A Structured Approach to Predicting Image Enhancement Parameters

Parag Shridhar Chandakkar Baoxin Li
School of Computing, Informatics and Decision Systems Engineering, Arizona State University
{pchandak,baoxin.li}@asu.edu

Abstract

Social networking on mobile devices has become a commonplace of everyday life. In addition, photo capturing process has become trivial due to the advances in mobile imaging. Hence people capture a lot of photos everyday and they want them to be visually-attractive. This has given rise to automated, one-touch enhancement tools. However, the inability of those tools to provide personalized and content-adaptive enhancement has paved way for machine-learned methods to do the same. The existing typical machine-learned methods heuristically (e.g. kNN-search) predict the enhancement parameters for a new image by relating the image to a set of similar training images. These heuristic methods need constant interaction with the training images which makes the parameter prediction sub-optimal and computationally expensive at test time which is undesired. This paper presents a novel approach to predicting the enhancement parameters given a new image using only its features, without using any training images. We propose to model the interaction between the image features and its corresponding enhancement parameters using the matrix factorization (MF) principles. We also propose a way to integrate the image features in the MF formulation. We show that our approach outperforms heuristic approaches as well as recent approaches in MF and structured prediction on synthetic as well as real-world data of image enhancement.

1. Introduction

The growth of social networking websites such as Facebook, Google+, Instagram etc. along with the ubiquitous mobile devices has enabled people to generate multimedia content at an exponentially increasing rate. Due to the easy-to-use photo-capturing process of mobile devices, people are sharing close to two billion photos per day on the social networking sites [1]. People want their photos to be visually-attractive which has given rise to automated, one-touch enhancement tools. However, most of these tools are pre-defined image filters which lack the ability of doing content-adaptive or personalized enhancement. This has fueled the development of machine-learning based image enhancement algorithms.

Many of the existing machine-learned image enhancement approaches first learn a model to predict a score quantifying the aesthetics of an image. Then given a new low-quality image[2], a widely-followed strategy to generate its enhanced *version* is as follows:

- Generate a large number of candidate enhancement parameters[3] by densely sampling the entire range of image parameters. Computational complexity may be reduced by applying heuristic criteria such as, densely sampling only near the parameter space of most similar training images.
- Apply these candidate parameters to the original low-quality image to create a set of candidate images.
- Perform feature extraction on every candidate image and then compute its aesthetic score by using the learned model.
- Present the highest-scoring image to the user.

There are two obvious drawbacks for the above strategy. First, generating and applying a large number of candidate parameters to create candidate images may be computationally prohibitive even for low-dimensional parameters. For example, a space of three parameters where each parameter $\in \{0, ..., 9\}$ produces 10^3 combinations. Second, even if creating candidate images is efficient, extracting features from them is always computationally intensive and is the bottleneck. Also, such heuristic methods need constant interaction with the training database (which might be

[1] http://www.kpcb.com/internet-trends

[2] We call the images before enhancement as low-quality and those after enhancement as high-quality in the rest of this article. The process of enhancing a new image is called "the testing stage".

[3] The brightness, saturation and contrast are referred to as "parameters" of an image in this article.

stored on a server) that makes the parameter prediction sub-optimal. All these factors contribute to making the testing stage inefficient.

Our approach assumes that a model quantifying image aesthetics has already been learned and instead focuses on finding a structured approach to enhancement parameter prediction. During training, we learn the inter-relationship between the low-quality images, its features, its parameters and the high-quality enhancement parameters. During the testing stage, we only have access to a new low-quality image, its features, parameters and the learned model and we have to predict the enhancement parameters. Using these enhancement parameters, we can generate the candidate images and select the best one using the learned model. The stringent requirement of not accessing the training images arises from real-world requirements. For example, to enhance a single image, it would be inefficient to establish a connection with the training database, generate hundreds of candidate images, perform feature extraction on them and then find the best image.

The search space spanned by the parameters is huge. However, the enhancement parameters are not randomly scattered. Instead they depend on the parameters and features of the original low-quality image. Thus we hypothesize that the enhancement parameters should have a low-dimensional structure in another latent space. We employ an MF-based approach because it allows us express the enhancement parameters in terms of three latent variables, which model the interaction across: 1. the low-quality images 2. their corresponding enhancement parameters 3. the low-quality parameters. The latent factors are learned during inference by Gibbs sampling. Additionally, we need to incorporate the low-quality image features since the enhancement parameters also depend on the color composition of the image, which can be characterized by the features. The feature incorporation in this framework is achieved by representing the latent variable which models the interaction across these images as a linear combination of their features, by solving a convex $\ell_{2,1}$-norm problem. We review the related work on MF as well as image enhancement in the following section.

2. Related Work

Development of machine-learned image enhancement systems has recently been an active research area of immense practical significance. Various approaches have been put forward for this task. We review those works which improve the visual appearance of an image using automated techniques. To encourage research in this field, a database named MIT-Adobe FiveK containing corresponding low and high-quality images was proposed in [4]. The authors also proposed an algorithm to solve the problem of global tonal adjustment. The tone adjustment problem only manipulates the luminance channel, where we manipulate saturation, brightness and contrast of an image.

Content-based enhancement approaches have been developed in the past which try to improve a particular image region [2, 9]. These approaches require segmented regions which are to be enhanced. This itself may prove to be difficult. Approaches which work on pixels have also been developed using local scene descriptors. Firstly, similar images from the training set are retrieved. Then for each pixel in the input, similar pixels were retrieved from the training set, which were then used to improve the input pixel. Finally, Gaussian random fields maintain the spatial smoothness in the enhanced image. This approach does not consider the global information provided by the image and hence the enhancements may not be visually-appealing when viewed globally. In [8], a small number of image enhancements were collected from the users which were then used along with the additional training data.

