CONCEPT FOR A COMPOUND ANALYSIS IN ACTIVE LEARNING FOR REMOTE SENSING

S. Wuttke[a, b, *], W. Middelmann[a], U. Stilla[b]

[a] Fraunhofer IOSB, Gutleuthausstr. 1, 76275 Ettlingen, Germany - (sebastian.wuttke, wolfgang.middelmann)@iosb.fraunhofer.de
[b] Technische Universität München, Boltzmannstraße 15, 85748 Munich, Germany - (sebastian.wuttke, stilla)@tum.de

KEY WORDS: Analysis, Active Learning, Remote Sensing, Framework, Usability

ABSTRACT:

Active learning reduces training costs for supervised classification by acquiring ground truth data only for the most useful samples. We present a new concept for the analysis of active learning techniques. Our framework is split into an outer and an inner view to facilitate the assignment of different influences. The main contribution of this paper is a concept of a new compound analysis in the active learning loop. It comprises three sub-analyses: structural, oracle, prediction. They are combined to form a hypothesis of the usefulness for each unlabeled training sample. Though the analysis is in an early stage, different extensions are highlighted. Further we show how variations inside the framework lead to many techniques from the active learning literature. In this work we focus on remote sensing, but the proposed method can be applied to other fields as well.

1. INTRODUCTION

The success of supervised learning not only depends on the availability of labeled training examples, but also on the usefulness for the chosen classifier. Acquiring the labels for training samples results often in high costs in the form of resources, money or human annotation time (Settles, 2009). Especially in remote sensing acquiring the correct labels for the training data is expensive because it often involves ground surveys. Therefore it is important to chose as few samples as possible concentrating on the most informative samples first. Another frequently used form of acquiring ground truth for remotely sensed data is the visual inspection of aerial images by a human annotator. This method is cheap compared to a ground survey, but can lead to redundant training sets because pixels are often labeled by mass selection. This results in large training set sizes which considerably slow down the training phase of the algorithm. Therefore one should also concentrate on the most useful samples first and choose as few as possible.

Therefore it is desirable to chose as few samples for labeling as possible while simultaneously retaining high representativeness for good classification accuracy. Active learning (AL) achieves this by providing means to calculate the usefulness of samples and presents strategies to select samples for labeling in different scenarios of supervised classification tasks.

This process generally repeats the following three steps until a stopping criterion is met:

1. Evaluate usefulness of all samples
2. Select one and retrieve the true label
3. Train a supervised classifier

The core challenge is the definition of the usefulness heuristic. It can be defined with a confidence or certainty measure (Lewis and Catlett, 1994), by disagreement of a committee (Seung et al., 1992) or with diversity criteria (Brinker, 2003). The true label for the selected sample can be queried from the so called oracle. In most cases this is a human annotator. Examples are labeling remotely sensed images (Tuia et al., 2011), flagging e-mails as spam or tagging audio samples. Other forms of oracles are also possible. In the case of (King et al., 2004) the oracle was a "robot scientist" which executes autonomous biological experiments.

The used classification algorithm can be any supervised learning method. Common choices are the nearest neighbor classifier (Wuttke et al., 2012), support vector machines (Schölkopf and Smola, 2002), or Bayes classifier (Roy and McCallum, 2001). The results of the Active Learning Challenge (Guyon et al., 2011) present an overview of currently used algorithms.

Active learning can further be divided in three scenarios:

- **Pool-based:** All (or a large pool) of the unlabeled samples are available for selection.
- **Stream-based:** Samples are only available one at a time and must be selected or dismissed directly (Cohn et al., 1994).
- **Query-synthesis:** The queried samples are generated de-novo (Angluin, 1988).

The first scenario is the most common one (Settles, 2009). In remote sensing the available image data represents the pool from which the selection strategy can freely select samples to be annotated. As such this paper focuses on the pool-based scenario. Chapter 3.2 describes how the other scenarios fit into the proposed framework. An in depth look at active learning and different selection strategies is given in (Settles, 2009).

The notation used in this paper describes an active learning algorithm as a quintuple (C, O, U, L, H). $C_L : \mathcal{X} \rightarrow \Omega, x \mapsto y$ is a supervised classifier trained with the labeled training data $L = \{(x_i, y_i)\}_{i=1}^{l}$ with $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$, the d-dimensional feature space and $y_i \in \Omega = \{1...c\}$, the set of all classes. The oracle $O : \mathcal{X} \rightarrow \Omega, x \mapsto y$ can be queried to get the label for unlabeled data from $U = \{x_i\}_{i=l+1}^{l+u}$ with $u \gg l$. The samples are selected according to a ranking of the usefulness given by the hypothesis $H : \mathcal{X} \rightarrow \mathbb{R}, x \mapsto b$.

