DOI: 10.58895/ksp/1000044906-8 erschienen in:

OCM 2015 - 2nd International Conference on Optical Characterization of Materials, March 18th – 19th, 2015, Karlsruhe, Germany : Conference Proceedings

DOI: 10.58895/ksp/1000044906 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000044906/

Towards many-class classification of materials based on their spectral fingerprints

Matthias Richter[1,2] and Jürgen Beyerer[2,1]

[1] Karlsruhe Institute of Technology, Institute for Anthropomatics and Robotics
Adenauerring 4, D-76131 Karlsruhe, Germany
[2] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Fraunhoferstr. 1, D-76131 Karlsruhe

Abstract Hyperspectral sensors are becoming cheaper and more available to the public. It is reasonable to assume that in the near future they will become more and more ubiquitous. This gives rise to many interesting applications, for example identification of pharmaceutical products and classification of food stuffs. Such applications require a precise models of the underlying classes, but hand-crafting these models is not feasible. In this paper, we propose to instead learn the model from the data using machine learning techniques. We investigate the use of two popular methods: support vector machines and random forest classifiers. In contrast to similar approaches, we restrict ourselves to linear support vector machines. Furthermore, we train the classifiers by solving the primal, instead of dual optimization problem. Our experiments on a large dataset show that the support vector machine approach is superior to random forest in classification accuracy as well as training time.

1 Introduction

Recent developments in consumer electronics have made compact hand-held spectrometers affordable for end-consumers. Products operating in the visible to the near infrared electromagnetic spectrum will most likely be available within this decade and might even be integrated into our smart phones in the not too distant future. This development gives rise to many exiting opportunities from research over retail to

health care. An obvious application is the identification and characterization of materials such as ore, food, and pharmaceutical products. For this the spectral signatures of different materials have to be condensed in some sort of model. Although the properties of many substances are well understood, building this model by collecting and combining expert knowledge is infeasible due to the sheer number of different materials. Moreover, this approach does not scale when more categories are to be included in the model. Instead, it is desirable to automatically infer the model from a given annotated sample of signatures. This way, human interaction is reduced to simply providing measurements and labels (e.g. material classes).

In this paper, we propose to use machine learning techniques to *learn* the model from a large, labeled sample of spectral signatures. Here we only consider classification: Given the spectrum of an unknown object, find the material of that object. We base the methods on our existing spectral material database, which contains hyperspectral images covering the UV, VIS and SWIR spectral bands (see [1] for more details). This dataset poses a number key challenges to the classification system:

- The number of classes is very large and novel classes can be added to the dataset at any time.
- Since the database consists of hyperspectral *images*, the number of spectral signatures per material is very large. In addition, the dataset is imbalanced, meaning that certain materials are represented with more signatures than others.
- The signatures show a high intra-class variance, especially (but not only) in recordings of natural materials.
- The system has to be able to cope with the high dimensionality of the data and the resulting *curse of dimensionality*[3].

Using spectral signatures to classify different materials is not a new idea. Mostly due to historical reasons, a lot of this research is conducted in the area of remote sensing, where one goal is to discriminate different types of land coverage (e.g. urban areas, types of forestry, and healthy

[3] The number of training samples required to estimate reliable models grows exponentially in the dimensionality of the feature space (also known as *Hughes phenomenon*).

vs. diseased crops). Here, specialized image processing and machine learning techniques have been proven useful to both classify coverage as well as extract discriminative features from raw data [2]. As documented in several extensive surveys (e.g. [3, 4]), a large portion of this work is concerned with support vector machines (SVM). This is due to their ability to derive good decision boundaries even with very few samples and large feature spaces, which is typically the case in remote sensing [5]. Recently the attention has turned toward the Random forest (RF) framework (e.g. [6, 7]). One benefit is that RFs can be used to determine spectral bands that are most suitable for classification. This information can later be employed to significantly reduce the amount of the stored data.

