

Does this Shirt go with this Tie?

Monique Ryan, Dave Willis, Pranav Khaitan

Abstract

The aim of this project is a proof of concept for an internet apparel shopping assistant. We prove the concept by applying supervised learning algorithms to automatically classify shirt and tie combinations as good matches or bad matches based on their color and pattern features. Since it is more important for the system to have a low false positive rate than a low classification error the algorithms are tuned for high precision at the expense of classification error. This resulted in a precision of 91.37% with 27.25% classification error. Before precision tuning our best classification error is 23%. Our best results were achieved using features of top 2 essential colors and a custom pattern complexity metric and SVM with a quadratic kernel implemented as an ensemble. Investigations into other features and supervised learning algorithms are summarized.

1 Introduction

When shopping for clothes on internet shopping sites, the customer is commonly presented with tens to hundreds of images of items available for purchase and no easy way to filter the images based on the suitability of the items in relation to items they already own and other items they intend to purchase.

The aim of this project is to apply machine learning algorithms to data from images of pairings of clothing items to attempt to classify them as matches or mismatches. We choose classification of shirt and tie combinations as matches or mismatches as a starting point because the range of features relevant to matching shirts with ties is limited to color, texture and patterns in the shirts and ties.

Automation of the process of classifying combinations of clothing items as matches or mismatches would allow internet shopping websites to filter the items displayed based on items the customer has purchased previously or already selected for purchase.

2 Training Data

The training data consists of pairs of images of shirts and ties. The class of positive training examples consists of a collection of 418 expertly selected shirt and tie combinations culled from internet shopping sites. Refer to Figure 8 for examples.

The positive training data was generated by cropping a 60x60 image from each shirt and a 50x60 image from each tie.

The negative class was generated by combining cropped images of shirts and ties. Initially the class of negative training examples was generated by randomly selecting images of shirts from the collection and pairing them with randomly selected images of ties. However, a visual inspection of the resulting shirt/tie combinations revealed that over 50% of this set of shirt/tie combinations were considered to be good matches by all three team members. This is probably due to the high proportion of ties in the collection that by design go with a high number of the shirts in the collection. For example there were a high number of pale blue shirts and dark blue ties.

A second set of negative training examples was generated by pairing randomly selected shirts and ties from the raw images and displaying them to users who classified each of the pairings as matches or mismatches. A shirt and tie pair was added to the negative class if two users independently classified the combination as a mismatch. This approach aimed to minimize the effect of personal preference on the classification of combinations as matches or mismatches. 410 negative training examples were generated this way and used in all subsequent evaluation of the system.

The negative training set used in this project is somewhat limited since it only includes obvious mismatches. A number of shirt/tie combinations that could be described as mild mismatches were excluded from the dataset. However, the dataset used in this project is considered adequate for a proof of concept study.

If it were available, we believe that the ideal data set for training a system like this would be a set of graded preference data averaged over a range of people. This could be gathered by asking a number of people of rate how much they liked particular shirt/tie combinations on a scale eg 1-10.

3. Feature Selection

The features considered were primarily color and pattern of the shirts and ties. A combination of color and pattern features is used to train the classifier in the final solution. All feature generation was performed using Matlab.

3.1 Color Features

The color features used in the final solution consisted of the top two RGB values of the shirt and tie.

A custom function was developed to extract the top N colors from an image by percentage of pixels. When passed the parameter N (number of colors to consider) the function returns a feature matrix whose rows consist of the RGB values of the top N shirt colors, followed by the top N tie colors. When passed a flag, the function would also include the percentage of each color in the feature matrix returned. This function is implemented as follows:

1. Use K-Means clustering to group the colors into N clusters.
2. Combine very similar colors until only the essential colors remain.
3. An empty color is filled in with the top color at 0 percent.

The output of this algorithm for two cropped images and N = 5 is shown in Figure 2. Each cropped image has two versions, the original on the left and the output of this algorithm on the right. The numbers are the RGB colors and percentage.