*Corresponding author.

The framework presented in this paper is divided into an outer and an inner view. The outer view encapsulates all feature extraction, splitting in training and test data, and quality assessment. From this view the active learning process is a black box. The inner view describes the active learning algorithm consisting of the three steps mentioned above. The specific choice of the classification algorithm is given from outside of this framework and is not in the scope of this paper. It is assumed that the classifier is suitable for the given problem.

The division into these two views allows to distinguish clearly between the consequences of variations made to the individual parts. It is also easier to communicate at what times which information is available. Without this partition it is more difficult to allocate causes and effects.

In a more strict form, no information is shared between the two views other than the unlabeled training data before and the trained classifier after the iterations. Though in most cases additional meta-information is available and can be shared from the outer to the inner view. Such variations are described in the corresponding chapters.

The remainder of this paper is structured as followed: Section 2. describes the outer view in more detail. Section 3. describes the inner view. The main contribution of this paper, the compound analysis, is detailed in section 4. Section 5. discusses current problems from the active learning literature and how they can be addressed with the proposed compound analysis.

2. OUTSIDE VIEW

Individual components of the outer view are shown in figure 1 and are detailed in this section. On this layer of abstraction active learning is viewed as a black box. All described components of this view would work the same with a passive learning scheme. Costs generated by components in the outer view can not be attributed to the active learning, because they are necessary even in a *passive* learning scheme. Namely these are costs for feature extraction and determination of classification accuracy, which needs labeled samples that were not used for training.

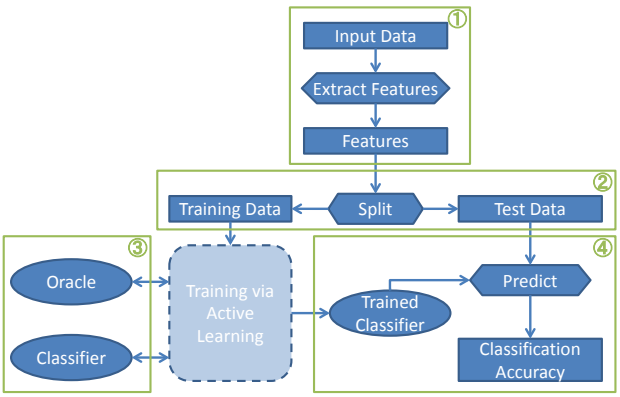

Figure 1. Outer view of the concept for active learning evaluation. The four components are detailed in their respective sections. On this level of abstraction active learning is viewed as a black box. The components would work in the same way if no *active* process were present. Therefore all costs associated with them are not attributed to the active learning scheme (i.e. costs to determine classification accuracy).

2.1 Feature Extraction

The input can be arbitrary data (e.g. images, multi- or hyperspectral data, video, 3d data). Task of the feature extraction is to transform the input data into d-dimensional features $\{x\} \in \mathcal{X} \subseteq \mathbb{R}^d$. The interface to the next component is a $d \times m$ matrix which contains all m samples as column vectors.

Possible feature extraction techniques include segmentation or edge detection on images. An often used feature for multispectral data is the normalized difference vegetation index (NDVI). Band selection (Maerker et al., 2011) is often used to reduce computational effort when working with hyperspectral data.

Variations

Segmentation In the simplest case for remote sensing images, one pixel corresponds to one sample. Due to the increasing spatial resolutions in remote sensing applications, the same ground area is mapped onto more pixels . Therefore the probability increases that neighbored pixels belong to the same class, "Smoothness Assumption" in (Schindler, 2012). Segmentation can alleviate this (Lee and Crawford, 2005), but has to be represented by a corresponding feature extractor.

Inter Sample Relations It is interesting to model relations between different samples (e.g. temporal correlations in video). As long as the feature extractor encodes this in one column vector per sample, it can be used in this framework.

2.2 Data Splitting

This component splits the data into to disjoint sets for training and testing. By moving this into the outer view, the actual classification algorithm can use all data its provided with. Otherwise much care must be taken to not introduce information from the testing samples into the training process such as their distribution or concrete values, as this would lead to unreliable performance measures.

Variations

Cross-Validation To further improve the reliability of the performance n-fold cross-validation can be conducted. This can be done transparently in this step of the process without changes to parts of the inner view.