Two recent works involving training a ranking model from low and high-quality images are presented in [5, 25]. The authors of [25] create a data-set of 1300 corresponding low and high-quality image pairs along with a record of the intermediate enhancement steps. A ranking model trained on this type of data can quantify the aesthetics of an image. In [5], non-corresponding low and high-quality image pairs extracted from the Web are used to train a ranking model. Both of these approaches use kNN-search during the testing stage to create candidate images. After extracting features and ranking them, the best image is presented to the user.

The task of enhancement parameter prediction could be related to the attribute prediction [17, 18, 11, 7]. However, the goal of the work on attribute prediction has been to predict relative strength of an attribute in the data sample (or image). We are not aware of any work which predicts parameters of an enhanced version of a low-quality image given only the parameters and features of that image. Since our approach is based on MF principles, we review the related recent work on MF.

MF [19, 15, 20, 10, 24] is extensively used in recommender systems [12, 1, 13, 23, 14, 22, 21]. These systems predict the rating of an item for a user given his/her existing ratings for other items. For example, in Netflix problem, the task is to predict favorite movies based on user's existing ratings. MF-based solutions exploit following two key properties of such user-item rating matrix data. First, the preferred items by a user have some similarity to the other items preferred by that user (or by other similar users, if we have sufficient knowledge to build a similarity list of users). Second, though this matrix is very high-dimensional, the patterns in that that matrix are structured and hence they must lie on a low-dimensional manifold. For example, there are $17,770$ movies in Netflix data and ratings range from $1-5$. Thus, there are 5^{17770} rating combinations possible

per user and there are $480,189$ users. Therefore, the number of actual variations in the rating matrix should be a lot smaller than the number of all possible rating combinations. These variations could be modeled by latent variables lying near a low-dimensional manifold. This principle is formalized in [15] with probabilistic matrix factorization (PMF). It hypothesizes that the rating matrix can be decomposed into two latent matrices corresponding to user and movies. Their dot product should give the user-ratings. This works fairly well on a large-scale data-set such as Netflix. However, a lot of parameters have to be tuned. This requirement is alleviated in [20] by developing a Bayesian approach to MF (BPMF). BPMF has been extended for temporal data (BPTF) in [24]. MF is used in other domains such as computer vision to predict feature vectors of another viewpoint of a person given a feature for one viewpoint [6]. We adopt and modify BPTF since it allows us to model joint interaction across low-quality images, corresponding enhancement parameters and the low-quality parameters. In the next section, we detail our problem formulation and proposed approach.

3. Problem Formulation and Proposed Approach

We have a training set consisting of N images $\{\mathbf{S}_1, \ldots, \mathbf{S}_N\}$[4]. Parameters of all images are represented as $\mathbf{A} = \{A_1, \ldots, A_N\}$ where $A_i \in \mathbb{R}^{K\times 1}\ \forall\ i \in \{1, \ldots, N\}$. Each image has M enhanced versions and each version has the same size as that of its corresponding low-quality image. All versions corresponding to the i^{th} image are represented as $\{\mathbf{W}_i^1, \ldots, \mathbf{W}_i^M\}$. All versions are of higher quality as compared to its corresponding image. Parameters of all M versions of the i^{th} image (also called as candidate parameters) are represented as $\mathbf{A'} = \{A_i'^1, \ldots, A_i'^M\}$, where $A'^j_i \in \mathbb{R}^{K\times 1}\ \forall\ i, j$. Features of all low-quality images are represented as $\mathbf{F} = \{F_1, \ldots, F_N\}$ where $F_i \in \mathbb{R}^{L\times 1}\ \forall\ i$. In practice, we observe that $M \ll N, K < M$. Our goal is to be able to predict the candidate parameters for all the versions of the i^{th} image by only using the information provided by A_i and F_i. To the best of our knowledge, this is a novel problem of real significance that has not been addressed in the literature. We now explain our proposed approach.

As mentioned before, our task is to predict the candidate parameters for all the enhanced versions of a low-quality image with the help of its parameters and features. The values for all the K parameters corresponding to N images and their $N \cdot M$ versions (total $N + N \cdot M$) can be stored in three-dimensional matrix $\mathbf{R} \in \mathbb{R}^{N\times(M+1)\times K}$. We need to predict $\hat{R}^k_{ij} = R^k_i + \Delta R^k_{ij}$ or in turn just ΔR^k_{ij}. R^k_i denotes the k^{th} parameter value ($k \in \{1, \ldots, K\}$) of the i^{th} low-quality image and $\hat{R}^k_{ij}$ is the k^{th} parameter value of j^{th} version of the i^{th} image. Given a new n^{th} low-quality image, we only need to predict $\Delta R^k_{nj}\ \forall\ j = \{1, \ldots, M\}, \forall\ k$.

During training, we can compute ΔR^k_{ij} from available R^k_{ij} and $\hat{R}^k_{ij}$. Following MF principles, we express $\mathbf{\Delta R}$ as an inner product of three latent factors, $\mathbf{U} \in \mathbb{R}^{D\times N}, \mathbf{V} \in \mathbb{R}^{D\times M}$ and $\mathbf{T} \in \mathbb{R}^{D\times K}$ [20, 24]. D is the latent factor dimension. These latent factors should presumably model the underlying low-dimensional subspace corresponding to the low-quality images, its enhanced versions and its parameters. This can be formulated as:

$$\Delta R^k_{ij} =< U_i, V_j, T_k >\equiv \sum_{d=1}^{D} U_{di} V_{dj} T_{dk}, \qquad (1)$$