Figure 2 – Output of Color Extraction function with N = 5, percentages flag true.

A number of feature matrices were generated using this function with varying values of N and with and without color percentages. The best results were achieved with N=2 without color percentages. The resulting feature matrix was therefore an m by 12 matrix where m is the number of training samples, consisting of the three RGB values for each of the top two colors in the shirt and the tie.

A number of options were considered as possible color features. Each of the options was evaluated by training an SVM classifier on the resulting feature matrix and evaluating the classifier using 30% holdout cross validation. A comparison of the different feature options is in the Results section.

Average Shirt Color and Average Tie Color – Initially average RGB colors were calculated over all the pixels in the shirt and all the pixels in the tie resulting in a feature matrix with 6 columns. This method of generating color features would be expected to perform poorly on shirts and ties with multiple colors.

RGB vs. HSV Color Model – both average RGB color and average HSV color features were generated for comparison. Because HSV (Hue, Saturation and Value - brightness) is used in color theory literature [1] to define color harmony and contrast we posited representing color using the HSV numbers would produce better results. However comparative results for average RGB were superior.

Color Histogram – A color histogram specifies N color buckets and records the number of pixels in an image that fall into each the N buckets. Initial experiments using the color histogram as input features for the classifier were not promising. Intuitively this is to be expected as it involves grouping a range of colors together, which can lose subtle distinctions. It is often the case that two similar colors, eg two similar shades of blue, go together well while a different set of similar blues will not.

Color distance measurement – Color Theory literature [3], [4], [5] defines a number of traditional color palettes that work well and appear harmonious. These color schemes can be defined in terms of the relative position of the colors on the color wheel. Examples include monochromatic color scheme, analogous color scheme and complementary colors. A measurement of the distance between two sets of colors was attempted where a small difference indicates the two sets of colors are harmonious and a large distance indicates they are not. The results were inferior to Average RGB color. With this experience we agree with Professor Bokor, Department of Psychology, University of Virginia who wrote:

“A uniform metric for the perception of color has remained something of a mirage, at once seeming to be easily attainable and yet always just out of reach”

3.2 Pattern Features

The pattern features used in the final solution are based on a metric related to the discrete cosine transforms (DCT) of the shirt and tie images. This metric is a measure of the visual boldness or complexity of the image. We observed that in the class of matching shirts and ties, there is an inverse correlation between the visual boldness of the shirt and the visual boldness of the tie. That is, a bold/complex tie should be paired with a plain shirt and vice versa.

The boldness metric was calculated as follows:

1. Calculate the DCT of each row and column of the image and store the DCT coefficients.
2. Calculate the weighted average of the DCT coefficients of the image (over all rows, columns and RGB components) using an exponentially decreasing weight for higher frequency components and a weight of 0 for the 0Hz component.

Examples of patterns and the values of the complexity metric for the images follow in Figure 3:

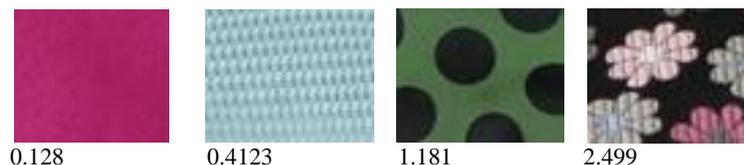


Figure 3 – Examples of patterns and their corresponding Complexity Values

Two broad approaches were considered for specifying the pattern features observed in shirts and ties.

Detecting edges within patterns – this approach involves converting the image to grayscale and detecting edges in the pattern to determine characteristics of the pattern. For example, does the pattern contain stripes or spots and how wide are the stripes or spots. This approach could also be used to define and average boldness/complexity metric by summing the number of pixels that correspond to edges in the image. The limitation of this approach is that it fails to account for the interaction between the magnitude of changes in color and the visual complexity or boldness of the image.