2.3 External Information

This component consists of the oracle and the classifier. This section details why they are considered external and not part of the framework.

The type of oracle strongly depends on the type of data used. Further to the different oracles described in the introduction there can be other scenarios. For example multiple oracles working in parallel or with great latency. To simplify the framework all these choices are not modeled and made beforehand. In this work the oracle is automated in the form that ground truth (the true labels) is known in advance, but held back from the classification algorithm. It is revealed automatically only for samples queried from the oracle. This enables evaluation without human interaction during development. This is a common approach to automate the testing of active learning algorithms (Guyon et al., 2011). In real-world applications the querying of the oracle creates costs. Details on query costs are described in section 3.2.

Many different classifiers are currently used in the literature for active learning on remotely sensed data (Tuia et al., 2011), (Crawford et al., 2013), (Persello and Bruzzone, 2014). Today the choice of the classifier has more influence on the performance than the choice of the AL strategy (Guyon et al., 2011). One goal of this framework is to focus on the analysis of the other steps of the active learning process. Therefore the supervised classification algorithm is outside of the scope of this framework and is considered as external for our study. Its properties are known beforehand and depending on those meta-information different implementations of the parts of the framework are possible. Future analysis should make the performance more independent from the choice of the classifier.

Variations

Examples for such meta-information and their influence on the framework:

- **No meta-information:** Some optimizations of the prediction analysis in section 4.3 are not possible.
- **Number of classes:** If known, the analysis of the structure 4.1 and oracle 4.2 can be optimized.
- **Type of classifier (generative vs. discriminative):** If a generative classifier is chosen, information about the model (e.g. mean and variation of a normal distribution) can improve the active learning process.
- **Support of rejection class:** The handling of outliers is different, which influences the prediction analysis 4.3.
- **Availability of certainty measure:** If the classifier does not provide an inherent certainty measure, different strategies for the prediction analysis are needed.

2.4 Quality Assessment

The assessment of classification accuracy needs the true labels of the test samples. The costs associated with acquiring these are not attributed to the active learning process. Therefore the quality assessment is located in the outer view. This means all relevant costs are coming only from the inner view.

There are different metrics for measuring the performance and quality of classification algorithms. The first two are also used for passive learning. Whereas the second two are only defined for active learning because they depend on the number of used training samples:

- **Confusion matrix:** Used in multiclass problems to show inter-class misclassification.
- **Receiver operating characteristic curve (ROC curve):** Used in two-class classification problems to show the dependency between true positive rate and false positive rate. Extensions for multiclass problems exist (ROC surface), but are beyond the scope of this paper.
- **Correctly classified curve:** Used in two- and multiclass problems. Plots fraction of correctly classified samples against number of training samples used.
- **Learning curve:** Used in two-class problems. Plots area under the ROC curve versus the logarithm of number of used training samples (Guyon et al., 2011).

Apart from the above mentioned metrics, it is possible to define scalar metrics which combine the performance of a classifier into a single number:

- **Area under ROC curve (AUC):** Used for two-class problems. Based on the ROC curve. Larger is better.
- **Volume under ROC surface (VUS):** Used for multiclass problems. Based on the ROC surface (Ferri et al., 2003). Larger is better.
- **Area under learning curve:** Used for two-class problems. Based on the learning curve from above. Larger is better. Used in the Active Learning Challenge (Guyon et al., 2011).

3. INNER VIEW

The inner view represents the common active learning loop:
1) evaluate usefulness, 2) select samples, and 3) train classifier. Figure 2 shows the interaction between the parts. Each part is described in the following sections. The main focus of this paper is the usefulness hypothesis and is detailed in section 4.

As described before, the main advantage of the separation in an inner and outer view is the possibility to account for the different influences. As shown in figure 2 the only information known to the active learning process is the unlabeled training data and possibly some meta-information from the classification algorithm or oracle.

Many definitions of usefulness are based on the current model of the classifier: margin sampling (Schohn and Cohn, 2000), uncertainty sampling (Lewis and Catlett, 1994), query by committee (Seung et al., 1992). This leads to a problem at the first iteration(s) of the learning loop as there are no or not enough samples labeled yet. This is also known as ”cold start” problem (Zhang et al., 2014). Often this is solved by selecting a few random samples and labeling them. Other solutions are semi-supervised (Zhang et al., 2014) and unsupervised techniques (Cebron, 2008).