In light of this vast body of research and many successful applications, it is a bit surprising that this development is only slowly followed on a finer scale, e.g. in automated visual inspection. This is probably best explained by the relatively high costs of hyperspectral imaging sensors as well as slow data processing. Nonetheless, Lorente et al. reported recent developments in both sensor technology and techniques to analyze the data [8]. They find that the most prevalent methods revolve around classical statistical tools like principal component analysis and partial least squares for dimensionality reduction and linear discriminant analysis and artificial neural networks for classification.

However, all methods presented in [8] as well as the approaches found in remote sensing only deal with with a relatively moderate number of classes. In this paper, we instead ask the question: What machine learning methods are best suited to classify a large number of different classes based on their spectral signatures? For that we compare two promising candidates: support vector machines and random forests.

2 Methods

Given a set $\mathcal{D} = \{(\mathbf{s}_n, y_n)\}|n = 1, \ldots, N\}$ of spectral signatures $\mathbf{s}_n \in \mathbb{R}^D$ and class labels $y_n \in \mathbb{N}$, we search a hypothesis $H(\mathbf{s}) = y$ that maps unknown signatures to the most probable class. We wish the hypothesis to be consistent with $\mathcal{D}$, that is the number of misclassifications $H(\mathbf{s}_i) \neq y_i$ should be small. At the same time it should generalize well and not make too many mistakes on unseen data.

We take the spectral signature $\mathbf{s} = (s_1, \ldots, s_D)^\top$ to be the output of a hyperspectral sensor, meaning that each s_d corresponds to the measured intensity of a certain spectral band. However, we don't pay too much attention to the physical world: Whether $\mathbf{s}$ represents reflectance or transmission or how $\mathbf{s}$ has been processed to ensure comparability (black-/white-balance, normalization, etc.) is not of our concern, as long as these properties are consistent over all measurements.

2.1 Linear Support Vector Machines

As mentioned in Section 1, support vector machines have been very successfully applied to classification of hyperspectral data. A linear SVM is a binary classifier ($y = \pm 1$), that classifies $\mathbf{s}$ according to

$$H(\mathbf{s}) = \operatorname{sgn}(\mathbf{w}^\top \mathbf{s} + b). \tag{8.1}$$

From a geometric perspective $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ define a hyperplane in the D-dimensional feature space. Classification into either the positive or negative class amounts to determining whether $\mathbf{s}$ falls "left" or "right" of that hyperplane. In training, $\mathbf{w}$ and b are chosen to best separate the two classes while simultaneously maximizing the *margin*, i.e. the distance $\frac{1}{\|\mathbf{w}\|}$ of the hyperplane to nearest training samples (the *support vectors*). This amounts to solving the unconstrained optimization problem [9]

$$\text{minimize} \quad \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} L(y_n, \mathbf{w}^\top \mathbf{s}_n + b), \tag{8.2}$$

where $L(y, t)$ denotes the *loss* to classify a sample as $\operatorname{sgn}(t)$ when the true class is y. We use $L(y, t) = max(0, 1 - yt)^2$, but other loss functions are possible as well. The parameter C balances the two conflicting training goals: a large value puts more emphasis on class separation, while a small value results in larger margins.

Traditionally equation (8.2) is minimized by solving a dual problem. However, it has been shown that if the number of dimensions D is small in comparison to the number of training samples N, primal optimization is both faster and produces more stable results [9].

There are also extension to nonlinear classification by using kernel functions to implicitly map features into a higher-dimensional space,

where linear separation is possible. However, as the dimensionality of the spectral signatures is already very high and because it can reasonably be assumed that the signatures are normally distributed and therefore linearly separable, we chose to stick with linear SVMs.