Discrete Cosine Transform Components – this approach involves extracting features from the DCT coefficients of the image. When considering DCT components to define a complexity/boldness metric the two primary variables that were considered were which frequency components are most significant and how to combine the results over all the rows and columns of the image and the RGB components of the image.

4 Supervised Learning Algorithms

The final implementation used an SVM classifier with a quadratic kernel implemented using Matlab’s Biometrics Toolbox. The SVM classifier was run separately on the color and pattern features and the resulting classifications were combined using an ensemble method that classified a shirt and tie combination as a match if both the color and pattern features were classified as matches and a mismatch otherwise. This approach was taken to increase the precision of the classifier at the expense of a higher classification error since intuitively, a shirt and tie combination will only match if both the colors and the patterns are a match. The features described above produced data that is non-Gaussian and inter-dependent leading us to chose Logistic Regression and SVM as our supervised learning algorithms. The ensemble method was chosen to reduce the number of false positives. These classifiers were evaluated using a range of different input features and using 30% holdout cross validation. Refer to the Results section for a comparison of the results of the SVM classifier and the Logistic Regression classifier.

5 Results

The system was evaluated using 30% holdout cross validation with a dataset containing 418 positive training examples and 410 negative training examples. The table below shows the comparative error and precision results obtained using the features and supervised learning algorithms described above. For these experiments we held the decision boundary constant at 50%.

Features	Error	Logistic Regression	SVM – Linear Kernel	SVM – Quadratic Kernel
Average RGB Color	Classification Error	32.11	32.27	26.85
	Precision	67.00	65.10	72.59
Average HSV Color	Classification Error	33.71	35.06	33.63
	Precision	66.43	66.10	54.99
Patterns Only	Classification Error	27.33	27.65	24.22
	Precision	71.03	69.39	79.72
Top Three Colors Only	Classification Error	31.16	30.20	32.75
	Precision	69.47	67.92	66.20
Top Two Colors Only	Classification Error	32.43	28.29	25.98
	Precision	65.12	71.37	73.11
Top Two Colors + Patterns	Classification Error	25.18	25.34	22.63
	Precision	74.14	74.67	77.75
Top Two Colors + Pattern - ensemble	Classification Error	26.77	26.22	25.18
	Precision	81.35	79.93	87.71

The SVM classifier using the quadratic kernel and ensemble method for color and pattern features with top two colors and the pattern complexity metric used as input features gave the highest precision (87.71%) .

Since it is considered more important for the system to have a low false positive rate than a low overall classification error, based on the ROC curve, the SVM soft margin classifiers (C1, C2) were tuned to increase the precision at the expense of increasing the overall classification error.