where U_{di} denotes the d^{th} feature of the i^{th} column of $\mathbf{U}$. Presumably, as we increase D, the approximation error $\Delta R^k_{ij} - < U_i, V_j, T_k >$ should decrease (or stay constant) if the prior parameters for latent factors $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ are chosen correctly. Following [20], we choose normal distribution (with precision α) for: 1. the conditional distribution $\mathbf{\Delta R}|(\mathbf{U}, \mathbf{V}, \mathbf{T})$ and 2. for prior distributions - $p(\mathbf{U}|\Theta_U), p(\mathbf{V}|\Theta_V)$ and $p(\mathbf{T}|\Theta_T)$, where $\Theta_U = (\mu_U, \mathbf{\Lambda}_U^{-1})$, $\Theta_V = (\mu_V, \mathbf{\Lambda}_V^{-1})$, $\Theta_T = (\mu_T, \mathbf{\Lambda}_T^{-1})$. Θ_U, Θ_V and Θ_T are hyper-parameters, and μ and $\mathbf{\Lambda}$ are the multivariate precision matrix and the mean respectively. Since the Wishart distribution is a conjugate prior for multivariate normal distribution (with precision matrix), we put Gaussian-Wishart priors on all hyper-parameters[5]. We could find the latent factors $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ by doing inference through Gibbs sampling. It will sample each latent variable from its distribution, conditional on the values of other variables. The predictive distribution for ΔR^k_{ij} can be found by using Monte-Carlo approximation (explained later).

However, it is important to note the following major differences in our problem when compared with the previous work on MF [20, 24]. In product or movie rating prediction problems, an average (non-personalized) recommendation may be provided to a user who has not provided any preferences (not necessarily constant for all users). In our case, each image may require a different kind of parameter adjustment to create its enhanced version and thus no "average" adjustment exists. As explained before, the adjustment should depend on the image's features, which characterize that image (e.g. bright vs. dull, muted vs. vibrant). In our problem, it is particularly difficult to get a good generalizing performance on the testing set as we shall see later. The loss in performance of existing approaches on the testing set can

[4]We use bold letters to denote matrices. Non-bold letters denote scalars/vectors which will either be clear from the context or will be mentioned. $X^i, X_i, \mathbf{X}^T, X_{ij}$ and $||\mathbf{X}||_p$ denote row, column, transpose, entry at row i and column j of a matrix $\mathbf{X}$ and p^{th} norm of matrix $\mathbf{X}$ respectively.

[5]For details, see supplementary material on author's website.

be attributed to the different requirements for parameter adjustments for each image. Thus it becomes necessary to include the information obtained from image features into the formulation. We show that simply concatenating the parameters and features and applying MF techniques presented in [20, 24] does not provide good performance, possibly because they lie in different regions of the feature space.

To overcome this problem, we observe that the conditional distribution of each U_i factorizes with respect to the individual samples. We propose to express $\mathbf{U}$ as a linear function of $\mathbf{F}$ by using a convex optimization scheme. We then integrate it into the inference algorithm to find out the latent factors. The linear transformation can be expressed as,

$$U_i = F_i^T \mathbf{P} + Q, \ \forall \, i \in \{1, \ldots, N\}, \tag{2}$$

where $F_i \in \mathbb{R}^{L \times 1}, U_i \in \mathbb{R}^{D \times 1}, \mathbf{P} \in \mathbb{R}^{D \times D}$ and $Q \in \mathbb{R}^{1 \times D}$. Note that to carry out this decomposition, we have to set $D = L$. This is not a severe limitation since L is usually large (~ 1000) and as we have mentioned before, increasing D should decrease the approximation error at the cost of increased computation. Henceforth we assume that our feature extraction process generates $F_i \in \mathbb{R}^{D \times 1}$. Also, note that large L does not mean that the latent space is no longer low-dimensional, because L is still smaller as compared to all the possible combinations of parameters (e.g. 5^{17770}).

We propose an iterative convex optimization process to determine coefficients $\mathbf{P}$ and Q of Equation 2. We propose the following objective function to determine them: q

$$\min_{\mathbf{P},Q} \sum_{i=1}^{N} ||F_i^T \mathbf{P} + Q - U_i^T||_2 + \beta ||\mathbf{P}||_{2,1} + \gamma ||Q||_2 \tag{3}$$

The objective function tries to reconstruct $\mathbf{U}$ using $\mathbf{P}, Q$ and F while controlling the complexity of coefficients. Let's concentrate on the structure of $\mathbf{P}$ (by neglecting the effect of Q momentarily). The columns of $\mathbf{P}$ act as coefficients for F_i. Ideally, we would want the elements of U_i to be determined by a sparse set of features, which implies sparsity in the columns of $\mathbf{P}$. To this end, we impose $\ell_{2,1}$-norm on $\mathbf{P}$, which gives us a block-row structure for $\mathbf{P}$.

Let us consider the structure of Q along with $\mathbf{P}$. Equation 2 shows that different columns of U_i depend on different image features F_i. Also, we expect that a different set of columns of $\mathbf{P}$ will get activated (take on large values) for different F_i. We add an offset $Q \in \mathbb{R}^{1 \times D}$ for regularization. Thus the offset introduced by Q remains constant across all the images but changes for each $F_{i,j}$. Making Q to be a row vector also forces $\mathbf{P}$ to play a major role in Equation 3. This in turn increases the dependence of U_i on F_i. If we were to define Q as the same size of $\mathbf{U}$ (which would mean different offsets for each image as well as its features), it would pose two potential disadvantages. Firstly, optimal $\mathbf{P}$ and Q could be (trivially) obtained by just setting each entry of $\mathbf{P}$ to a very small value and letting a column of $Q \approx U_i$ (which makes F_i redundant). Secondly, while testing for a new image, we would have to devise a strategy to determine the suitable value for Q. For example, we could take the column of Q that corresponds to the nearest training image. This adds unnecessary complexity and reduces generalization. By making Q a row vector, we consider that it may be possible to arrive to the space of enhancement parameters by linearly transforming the low-quality image features with a constant offset. In other words, we want $\mathbf{P}$ to transform the features into a region in the latent space where all the other high-quality images lie and Q provides an offset to avoid over-fitting. This is a joint $\ell_{2,1}$-norm problem which can be solved efficiently by reformulating it as convex. We thus reformulate Equation 3 as follows, inspired by [16]:

$$\min_{\mathbf{P},Q} \frac{1}{\beta} \sum_{i=1}^{N} ||F_i^T \mathbf{P} + Q - U_i^T||_2 + ||\mathbf{P}||_{2,1} + \frac{\gamma}{\beta} ||Q||_2 \tag{4}$$