3.1 Analysis

The central idea of the compound analysis is a usability measure for unlabeled samples. If it is combined with a maximum selection strategy it is similar to the maximization of the expected

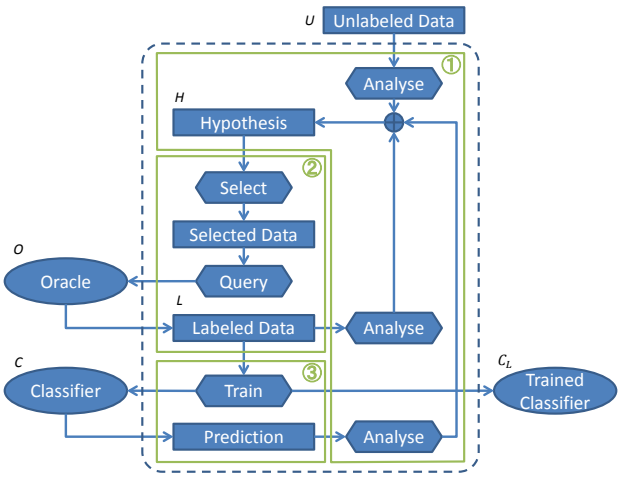

Figure 2. The inner view represents the common active learning loop: evaluate, select & query, and train. The different parts are labeled according to the quintuple (C, O, U, L, H) in the notation as presented in the introduction.

information gain, (Seung et al., 1992) or minimizing the expected error (Roy and McCallum, 2001).

The usability heuristic is defined as a function $H : \mathcal{X} \rightarrow \mathbb{R}$ which returns a usability b for each sample x. This function is a combination of different sub functions, hence the term compound analysis. It is described in detail in section 4.

3.2 Selection and Query

This work uses a compound analysis to determine the usefulness of each sample. As a result selecting the best sample becomes a simple maximum decision. Therefore the selection strategy chooses the k best samples and queries the oracle for their true labels. The choice of $k = 1$ re-evaluates the usefulness for each new labeled sample and can therefore use the most information. As this requires the re-training of the classifier for each selected sample, it is often computational too expensive. This can be alleviated by using classification algorithms which support incremental training (e.g. nearest neighbor). The solution in most active learning literature is to choose $k > 1$, see (Guyon et al., 2011). (Chakraborty et al., 2014) integrate the choice of an ideal k into their heuristic of the usefulness. For a choice of $k > 1$ it is possible that the selected samples contain redundant information. Diversity based heuristics alleviate this (Brinker, 2003).

Variations

Variable Querying Costs So far we assumed that each queried sample produces the same costs for the oracle. Therefore it can be beneficial to include a cost term in the selection strategy: $c_1 + n * c_2$. This approach allows to model different scenarios, some examples are given below:

- $c_1 \approx 0, c_2 \approx 0$: No need for active learning. Many samples can be cheaply queried at once.
- $c_1 \gg c_2$: Acquiring a satellite image with additional information from which the true labels can be calculated cheaply.
- $c_1 \ll c_2$: Satellite image available, but has to be labeled manually.

(Demir et al., 2014) implements variable query costs for ground surveys by using ancillary data like road networks and digital elevation models.

Query Types Two very different query types can be identified: 1) Sample given, label wanted 2) Label given, sample wanted. The first type is most often used in the active learning literature. It represents the learner selecting a sample and querying the oracle for the corresponding label. The second type is most useful if the classes have a very uneven distribution, that can not be reproduced from the data alone or if some classes are not yet discovered. In this case the learner would identify an underrepresented class and query the oracle for a sample of this specific class. A remote sensing example would be the task for the human annotator to label more pixels of the vegetation class. This type of query results in much greater costs for the oracle because it has to search multiple samples before it can return one that fits the query. Therefore these two types should be modeled separately.

Stream-based AL If the unlabeled samples arrive one at a time and can not be saved, an immediate decision has to be made to query the sample or not. Real world examples for this are part-of-speech tagging (Dagan and Engelson, 1995) and sensor scheduling (Krishnamurthy, 2002). This scenario is called stream-based active learning (Settles, 2009). A naive method to incorporate this into the framework is to implement a threshold t based on the utility of the current sample and query the oracle only if $H(x) > t$.

Incorrect Oracle So far we assumed that the labels received from the oracle are always correct. If this assumption is dropped, one has to estimate the confidence of the oracle at the same time as trying to learn from the given noisy labels. This problem is known as the multi-armed bandit problem (Beygelzimer et al., 2011), but is beyond the scope of this work.