Multiclass SVM

Equation (8.1) formalizes SVM as a binary classifier, but the goal is to classify spectral signatures into one of (very) many different classes. A straightforward way to extend SVM to multi-class problems is to train multple classifiers $h_{ij}(\mathbf{s})$ that separate class i from class j – one for each unordered pair of classes. Classes are assigned by majority-vote[4]:

$$H(\mathbf{s}) = \arg\max_y \sum_{i,j} \mathbf{1}[h_{ij}(\mathbf{s}) = y]. \tag{8.3}$$

While simple, this *one-vs-one* scheme has two key advantages over a closed form multi-class SVM formulation: Firstly, when novel classes are introduced only parts of the classifier have to be retrained. Secondly, it allows to exploit prior knowledge in classification: When one knows a list of candidate classes for a given material (e.g. minearals, plastics), only a small number of classifiers has to be evaluated, which decreases computation time as well as the risk of misclassification. This comes at the cost of increased computation effort in training and classification. However, as classification only requires a single dot-product, this is not a major concern.

2.2 Random Forests

Random forests have become popular alternative to SVMs. The training algorithm is relatively simple, yet provides good results out of the box with little to no parameter tweaking. A random forest is an ensemble of decision trees, where each tree h_t is trained separately on a different random subset of the training samples. The nodes k in each tree correspond to thresholds τ_k and features s_k. The restulting binary test $s_k < \tau_k$ divides the training set for $\mathcal{D}_k$ that node into $\mathcal{D}_k^+$ and $\mathcal{D}_k^-$. It is chosen

[4] We abuse notation and write $h_{ij}(\mathbf{s}) = y$ to denote that h_{ij} voted in favor of class y.

to minimize the gini impurities, i.e. with $\mathcal{D}_k^\circ$ denoting either split:

$$I_G(\mathcal{D}_k^\circ) = 1 - \sum_y \left(\frac{\sum_{\mathbf{x}_n \in \mathcal{D}_k^\circ} \mathbf{1}[y_n = y]}{|\mathcal{D}_k^\circ|} \right)^2. \tag{8.4}$$

The training procedure is repeated recursively on the two child nodes until some stopping criterion is met. To encourage diversity in the trees of the ensemble, only a random subset of N_f features (usually $N_f = \lceil \sqrt{D} \rceil$) is considered for each split. After construction, the M_t leaf nodes of the tree correspond to disjoint regions $\mathcal{R}_{tm}$ in the feature space; each region is associated with an estimate of class membership probability

$$\hat{p}_{tm}(y) = \frac{1}{|\mathcal{R}_{tm}|} \sum_{\mathbf{s}_n \in \mathcal{R}_{tm}} \mathbf{1}[y_n = y]. \tag{8.5}$$

The class of a sample is predicted by traversing each tree and averaging the reached probability estimates,

$$H(\mathbf{s}) = \arg\max_y \hat{p}(y|\mathbf{s}), \text{ where} \tag{8.6}$$

$$\hat{p}(y|\mathbf{s}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M_t} \mathbf{1}[\mathbf{s} \in \mathcal{R}_{tm}] \; \hat{p}_{tm}(y). \tag{8.7}$$

As this procedure only involves thresholds and sums, RFs are very fast classifiers. Training and prediction are also trivial to parallelize by training or traversing each tree in its own thread. Furthermore, each tree can in principle be inspected and the decisions it takes can be understood. However, the main advantage over SVMs is that RFs are inherently multi-class classifiers. Yet in our case this can also be seen as a drawback, as prior knowledge cannot easily be incorporated and the whole classifier has to be retrained when novel classes are introduced.

3 Experiments

We compared the suitability of both methods using two subsets of our hyperspectral database [1]. We constrained our experiments to measurements in the near infrared spectrum; the dimensionality of the spectral signatures was $D = 224$. The first subset contained 13 classes and of

Table 8.1: Overview over the dataset used in this study.