The $\ell_{2,1}$-Norm of a matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ is defined as, $\ell_{2,1}(\mathbf{X}) = \sum_{i=1}^{M} ||\mathbf{X}^i||_2$. Also, for a row vector Q, we have $||Q||_2 = ||Q||_{2,1}$. Thus Equation 4 can be further written as:

$$\min_{\mathbf{P},Q} \frac{1}{\beta} ||\mathbf{F}^T \mathbf{P} + 1^N Q - \mathbf{U}^T||_{2,1} + ||\mathbf{P}||_{2,1} + \delta ||Q||_{2,1}, \tag{5}$$

where $\delta = \frac{\gamma}{\beta}$ and 1^N is a column vector of ones $\in \mathbb{R}^N$. Now, put $\mathbf{F}^T \mathbf{P} + 1^N Q - \beta \mathbf{E} = \mathbf{U}^T$. Thus Equation 5 becomes:

$$\begin{aligned} &\min_{\mathbf{E},\mathbf{P},Q} \ ||\mathbf{E}||_{2,1} + ||\mathbf{P}||_{2,1} + \delta ||Q||_{2,1}, \\ &\text{s.t. } \mathbf{F}^T \mathbf{P} + 1^N Q - \beta \mathbf{E} = \mathbf{U}^T, \\ &\min_{\mathbf{E},\mathbf{P},Q} \left|\left| \begin{bmatrix} \mathbf{E} \\ \mathbf{P} \\ \delta Q \end{bmatrix} \right|\right|_{2,1} \text{s.t. } \begin{bmatrix} -\beta \mathbf{I}_N & \mathbf{F}^T & \delta^{-1} 1^N \end{bmatrix} \begin{bmatrix} \mathbf{E} \\ \mathbf{P} \\ \delta Q \end{bmatrix} = \mathbf{U}^T \end{aligned} \tag{6}$$

Equation 6 is now in the form of: $\min_{\mathbf{X}} ||\mathbf{X}||_{2,1}$ s.t. $\mathbf{Z}\mathbf{X} = \mathbf{B}$ and thus convex. It can be iteratively solved by an efficient algorithm mentioned in [16]. We set $\beta = 0.1$ and $\delta = 3$. Once we have expressed $\mathbf{U}$ as a function of $\mathbf{F}$, we use Gibbs Sampling to determine the latent factors $\mathbf{P}, Q, \mathbf{V}$ and $\mathbf{T}$ [20]. As mentioned before, the predictive distribution for a new parameter value $\Delta \hat{R}_{ij}^k$ is given by a multidimensional integral as:

Algorithm 1 Gibbs Sampling for Latent Factor Estimation

Initialize model parameters $\{\mathbf{P}^{(1)}, Q^{(1)}, \mathbf{V}^{(1)}, \mathbf{T}^{(1)}\}$. Obtain $\left(\mathbf{U}^{(1)}\right)^T = \mathbf{F}^T\mathbf{P}^{(1)} + Q^{(1)}$

For $y = 1, 2, \ldots, Y$

- Sample the hyper-parameters according to the derivations [6]:
  $\alpha^{(y)} \sim p(\alpha^{(y)}|\mathbf{U}^{(y)}, \mathbf{V}^{(y)}, \mathbf{T}^{(y)}, \Delta\mathbf{R})$,
  $\Theta_U^{(y)} \sim p(\Theta_U^{(y)}|\mathbf{U}^{(y)})$, $\Theta_V^{(y)} \sim p(\Theta_V^{(y)}|\mathbf{V}^{(y)})$, $\Theta_T^{(y)} \sim p(\Theta_T^{(y)}|\mathbf{T}^{(y)})$
- For $i = 1, ..., N$, sample the latent features of an image (in parallel):
  $U_i^{(y+1)} \sim p(U_i|\mathbf{V}^{(y)}, \mathbf{T}^{(y)}, \Theta_U^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$

  Determine $\mathbf{P}^{(y+1)}$ and $Q^{(y+1)}$ using the iterative optimization by substituting $\mathbf{B} = \left(\mathbf{U}^{(y+1)}\right)^T$.

Reconstruct $\mathbf{U}^{(y+1)}$: $\left(\hat{\mathbf{U}}^{(y+1)}\right)^T = \mathbf{F}^T\mathbf{P}^{(y+1)} + Q^{(y+1)}$

- For $j = 1, ..., M$, sample the latent features of the enhanced versions (in parallel):
  $V_j^{(y+1)} \sim p(V_j|\hat{\mathbf{U}}^{(y+1)}, \mathbf{T}^{(y)}, \Theta_V^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$
- For $k = 1, ..., K$, sample the latent features of parameter (in parallel):
  $T_k^{(y+1)} \sim p(T_k|\hat{\mathbf{U}}^{(y+1)}, \mathbf{V}^{(y+1)}, \Theta_T^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$

$$p(\Delta\hat{R}_{ij}^k|\Delta\mathbf{R}) = \int p(\Delta\hat{R}_{ij}^k|U_i, V_j, T_k, \alpha)\cdot p(\mathbf{U}, \mathbf{V}, \mathbf{T}, \alpha, \Theta_U, \Theta_V, \Theta_T|\Delta\mathbf{R})\cdot d(\mathbf{U}, \mathbf{V}, \mathbf{T}, \alpha, \Theta_U, \Theta_V, \Theta_T). \quad (7)$$