Semi-Supervised Learning Replacing the oracle O with the classifier C results in classical semi-supervised learning. The main advantage is that no human interaction is needed. Semi-supervised learning can be seen as the opposite strategy to uncertainty sampling. In a semi-supervised setting the learner focuses on the samples of which the classifier is most confident. Whereas an uncertainty sampling strategy focuses on the samples of which the classifier is most uncertain.

3.3 Training

The supervised classification algorithm gets supplied with the samples and their corresponding labels and is trained in the same way as in a passive learning setting. Afterwards all samples (labeled and yet unlabeled) are classified and made available for the compound analysis of the next iteration. If the classifier provides a confidence or certainty measure about each classification this information can be used, too.

4. COMPOUND ANALYSIS

The compound analysis represents a hypothesis of the usefulness for each unlabeled sample at the current state of the learning process. It is defined as follows:

$$H(x) = h(f_S(x), f_O(x), f_C(x)), \tag{1}$$

where

x : Unlabeled sample $\in \mathcal{X}$
f_S : Usability from structural information
f_O : Usability from oracle information
f_C : Usability from classifier information

The exact definition of h to combine the sub functions is subject of ongoing research. With this approach different active learning scenarios can be modeled.

One of the earliest AL strategies are membership queries (Angluin, 1988). They are part of the query synthesis scenario. Their main feature is that the learning algorithm creates the queried samples de-novo. That means they are not necessarily from the set of unlabeled samples, but instead are synthesized. To implement query synthesis in the proposed framework the queried sample is defined by

$$x^* = \arg\max_{x \in \mathcal{X}} H(x) \tag{2}$$

In most cases this maximum can not be calculated in a closed form, because of the definition of the usability function h and its sub functions. Though different numerical optimization techniques like grid search or gradient descent are possible.

Variations

Spatial Information With suitable feature extraction even real-world spatial information can be incorporated into the usefulness measure. (Crawford et al., 2013) describe three ways: 1) minimize "travel distance", 2) minimize collocation, 3) incorporate segmentation problem. These methods could be incorporated by adding a fourth sub function into the compound function h.

4.1 Structure Analysis

This analysis extracts information from the structure of the unlabeled data via the definition of the function f_S. This kind of analysis is used in the active learning literature, but not as widely as one would suspect. (Guyon et al., 2011) report that only 57% of the participants of the Active Learning Challenge used information from the unlabeled data. If the compound function h is defined in a away that ignores f_O and f_C and only uses only f_S, the result is an unsupervised learning process.

Examples for the implementation of f_S are:

- **Clustering algorithms:** (Patra and Bruzzone, 2011) are using a cluster-assumption based approach and use a simple histogram-thresholding algorithm to evaluate the usefulness of the unlabeled samples.
- **Dimensionality changes:** Sudden changes in data dimensionality often imply a change in the underlying class. This is often exploited in manifold learning.
- **Density changes:** This is related to the cluster-assumption. Regions where the data density changes are often interesting for labeling and therefore should have a high usability.
- **Exploration vs. exploitation:** In early iterations exploration should dominate the usefulness, whereas in later iterations exploitation of already discovered structures should be increased (Cebron, 2008).

Even if some of the above usability measures do not result in a clear answer (e.g. no clearly distinguishable clusters, no apparent dimensionality changes) this information can be helpful. In this case the weighting of f_S in the compound function h should be reduced, so that the influence of other information sources is increased.

Variations

Reduced Sample Availability In some cases it is not possible to get the true labels for all samples. An example in remote sensing is a restricted area where a ground survey is impossible or no current aerial images are available. A solution therefor would be to query the nearest possible neighbor instead of the unavailable sample.

Unknown Class Count If the number of classes is unknown it is difficult to estimate the state of the exploration of the feature space or to use unsupervised methods like k-means clustering. One possible solution is to use hierarchical clustering to estimate the number of clusters in the given data. It starts with one cluster and subdivides it depending on the ratio of intra-cluster to inter-cluster similarities.

4.2 Oracle Analysis

The sub function f_O captures the agreement between the hypothesis before and after the oracle queries. If there is great disagreement the training of the classifier might be unnecessary. Instead the hypothesis should be updated and new samples queried. Some questions that should be answered are:

- Are the given labels as expected?
- Is the amount of labeled samples sufficient for the training of the classification algorithm?
- Is each class well enough represented, does one class dominate the rest or is one class underrepresented?
- Are all classes encountered or are some missing?
- Are there specific classes which get mixed a lot (e.g. analyze the confusion matrix)?