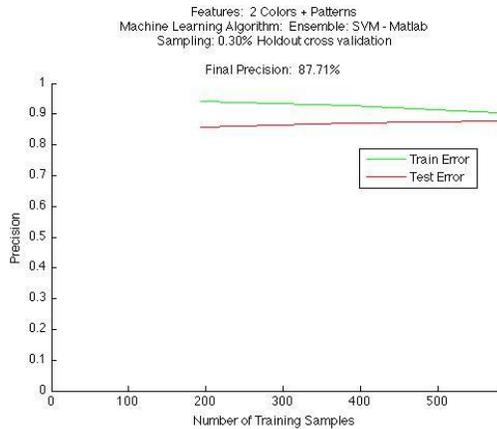


Figure 4 –Precision Learning Curve – C1 = 1, C2 = 1

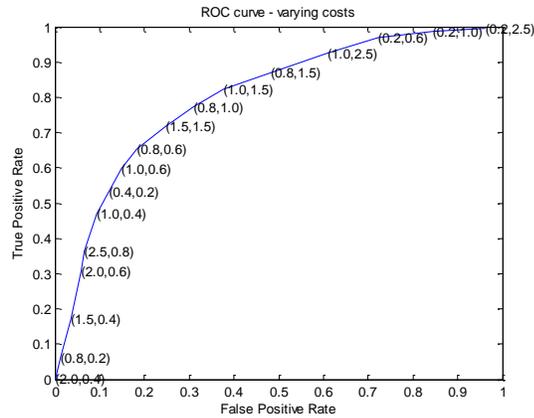


Figure 5 – ROC Curve

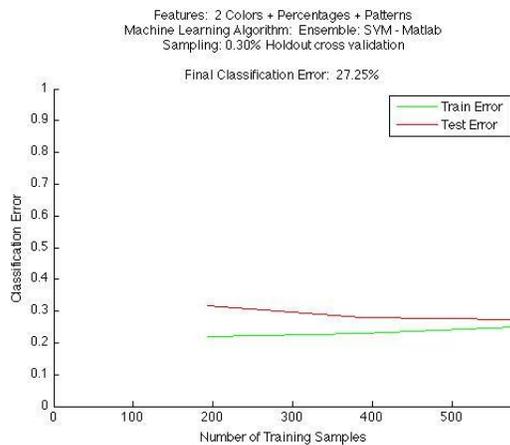


Figure 6 – Classification Error Learning Curve
C1 = 1, C2 = 0.6 (tuned for high precision)

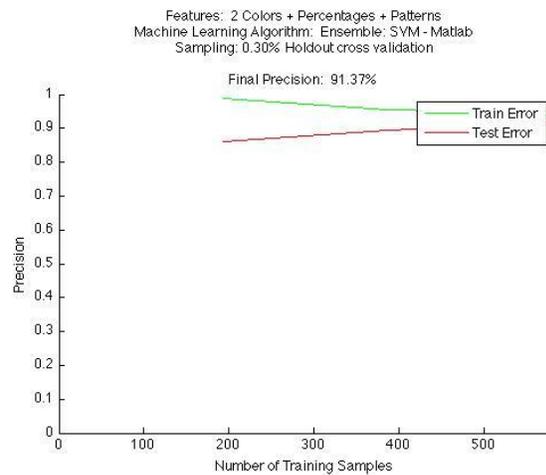


Figure 7 – Precision Learning Curve
C1 = 1, C2 = 0.6 (tuned for high precision)

This resulted in a final solution with 91.37% precision and 27.25% overall classification error with soft margin classifiers at 1.0 and 0.6.

6 Conclusions and Future Work

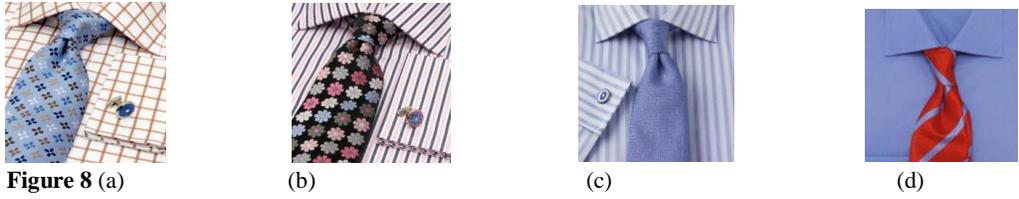
We believe that this implementation has proved that the concept of designing a classifier for matching or non-matching shirt and tie combinations with high precision is feasible using Machine Learning algorithms.

The high precision achieved by the final solution comes at the expense of a relatively high overall classification error, around 27.3% however, it was possible to achieve a classification error as low as 23.5% using a different set of features and combination of algorithms.

It is intuitively infeasible for a classifier of this kind using only pass/fail data to aim for an overall classification error close to 100% due to the variability between individuals in what constitutes a good match and what constitutes a mismatch and the difficulty of mathematically quantifying features relevant to the overall harmony of the ensemble.