We resort to numerical approximation techniques to solve the above integral. To sample from the posterior, we use Markov Chain Monte Carlo (MCMC) sampling. We use the Gibbs sampling as our MCMC algorithm. We can approximate the integral by,

$$p(\Delta\hat{R}_{ij}^k|\Delta\mathbf{R}) \approx \sum_{y=1}^{Y} p\left(\Delta\hat{R}_{ij}^k|U_i^{(y)}, V_j^{(y)}, T_k^{(y)}, \alpha^{(y)}\right) \quad (8)$$

Here we draw Y samples and the value of Y is set by observing the validation error. The sampling from $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ is simple since we use conjugate priors for the hyper-parameters. Also, a random variable can be sampled in parallel while fixing others which reduces the computational complexity. Algorithm 1 shows how to iteratively sample $\mathbf{U}, \mathbf{V}, \mathbf{T}$ and obtain $\mathbf{P}$ and Q. Note that it is required in the algorithm to reconstruct $\mathbf{U}^{(y+1)}$ at every iteration since there will always be a small reconstruction error $||\hat{\mathbf{U}}^{(y+1)} - \mathbf{U}^{(y+1)}||$. The error occurs because we force Q to be a row vector, which makes the exact recovery of $\mathbf{U}^{(y+1)}$ difficult. The reconstructed error causes adjustment of $\mathbf{V}$ and $\mathbf{T}$. Once we obtain the four latent factors, our task is to predict the parameter values for M enhanced versions having K parameters each. Suppose F_t is the feature vector of a new image, then the parameter values $\Delta\hat{R}_{tj}^k$ can be simply obtained by computing, $\Delta\hat{R}_{tj}^k =< F_t^T\mathbf{P} + Q, V_j, T_k > \ \forall\ j \in \{1, \ldots, M\}$ and $k \in \{1, \ldots, K\}$. If the parameter value predictions lie beyond a certain range then a thresholding scheme can be used based on the prior knowledge. For example, to constrain the predictions between $[0, 1]$, a logistic function may be used.

[6] See supplementary material on author's website for detailed derivations.

4. Experiments

We conduct two experiments to show the effectiveness of our approach. We did the first one on a synthetic data and compared it with: 1. BPMF 2. our own discrete version of BPTF, called D-BPTF. 3. multivariate linear regression (MLR) 4. twin Gaussian processes (TGP) [3] 5. Weighted kNN regression (WKNN). For D-BPTF, we make minor modifications in the original BPTF approach [24] by removing the temporal constraints on their temporal variable, since there are no temporal constraints in our case. The inference for their temporal variable is then done in the exactly same manner as the other non-temporal variables. This gave us a marginal boost in the performance. For MLR, We use a standard multivariate regression by maximum likelihood estimation method. Specifically, we use MATLAB's `mvregress` command. TGP is a generic structured prediction method. It accounts correlation between both input and output resulting in improved performance as compared to MLR or WKNN. The WKNN approach predicts the test sample as a weighted combination of the k-nearest inputs. The first two algorithms do not allow features inclusion. For MLR, TGP and WKNN, we concatenate A_i and F_i, and use it to predict $A_i'^j$. Even for our approach, we concatenate A_i and sample feature to form F_i. The intuition behind this concatenation is that the enhancement parameters should be a function of input parameters as well along with the features. We did observe performance boost after concatenating the features and parameters.

The second experiment demonstrates the usefulness of this approach in a real-world setting where we have to predict paramters of the enhanced versions of an image (then generate those versions by applying predicted parameters to the input low-quality image) without using any information about the versions. We compare our approach with the