So far these questions seem to be unanswered in the field of active learning and mathematical definitions pose a challenging problem.

One method to answer these questions is to cluster the unlabeled samples and query each cluster center. If some clusters belong to the same class, try to combine them. Or suggest to the user to split the given class into sub classes based on the found clusters. Though this interaction is interesting, for now it is beyond the scope of this work.

Another method is to query multiple samples from one cluster and check if they are from the same class. If this is not the case, the cluster should be divided further. Alternatively it can be suggested to the user to combine the two classes. Otherwise one gains the information that the cluster assumption does not hold for this data set and the influence of the corresponding usability measures should be decreased in the compound function h.

Variations

Feasible Training Some classifiers need a minimum amount of labeled training data to make their training feasible. For others it is mandatory to have samples from all the classes. For example in a two class scenario algorithms which do not support a rejection class need samples from both classes. Novelty detection algorithms on the other hand are able to train even with samples from only one class.

Noisy Oracle In remote sensing the task of the oracle is often carried out by a human operator who annotates aerial images with the help of additional materials. Because the human is not error-free, modeling the confidence of the oracle can improve the classification performance. (Tuia and Munoz-Mari, 2013) show that the user's confidence needs to be learned in order for AL to be efficient. This can be adapted in the framework by changing the sub function f_O of the compound function h.

Feature Selection If the labels given by the oracle do not correspond to the expected (by the hypothesis) or even predicted (by the classifier) labels, one solution would be the automatic generation of new features. Though this topic exceeds the scope of this paper.

4.3 Prediction Analysis

Many classification algorithms provide a confidence score for their results. This score can be used to calculate the usability of samples. Uncertainty sampling is the primal example for this. If the sub function f_S and f_O are weighted with 0 and f_C is the only measure for the usability function h, this framework becomes an uncertainty sampling strategy.

Some example definitions of the usability sub function f_C are given below. They are suited for use with different classification algorithms.

- **K-nearest neighbor:** Disagreement among the k nearest neighbors with vote entropy. Larger disagreement means larger usability:

$$f_C^{\mathrm{NN}}(x) = \sum_{y \in \Omega} \frac{\mathrm{vote}(x,y)}{k} \log \frac{\mathrm{vote}(x,y)}{k}, \qquad (3)$$

where $\mathrm{vote}(x,y) = \sum_{i=1}^{k} \mathbf{1}_{\{N_i(x)=y\}}$ is the number of neighbors "voting" for label y. $N_i(x)$ is the label of the i-th nearest neighbor of x.

- **Two-class SVM:** A widely used measure is margin sampling (Mitra et al., 2004). The usability is based on the distance from the SVM hyperplane. The closer x is to the hyperplane, the more useful is it: $f_C^{\mathrm{SVM}}(x) = -|s(x)|$, where s is the SVM decision function:

$$s(x) = \sum_{i \in \{1 \ldots SV\}} \alpha_i y_i K(x, x_i) + b \qquad (4)$$

where SV is the number of support vectors and $K(x, x_i)$ is a kernel function, which defines the similarity between the sample x and the support vector x_i. $\alpha_i > 0$ and $y_i \in \{+1, -1\}$ are the coefficient and label of the i-th support vector.

- **Multiclass SVM:** If the two most probable classes are close in their posterior classification probability ("on a tie"), the usability of the sample is high (similar to the reasoning for query by committee). Following (Tuia et al., 2011) we define our usability measure using their "breaking ties" (BT) heuristic:

$$f_C^{\mathrm{BT}}(x) = -(p(y = \omega_1|x) - p(y = \omega_2|x)), \qquad (5)$$

where $\omega_1, \omega_2 \in \Omega$ are the first and second most probable classes for sample x and $p(y = \omega|x)$ is the estimated posterior probability of the SVM classification that sample x belongs to class ω. See (Platt, 2000) for details on SVM posterior probability estimation.

- **Maximum-likelihood:** The easiest way to integrate a maximum likelihood classifier into an AL process is uncertainty sampling (Wuttke et al., 2014). The transformation into a usability score follows the above definitions. Higher uncertainty leads to higher usability.

(Cebron, 2008) shows that using an exploration and exploitation phase is beneficial for active learning. The reasoning is that first the feature space should be "explored" as much as possible to discover all classes. Later the "exploitation" of the class borders decreases the generalization error. To incorporate his approach into this framework the function f_K can be defined using his "Active Learning Vector Quantization".