The pattern boldness metric, that was developed to quantify shirt and tie pattern complexity performed relatively well. An overall classification error of 25% was recorded when the SVM classifier was trained with pattern data alone. However, certain shirt/tie combinations would always be misclassified using this metric. The four images below all appear in the positive training set. The two images on the left (a, b) are always misclassified as mismatches using our complexity metric and the two images on the right (c, d) are classified as matches. Intuitively, the reason for the misclassification of (a, b) is that both the shirt and tie are assigned a high complexity value which is unusual. It is possible that a more sophisticated feature set to describe patterns would lead to improved classification. A possible direction of further inquiry would be to investigate which DCT components were most relevant in pattern classification using Principle Components Analysis.

Using graded preference data may also improve results in this area since (a, b) may not be universally accepted as matches.



The color features also performed quite well considering the limitations on the available training data. An overall classification error of 27% was recorded when the SVM classifier was trained with color data alone (two top colors). Examination of the false positives and false negatives returned by this classifier revealed two primary trends. Shirt/tie combinations where the shirt or tie is purple or green are likely to be misclassified as mismatches. This suggests that purple and green may be over-represented in the negative training class and under-represented in the positive training class, leading the classifier to classify all purple or green shirts and ties as mismatches.

Combinations with more than two significant colors are likely to be misclassified. For example a blue and white shirt containing fine orange stripes will clash with a maroon colored tie however the classifier will often report a false positive on this combination (Figure 9 a, b). This is to be expected as the classifier only considers the top two colors.

Colors where the mismatch is between two subtly different colors are also often misclassified (Figure 9 c, d).



It is possible that using a larger training set with a more even distribution of colors would improve results in this area. The training graphs for the classifier when trained on the top 3 colors and the top two colors are shown below. The training graph for the top 3 colors suggests that more data could improve the final precision since the training error is low.

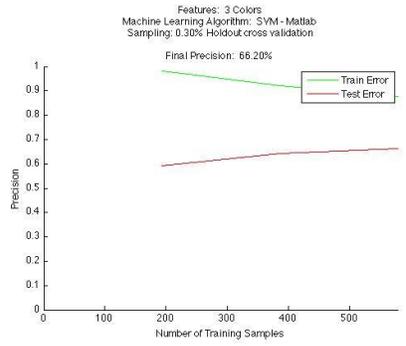


Figure 10 – Learning Curve 3 Colors

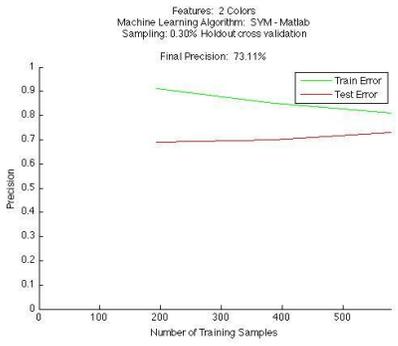


Figure 11 – Learning Curve 2 Colors

One of the limitations of this system is variability caused by personal preference in the negative training data. Most work in this area uses graded preference data [2] which would probably be more appropriate than pass or fail training data. Another possible direction for further work would be to investigate customizing the system to account for personal preferences.

References

[1] Stephen Palmer, “Vision Science, Photons to Phenomenology”, Bradford Books, MIT Press, Cambridge, MA, 1999
 [2] Stephen Palmer, Karen Schloss, “Color Aesthetics – The Berkeley Color Project”, Visual Perception and Aesthetics Lab, 2009, University of California Berkeley, <<http://socrates.berkeley.edu/~plab/color.html>>
 [3] Malane Newman, “Color Theory & the Color Wheel”, MalaneNewman, 2007, <http://www.malanenewman.com/color_theory_color_wheel.html>
 [4] Igor Asselbergs, Misc Archives, LivelyGrey.com, 2008 <<http://www.livelygrey.com/misc>>
 [5] Pabini Gabriel-Petit “Color Theory for Digital Displays: A Quick Reference, Part I” UXmatters, Jan 2006, <<http://www.uxmatters.com/mt/archives/2006/01/color-theory-for-digital-displays-a-quick-reference-part-i.php>>