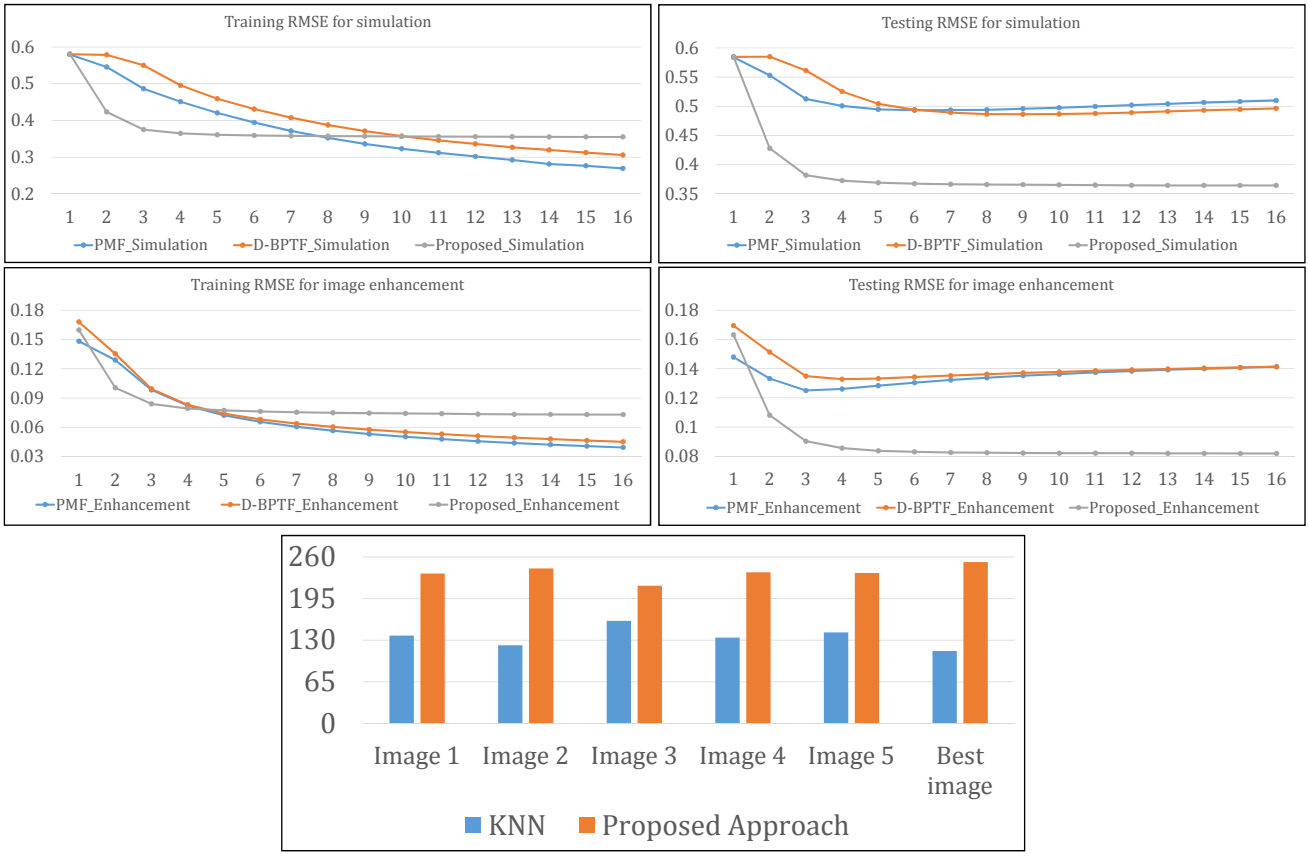

Figure 1. Top plots: train and test RMSEs for both the experiments. Bottom plot: First 5 sets of bars show votes for version 1 to 5 of kNN vs. the best image of our approach. The last set of bars shows votes for the best image of both approaches. Please zoom in for better viewing. See in color.

competing 5 algorithms in addition to kNN-search as it is also used in [26, 5]. We also analyzed the effect of Q in our solution by: removing Q i.e. $\mathbf{U} = \mathbf{F}^T\mathbf{P}$.

4.1. Data set description and experiment protocol

The synthetic data is carefully constructed by keeping the following task in mind. We are given a training set consisting of: 1. $\mathbf{F} \in \mathbb{R}^{D\times N}$; 2. $\mathbf{A} \in \mathbb{R}^{K\times N}$; and 3. *only* parameters of M versions for each input sample - $\mathbf{A}' \in \mathbb{R}^{K\times N\times M}$. Our aim is to predict parameters for a set of M versions given a new F_i and A_i. In real-world problems, $\mathbf{A}$ and $\mathbf{F}$ are interdependent. The parameters of M versions are dependent on both $\mathbf{A}, \mathbf{F}$. Hence we construct the synthetic data as follows.

Firstly, we generate a set of 3-D input parameters - $\mathbf{A}$ - drawn from a uniform distribution $[0, 1]$. Then we generate a 50-D feature set $\mathbf{F}$, where each element of F_i is related to all $A_{k,i}\ \forall\ i = \{1, \ldots, 10^3\}, k = \{1, 2, 3\}$ by a nonlinear function. For example, $F_{j,i} = r_1^{A_{1,i}} + \frac{1}{1+e^{-r_2 A_{2,i}}} + A_{3,i}^{r_3}, \forall\ j \in \{1, \ldots, 50\}$ and r_1, r_2, r_3 are random numbers. The parameters of enhanced versions, $A'_{k,i,m}$, are also non-linearly related to $A_{k,i}\ \forall\ k, \forall\ m \in \{1, \ldots, 4\}$ and F_i. For example, $A'_{k,i,m} = \eta\left(r_1^{A_{1,i}} + \frac{1}{1+e^{-r_2 A_{2,i}}} + A_{3,i}^{r_3}\right) + (1 - \eta) \cdot ||F_i||_2$. The contribution of F_i is decided by η. We perform 3-fold cross-validation. We predict the values of $\mathbf{A}'$ in the test set (disjoint from training) using corresponding $\mathbf{A}$ and $\mathbf{F}$. RMSE is computed between the predicted and actual $\mathbf{A}'$.

The MIT-Adobe FiveK data-set contains 5000 high-quality photographs taken with SLR cameras. Each photo is then enhanced by five experts to produce 5 enhanced versions. We extract average saturation, brightness and contrast for every image, which are parameters $\in \mathbf{A}$. We also extract 1274-D color histogram with 26 bins for hue, 7 bins each for saturation and value. We also calculate localized features of 144-D each for contrast, brightness and saturation. Finally, we append average saturation, brightness and contrast of the input low-quality image, which are its parameters. Thus we get a 1709-D $(= 1274 + 3 \times 144 + 3)$ representation for every image $\in \mathbf{F}$. We train using 4000 images and use 500 images each for validation and testing. We predict parameters for 5 versions in a 3×5 matrix for

