, L. Marchesotti, and F. Perronnin. Ava: a large-scale database for aesthetic visual analysis. In *Proc. IEEE CVPR*, pages 2408–2415, 2012.

[22] B. Nadler, N. Srebro, and X. Zhou. Statistical analysis of semi-supervised learning: the limit of infinite unlabelled data. In *NIPS*, pages 1330–1338, 2009.

[23] M. Ovsjanikov, W. Li, L. Guibas, and N. J. Mitra. Exploration of continuous variability in collections of 3d shapes. *ACM TOG (Proc. SIGGRAPH)*, 30(4):33:1–33:10, 2011.

[24] D. Parikh and K. Grauman. Relative attributes. In *Proc. ICCV*, pages 503–510, 2011.

[25] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 2017.

[26] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[27] B. Reinert, T. Ritschel, and H.-P. Seidel. Interactive by-example design of artistic packing layouts. *ACM TOG (Proc. SIGGRAPH Asia)*, 32(6):218:1–218:7, 2013.

[28] W.-Y. Shen and H.-T. Lin. Active sampling of pairs and points for large-scale linear bipartite ranking. In *Proc. ACML*, pages 388–403, 2013.

[29] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *NIPS*, pages 2951–2959, 2012.

[30] M. Szummer and E. Yilmaz. Semi-supervised learning to rank with preference regularization. In *Proc. ACM CIKM*, pages 269–278, 2011.

[31] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *Proc. ICCV*, 2013.

[32] J. J. Verbeek and N. Vlassis. Gaussian fields for semi-supervised regression and correspondence learning. *Pattern Recognition*, 39(10):1864–1875, 2006.

[33] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM TOG (Proc. SIGGRAPH)*, 24(3):426–433, 2005.

[34] C. Wallraven, R. Fleming, D. Cunningham, J. Rigauc, M. Feixas, and M. Sbert. Categorizing art: Comparing humans and computers. *Computers & Graphics*, 33(4): 35:1–35:10, 2009.

[35] M. J. Wilber, I. S. Kwak, D. Kriegman, and S. Belongie. Learning concept embeddings with combined human-machine expertise. In *Proc. ICCV*, pages 981–989, 2015.

[36] Z.-J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T.-S. Chua, and X.-S. Hua. Visual query suggestion: Towards capturing user intent in internet image search. *ACM TOMM*, 6(13): 13:1–13:19, 2010.

[37] D. Zhou, O Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 1330–328, 2004.

[38] X. Zhu. Semi-supervised learning literature survey. Technical Report TR 1530, Department of Computer Science, University of Wisconsin–Madison, 2008.

[39] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.