Samples in set A		Additional samples in set B	
Material class	**Samples**	**Material class**	**Samples**
Acolon grapes	37 933	Dornfelder grapes	41 688
Calcite	190 656	Hard candies	20 731
Chalcedony	117 652	Hazelnuts	34 911
Dolomite	300 286	Lemberger grapes	77 388
Ivy leaves	108 859	Milk	123 308
Maple leaves	129 987	Müller-Thurgau	109 695
Magnesite	565 639	Pinot blanc grapes	720 936
Pinot gris grapes	91 389	Pinot meunier grapes	41 315
Pinot noir grapes	688 459	Pinot noir précoce grapes	73 667
Quartz	118 675	Riesling grapes	810 418
Sausage	344 588	Serpentine	421 729
Talc	136 520	Trollinger grapes	189 095
White bread	430 960	Wheat flour	112 132
		White Sugar	139 945

more than 3 000 000 samples. The second subset contained 14 additional materials, resulting in 27 classes and more than 6 000 000 samples in total. Table 8.1 lists the used materials and number of samples for each class. The dataset contains very different (e.g. *Quartz*, *Ivy leaves*), very similar (wine varieties, minearals) and very loosely defined (*Sausage*, *White bread*) materials. It is also heavily imbalanced: *Riesling*, for example, is represented by twenty times more samples than *Acolon*.

We performed stratified 5-fold cross-validation in all experiments. With linear SVM, we set $C = 1$ and did not perform any parameter tuning. With RF, we set the number of trees to $T = 10$ and experimented with different maximum tree depths d as stopping criterion.

With 13 classes, training all 78 SVM classifiers took $42min$ on a $2.6GHz$ 16-core Intel Xeon CPU. This amounts to roughly $30s$ per classifier or $0.2ms$ per training sample. Training a RF with $d = 10$ required $41min$ ($0.9ms$/sample) to complete, while a RF with $d = 20$ increased training time to $52min$. On average, classifying a single sample required $171\mu s$ with one-vs-one SVMs, while with RFs it took only $8\mu s$ resp. $10\mu s$ with $d = 10$ and $d = 20$ respectively.

Classifier performance is summarized in Table 8.2. It can be seen that

Table 8.2: Overview over the results of our experiments.

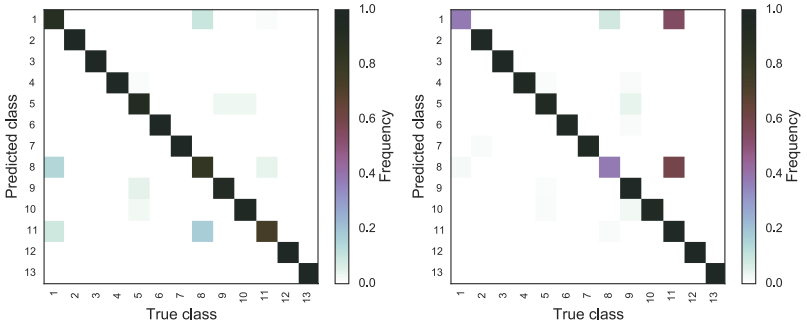
Dataset	Classifier	Mean F_1	Min F_1
A	SVM	$0.95_{\pm 0.0003}$	$0.79_{\pm 0.001}$
A	RF ($d = 10$)	$0.85_{\pm 0.003}$	$0.59_{\pm 0.025}$
A	RF ($d = 20$)	$0.90_{\pm 0.001}$	$0.49_{\pm 0.007}$
B	SVM	$0.79_{\pm 0.0002}$	$0.28_{\pm 0.002}$
B	RF ($d = 20$)	$0.74_{\pm 0.001}$	$0.03_{\pm 0.003}$
B	RF ($d = 30$)	$0.79_{\pm 0.001}$	$0.16_{\pm 0.005}$
B	RF ($d = 50$)	$0.79_{\pm 0.001}$	$0.22_{\pm 0.005}$

the SVM approach consistently outperforms RFs, but the results suggest that RF can reach similar performance with deeper decision trees. However, deeper trees will also significantly increase the training time.