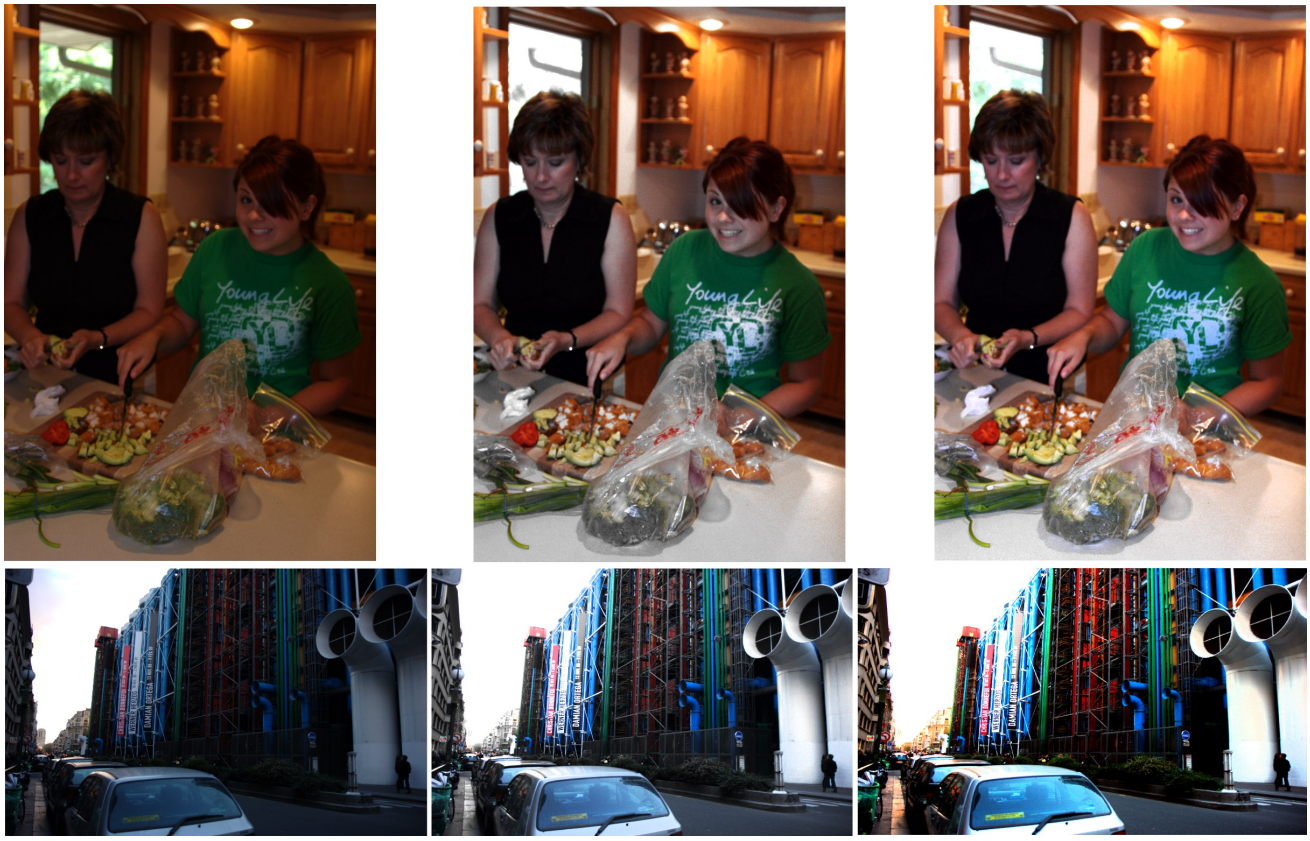

Figure 2. Left: Original image, Middle: enhanced image by kNN and Right: proposed approach [7]. See in color.

each image in the testing set. An entry $A'_{i,j}$ denotes the value for i^{th} parameter of j^{th} enhanced version. To enable comparison with the expert-enhanced images of the data-set, we also compute parameters for 5 enhanced versions for each image, which we treat as ground-truth. We evaluate this experiment in two ways. Firstly, we calculate RMSE between the parameters of 5 expert-enhanced photos and the parameters of the predicted versions using five aforementioned algorithms. Secondly, we conduct a subjective test under standard test settings (constant lighting, position, distance from the screen). In this case, we compare our approach with the popular kNN-search-based approach. It first finds the nearest original image in the training set to the testing image - $\mathbf{im}$ - and then applies the same parameter transformation to $\mathbf{im}$ to generate 5 version. In our approach, we predict the parameters for enhanced versions using the proposed formulation. We threshold the parameter values as:

$$\begin{aligned} A'_{k,i,m} &= \min(A'_{k,i,m}, A_{k,i} + \lambda_k A_{k,i}), \\ A'_{k,i,m} &= \max(A'_{k,i,m}, A_{k,i} - \zeta_k A_{k,i}), \end{aligned} \tag{9}$$

where λ and ζ are multipliers for the k^{th} parameter. In our case, the multipliers for saturation, brightness and contrast are: $\lambda = \{0.4, 0.4, 0.05\}, \zeta = \{0.3, 0.3, 0.01\}$. As mentioned before, the clipping scheme in our formulation should be set using prior knowledge. Here, we know that the enhanced images usually have a larger increase (as compared to decrease) associated with their parameters. Also, changing contrast by a very small amount affects the image greatly.

The predicted parameters are applied to the input image to obtain its enhanced versions. The procedure is the same for both the approaches and is as follows. First we change contrast till the difference between the updated and the predicted contrast is marginal. We update contrast first since changing it updates both brightness and saturation. We then update brightness and saturation till they come significantly closer to their corresponding predicted values. This gives us 5 versions for both approaches. To allow comparisons within a reasonable amount of time, we use a pre-trained ranking weight vector w (from [5]) to select the best image of our approach (*im-proposed*) and kNN-approach (*im-kNN*). For the subjective test, people are told to compare *im-proposed* with the 5 enhanced versions of kNN-approach as well as with *im-kNN*. Thus for every input image, people

[7] See supplementary on author's website for additional full-resolution results.

perform 6 comparisons. The image order was randomized. We conducted the test with 11 people and 35 input images. Thus every person compared 210 pairs of images. They were told to choose a visually-appealing image. The third option of simultaneously preferring both images was also provided. This option has no effect on cumulative votes.

4.2. Results

The parameters for the synthetic data were more accurately predicted by our approach than BPMF, D-BPTF, MLR, TGP and WKNN. It is worth noting that though the training error continues to decrease for our approach, BPMF and D-BPTF, the testing error starts increasing after only 5 and 8 iterations for BPMF and D-BPTF, respectively. However, testing error in our approach decreases rapidly for 4 iterations and then it decreases very slowly for the next 12, as shown in Fig. 1. The RMSE on test set for BPMF, D-BPTF, MLR, TGP, WKNN and the proposed approach is $0.4933, 0.4865, 0.6293, 0.4947, 0.8014$ and 0.3644. The numbers show that our approach is able to effectively use the additional information provided by features and the interaction between $\mathbf{A}, \mathbf{F}$ and all versions to provide better prediction. On the other hand, BPMF and D-BPTF start over-fitting quickly due to lack of sample feature information while MLR and WKNN fail to model the complex interaction between variables. TGP performs better because of its ability to capture correlations between input and output. However, TGP still treats each version independently and thus its performance still falls short of our approach.