Another source of information during this part of the framework is the analysis of the generalization error of the trained classifier. This can be done by observing the prediction for the samples the classifier was trained with. If the predicted labels for training samples are different than the ground truth, the classifier has consistency errors or is generalizing too much.

Variations

Query by Committee Instead of using only one classifier as described above, one could form a committee of either multiple classifiers or different versions of the same classifier. The usability in this case is proportional to the disagreement of the committee and can be defined using (Settles, 2009) vote entropy:

$$f_C^{\mathrm{QbC}}(x) = \sum_{y \in \mathcal{C}} \frac{\mathrm{vote}_{\mathcal{C}}(x,y)}{|\mathcal{C}|} \log \frac{\mathrm{vote}_{\mathcal{C}}(x,y)}{|\mathcal{C}|}, \qquad (6)$$

where $\mathrm{vote}_{\mathcal{C}}(x,y) = \sum_{\theta \in \mathcal{C}} \mathbf{1}_{h_\theta(x)=y}$ is the number of "votes" that label y receives for sample x among the hypothesis in $\mathcal{C}$ and $|\mathcal{C}|$ is the committee size.

Less Information If no meta-information like confidence measures are available, the classification algorithm does not support a rejection class or the total number of classes is unknown, an appropriate definition of the sub function f_C is difficult. In this case it should be weighted less in the compound function h to accommodate this uncertainty. By doing this the rest of the framework can stay the same and changes in the results can be attributed to the relevant part.

5. DISCUSSION

(Cawley, 2011) reports that even simple random sampling approaches can be competitive with the top submissions of the Active Learning Challenge. Active learning has problems especially in small datasets and is outperformed by random sampling. They conjecture that poor selections at early stages in the training adversely affect the quality of subsequent selections. This hypothesis can be tested by implementing the function f_S in such a way that it deliberately gives high ratings to samples from unrelated feature space regions. This should result in in even worse performance. If on the other hand f_S incorporates diversity criteria the performance should increase. They further state that an effective active learning strategy must reach a near optimal trade-off between exploration and exploitation. A possible implementation can use the "weighted locking" scheme presented in (Wuttke et al., 2014).

(Persello and Bruzzone, 2014) found that diversity criteria can generally speed up the convergence of the iterative AL algorithm. This raises a very interesting question for further research: What influence have diversity criteria on the variance of AL results? Because early iterations extrapolate from very few samples AL is prone to selection bias. In fact, the very nature of the active selection strategy is to be biased. This should lead to a high sensitivity about the choice of the initial training samples.

They also found that AL was able to cope with ill-posed classification problems, which should be studied further. An example for an ill-posed problem are samples that are outliers. It is interesting to study how they effect diversity criteria. How susceptible are AL techniques to errors in the ground truth? Can this be compared with the noisy oracle scenario and are the same counter measures useful?

One problem from the remote sensing domain are mixed pixels. Though this should be eased with increasing spatial resolution it still poses a problem. One method to alleviate this in the case of hyperspectral data is unmixing to find the source materials the mixed pixel is composed of (Gross et al., 2012). Afterwards the pixel can be modeled with a multi label approach. How to incorporate this into the framework is an open question for now.

These and other questions show that still much research is needed on the field of AL. In our future work we plan on answering these questions with the help of the proposed compound analysis.

REFERENCES

Angluin, D., 1988. Queries and concept learning. Machine Learning 2(4), pp. 319–342.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L. and Schapire, R. E., 2011. Contextual bandit algorithms with supervised learning guarantees. In: Artifcial Intelligence and Statistics, JMLR, Vol. 15.

Brinker, K., 2003. Incorporating diversity in active learning with support vector machines. In: International Conference on Machine Learning, pp. 59–66.

Cawley, G., 2011. Some baseline methods for the active learning challenge. In: Journal of Machine Learning Research, Vol. 16.

Cebron, N., 2008. Aktives Lernen zur Klassifikation großer Datenmengen mittels Exploration und Spezialisierung. PhD thesis, Universität Konstanz, Konstanz.

Chakraborty, S., Balasubramanian, V. and Panchanathan, S., 2014. Adaptive batch mode active learning. IEEE Transactions on Neural Networks and Learning Systems.

Cohn, D., Atlas, L. and Ladner Richard, 1994. Improving generalization with active learning. Machine Learning 1994(15), pp. 201–221.

Crawford, M. M., Tuia, D. and Yang, H. L., 2013. Active learning: Any value for classification of remotely sensed data? Proceedings of the IEEE 101(3), pp. 593–608.