Figure 8.1 shows confusion matrices for the SVM and RF classifiers in the 13-class experiment. Overall the SVM classifier is relatively consistent; most often confused are the classes 1, 8 and 11 (*Acolon*, *Pinot gris* and *Pinot noir*) as well as the minerals *Quartz*, *Magnesite* and *Chalcedony* (classes 5, 9 and 10). RF is more robust towards the latter, but more susceptible to confuse the wine varieties to the point where *Acolon* and *Pinot gris* are more often classified as *Pinot noir* than the true class. Similar results can be observed in Fig. 8.2: Classification is quite consistent and confusion is concentrated on similar material groups (wine varieties and minerals). Here, too, random forest is more more susceptible to misclassification than one-vs-one SVM.

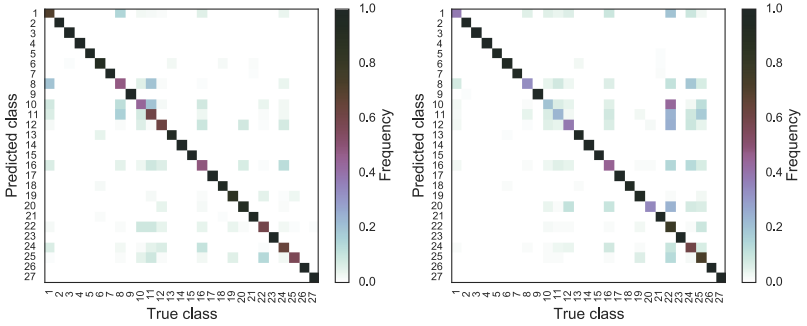
4 Summary

In this paper, we investigated the suitability of linear support vector machines and random forest classifiers for the task of discriminating a large number of materials using their spectral signatures. While RF supports multi-class classification out of the box, SVM does not. To achieve multi-class classification, multiple linear SVMs were combined to a single classifier in a one-vs-one scheme. Experiments showed that the SVM approach is superior both in classification performance and training time. The former can be attributed to multiple classifier fusion, while the latter is a result of the *primal* (instead of the usual *dual*) for-



(a) SVM (b) RF ($d = 20$)

Figure 8.1: Selected confusion matrices in the 13-class experiments.



(a) SVM (b) RF ($d = 50$)

Figure 8.2: Confusion matrices in the 27-class experiments.

malization of the training goal. Classification time, on the other hand is significantly slower. However, since it is still performed in the order of a few hundred microseconds, this is not too big of a concern. In light of these results, we suggest to prefer SVM over RFs. This has the additional benefit that prior knowledge about the material class can easily be incorporated in classification.

In the future, we expect to be able to reduce training time and increase classification performance by suitable pre-processing methods (e.g. whitening, dimensionality reduction) and automatic parameter

tuning. Additionally, we would like to explore the potential of the recently proposed extreme learning machines (ELM) [10] as an alternative to support vector machines. Preliminary experiments with ELM have shown promising results in that regard.

References

1. S. Irgenfried and C. Negara, "A framework for storage, visualization and analysis of multispectral data," in *OCM 2013 - Optical Characterization of Materials - conference proceedings*, 2013, pp. 203–214.
2. P. K. Varshney and M. K. Arora, *Advanced image processing techniques for remotely sensed hyperspectral data*. Springer, 2004.
3. A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote sensing of environment*, vol. 113, pp. S110–S122, 2009.
4. G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.
5. F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
6. P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 294–300, 2006.
7. V. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93–104, Jan. 2012.
8. D. Lorente, N. Aleixos, J. Gómez-Sanchis, S. Cubero, O. L. García-Navarrete, and J. Blasco, "Recent Advances and Applications of Hyperspectral Imaging for Fruit and Vegetable Quality Assessment," *Food and Bioprocess Technology*, vol. 5, no. 4, pp. 1121–1142, Nov. 2011.
9. O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.
10. G.-B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, pp. 1–15, 2014.