In the second experiment, the RMSE for BPMF, D-BPTF, MLR, TGP, WKNN and our approach is $0.1251, 0.1328, 1.2420, 0.1268, 0.1518$ and 0.0820 respectively. The testing error starts increasing after only 3 and 5 iterations for BPMF and D-BPTF, respectively. It is important to note that we do *not* use the clipping scheme mentioned in Equation 9 in order to do a fair comparison of RMSEs between all the five approaches and the proposed appraoch. For the subjective evaluation, Fig. 1 shows cumulative votes obtained for ours and the kNN-based approach for comparison between 5 images chosen by kNN and the best image chosen by our approach. Fig. 1 also shows votes obtained for the best images chosen by both approaches. Fig. 2 shows two input images enhanced by both the approaches. The top row of Fig. 2 shows that kNN reduces the saturation while increasing the brightness. Our approach balances both of them to obtain a more appealing image. In the bottom row, however, both approaches fail to produce aesthetic images as images become too bright. It is probably due to the portion of the sky in the input image. For both the images, most people prefer images enhanced by our approach. Computationally, our approach is superior than kNN. Complexity of our approach is independent of data-set size at testing time whereas kNN searches the entire data-set for the closet image and then applies its parameters.

Table 1. Effect of varying β and δ

Parameter setting	RMSE (lower the better)
$\beta = 0.001, \gamma = 6$	0.3162
$\beta = 0.01, \gamma = 6$	0.0962
$\beta = 0.02, \gamma = 0.1$	0.0907
$\beta = 0.2, \gamma = 0.05$	0.0930
$\beta = 0.8, \gamma = 0.05$	0.0872
$\beta = 0.1, \gamma = 0.3$	**0.0820**
$\beta = 0.1, \gamma = 0.8$	0.0821
$\beta = 0.1, \gamma = 2$	**0.0820**

We reconstructed $\mathbf{U} = \mathbf{F}^T\mathbf{P}$ and observed performance drop as it overfits. We get RMSE of 0.9305 and 0.3762 on enhancement and simulation data, respectively. We believe the real-world enhancement data has correlations naturally embedded in it unlike in synthetic data. Thus the performance drop is drastic in case of enhancement since the problem of recovering $\mathbf{P}$ only from $\mathbf{U}$ and $\mathbf{F}$ is ill-posed.

We also analyzed the effect of varying β and δ. Since our approach uses Bayesian probabilistic inference, small variations in β and δ do not significantly affect the performance. Table 1 lists the various parameter settings and its effect on the performance of the second experiment (i.e. image enhancement):

5. Conclusion

In this paper, we introduced a novel problem of predicting parameters of enhanced versions for a low-quality image by using its parameters and features. We developed an MF-inspired approach to solve this problem. We showed that by modeling the interactions across low-quality images, its parameters and its versions, we can outperform five state-of-art models in structured prediction and MF. We proposed inclusion of feature information into our formulation through a convex $\ell_{2,1}$-norm minimization, which works in an iterative fashion and is efficient. Thus our approach utilizes information which helps characterize input image. This leads to better generalization and prediction performance. Since other approaches do not model interdependence between image features and parameters of their corresponding enhanced versions, they start over-fitting quickly and produce an inferior prediction performance on the test set. Experiments on synthetic and real data demonstrated superiority of our approach over other state-of-art methods.

Acknowledgement: The work was supported in part by an ARO grant (#W911NF1410371) and an ONR grant (#N00014-15-1-2344). Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ARO or ONR.

References

[1] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304. ACM, 2011.

[2] F. Berthouzoz, W. Li, M. Dontcheva, and M. Agrawala. A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations. *ACM Trans. Graph.*, 30(5):120, 2011.

[3] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, 2010.

[4] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 97–104. IEEE, 2011.

[5] P. S. Chandakkar, Q. Tian, and B. Li. Relative learning from web images for content-adaptive enhancement. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015.

[6] C.-Y. Chen and K. Grauman. Inferring unseen views of people. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2011–2018. IEEE, 2014.

[7] L. Chen, Q. Zhang, and B. Li. Predicting multiple attributes via relative multi-task learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1027–1034. IEEE, 2014.

[8] S. B. Kang, A. Kapoor, and D. Lischinski. Personalization of image enhancement. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1799–1806. IEEE, 2010.

[9] L. Kaufman, D. Lischinski, and M. Werman. Content-aware automatic photo enhancement. In *Computer Graphics Forum*, volume 31, pages 2528–2540. Wiley Online Library, 2012.

[10] N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 601–608. ACM, 2009.

[11] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *Computer Vision–ACCV 2012*, pages 316–327. Springer, 2013.

[12] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[13] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[14] B. Marlin, R. S. Zemel, S. Roweis, and M. Slaney. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267*, 2012.

[15] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[16] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1813–1821. Curran Associates, Inc., 2010.

[17] D. Parikh and K. Grauman. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE, 2011.

[18] D. Parikh, A. Kovashka, A. Parkash, and K. Grauman. Relative attributes for enhanced human-machine communication. In *AAAI*, 2012.

[19] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

[20] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[21] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3, 2014.

[22] Q. Song, J. Cheng, and H. Lu. Incremental matrix factorization via feature space re-learning for recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 277–280. ACM, 2015.

[23] S. Wang, J. Tang, Y. Wang, and H. Liu. Exploring implicit hierarchical structures for recommender systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, 2015.

[24] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, volume 10, pages 211–222. SIAM, 2010.

[25] J. Yan, S. Lin, S. B. Kang, and X. Tang. A learning-to-rank approach for image color enhancement. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2987–2994. IEEE, 2014.

[26] J. Yan, S. Lin, S. B. Kang, and X. Tang. A learning-to-rank approach for image color enhancement. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2987–2994. IEEE, 2014.