Dagan, I. and Engelson, S. P., 1995. Committee-based sampling for training probabilistic classifiers. In: International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, pp. 150–157.

Demir, B., Minello, L. and Bruzzone, L., 2014. Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method. IEEE Transactions on Geoscience and Remote Sensing 52(2), pp. 1272–1284.

Ferri, C., Hernández-Orallo, J. and Salido, M. A., 2003. Volume under the roc surface for multi-class problems. In: European Conference on Machine Learning, LNCS, Vol. 2837, Springer, Berlin, Heidelberg, pp. 108–120.

Gross, W., Schilling, H. and Middelmann, W., 2012. An approach to fully unsupervised hyperspectral unmixing. In: International Geoscience and Remote Sensing Symposium, IEEE, pp. 4714–4717.

Guyon, I., Cawley, G., Dror, G. and Lemaire, V., 2011. Results of the active learning challenge. In: Journal of Machine Learning Research, Vol. 16.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, Philip G K, Bryant, C. H., Muggleton, S. H., Kell, D. B. and Oliver, S. G., 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature 427(6971), pp. 247–252.

Krishnamurthy, V., 2002. Algorithms for optimal scheduling and management of hidden markov model sensors. IEEE Transactions on Signal Processing 50(6), pp. 1382–1397.

Lee, S. and Crawford, M. M., 2005. Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure. IEEE Transactions on Image Processing 14(3), pp. 312–320.

Lewis, D. D. and Catlett, J., 1994. Heterogeneous uncertainty sampling for supervised learning. In: International Conference on Machine Learning, Morgan Kaufmann, pp. 148–156.

Maerker, J., Groß, W., Middelmann, W. and Ebert, A., 2011. Hyperspectral band selection using statistical models. In: Defense, Security, and Sensing, SPIE Proceedings, SPIE.

Mitra, P., Uma Shankar, B. and Pal, S. K., 2004. Segmentation of multispectral remote sensing images using active support vector machines. Pattern Recognition Letters 25(9), pp. 1067–1074.

Patra, S. and Bruzzone, L., 2011. A fast cluster-assumption based active-learning technique for classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 49(5), pp. 1617–1626.

Persello, C. and Bruzzone, L., 2014. Active and semisupervised learning for the classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 52(11), pp. 6937–6956.

Platt, J. C., 2000. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: A. J. Smola (ed.), Advances in large margin classifiers, MIT Press, Cambridge, Mass., pp. 61–74.

Roy, N. and McCallum, A., 2001. Toward optimal active learning through monte carlo estimation of error reduction. In: International Conference on Machine learning.

Schindler, K., 2012. An overview and comparison of smooth labeling methods for land-cover classification. IEEE Transactions on Geoscience and Remote Sensing 50(11), pp. 4534–4545.

Schohn, G. and Cohn, D., 2000. Less is more: Active learning with support vector machines. In: International Conference on Machine Learning.

Schölkopf, B. and Smola, A. J., 2002. Learning with kernels: Support vector machines, regularization, optimization, and beyond. Adaptive computation and machine learning, MIT Press, Cambridge, Mass.

Settles, B., 2009. Active learning literature survey. Technical Report 1648, University of Wisconsin, Madison.

Seung, H. S., Opper, M. and Sompolinsky, H., 1992. Query by committee. In: Computational Learning Theory, ACM, pp. 287–294.

Tuia, D. and Munoz-Mari, J., 2013. Learning user's confidence for active learning. IEEE Transactions on Geoscience and Remote Sensing 51(2), pp. 872–880.

Tuia, D., Volpi, M., Copa, L., Kanevski, M. F. and Munoz-Mari, J., 2011. A survey of active learning algorithms for supervised remote sensing image classification. IEEE Journal of Selected Topics in Signal Processing 5(3), pp. 606–617.

Wuttke, S., Middelmann, W. and Stilla, U., 2014. Bewertung von strategien des aktiven lernens am beispiel der landbedeckungsklassifikation. 34. Wissenschaftlich-Technische Jahrestagung der DGPF.

Wuttke, S., Schilling, H. and Middelmann, W., 2012. Reduction of training costs using active classification in fused hyperspectral and lidar data. In: Image and Signal Processing for Remote Sensing XVIII, SPIE Proceedings, Vol. 8537.

Zhang, M., Tang, J., Zhang, X. and Xue, X., 2014. Addressing cold start in recommender systems. In: Special Interest Group On Information Retrieval, ACM, pp. 73–82.