CoConut: Co-Classification with Output Space Regularization

Sameh Khamis
sameh@umiacs.umd.edu
University of Maryland
College Park, MD 20742

Christoph H. Lampert
chl@ist.ac.at
IST Austria
Am Campus 1, 3400 Klosterneuburg

Abstract

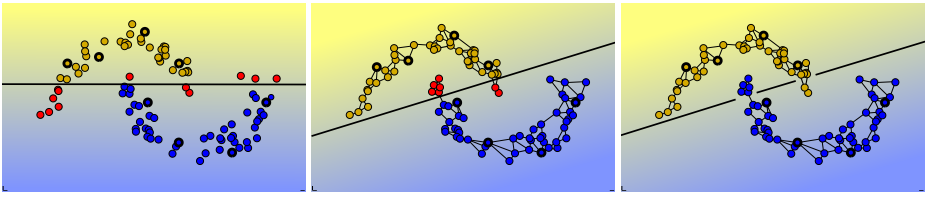
In this work we introduce a new approach to co-classification, *i.e*. the task of jointly classifying multiple, otherwise independent, data samples. The method we present, named CoConut, is based on the idea of adding a regularizer in the label space to encode certain priors on the resulting labelings. A regularizer that encourages labelings that are smooth across the test set, for instance, can be seen as a test-time variant of the cluster assumption, which has been proven useful at training time in semi-supervised learning. A regularizer that introduces a preference for certain class proportions can be regarded as a prior distribution on the class labels. CoConut can build on existing classifiers without making any assumptions on how they were obtained and without the need to re-train them. The use of a regularizer adds a new level of flexibility. It allows the integration of potentially new information at test time, even in other modalities than what the classifiers were trained on. We evaluate our framework on six datasets, reporting a clear performance gain in classification accuracy compared to the standard classification setup that predicts labels for each test sample separately.

1 Introduction

Classification is one of the most fundamental and best understood machine learning problems. Different scenarios, such as semi-supervised, multiple-instance, or active learning, differ strongly in their training procedure. They agree, however, in their prediction step at test time: each test sample is assigned a label individually. However, in many real-world applications the samples to be classified occur in batches, such as words in a document, images in a photo collection, or stocks in a portfolio.

Exploiting the fact that multiple test samples are available at the same time should make it possible to achieve increased classification accuracy. Consider the situation of a linear classifier, which is efficiently trainable and exhibits good generalization capabilities but has a decision hypersurface that might not perfectly reflect the class boundaries in feature space. Given sufficiently many test samples it should be possible to modulate the classifier's decision boundary, for example, based on the cluster assumption, which states that class decision boundaries typically do not cross high density regions (see Figure 1 for an illustration).

Despite its potential, the task of co-classification, *i.e*. classifying a set of points jointly, has received little attention in the literature. In this work, we introduce CoConut, a method for co-classification based on the established principle of regularized risk minimization. It



(a) no cluster assumption (CA) (b) CA at training time (c) CA at test time

Figure 1: Schematic illustration of the effect of the cluster assumption. Left: supervised training of a linear classifier with few training examples (bold circles): many mistakes occur at test time (red dots). Middle: imposing the cluster assumption during training adjusts the decision boundary thereby reducing the number of errors. Right: adopting the cluster assumption at test time allows non-linear corrections of the linear classifier decisions, resulting in a further reduction of errors.

jointly labels all test points by minimizing a regularized risk functional that consists of a per-sample loss and a regularizer that incorporates additional information in the output (label) space. The method is applicable to a wide range of classification problems, because it requires only the output of a set of classifiers as input, but makes no assumption on how they were trained. It is also efficient, as it requires no additional training step but only solves a regularized risk functional using efficient energy minimization techniques.

CoConut is able to exploit information absent during classifier training but available at test time. For instance, in robotics applications new sensors can be installed on a robot at the time of its deployment. While the data modalities acquired through these sensors were not present during training, it would still be beneficial if one could integrate them at operation time. Additionally, in graph or network applications, structural relations between data points are a natural source of information. For example, hyperlinks between webpages might indicate that those webpages share a topic. Similarly, with relational data: collaborating authors usually work in the same fields. Finally, having to train classifiers on imbalanced data is a common phenomenon in many settings. Imposing a preference for a certain label distribution at test time can ameliorate the damage of biased training data, which is possible with CoConut as well.

2 Regularized Co-Classification

We formalize the co-classification scenario in the following way. We are given a set of (test) examples, $X = \{x_1, \ldots, x_n\}$ from an input space $\mathcal{X}$, and we want to predict labels $Y = \{y_1, \ldots, y_n\}$ from a label set $\mathcal{Y} = \{1, \ldots, L\}$. For this task we have access to L fixed *base classifiers* with prediction functions, $f_1, \ldots, f_L : \mathcal{X} \to \mathbb{R}$, where for any $x \in \mathcal{X}$ and $l \in \mathcal{Y}$ the value $f_l(x)$ reflects a confidence that the sample x belongs to class l. The straight-forward choice for labeling the test points is then to predict (greedily) the most confident label for each sample, $y_i = \operatorname{argmax}_{l=1,\ldots,L} f_l(x_i)$.

We propose to compute a joint labeling $y^* = (y_1^*, \ldots, y_n^*) \in \mathcal{Y}^n$ of the test points by solving the following optimization problem:

$$y^* = \operatorname*{argmin}_{y \in \mathcal{Y}^n} \; -\sum_{l=1}^{L} [\![y_i = l]\!] f_l(x_i) + \lambda \Omega(y), \tag{1}$$

where Ω is a regularizer that penalizes undesirable label combinations and $\lambda \in \mathbb{R}^+$ is a constant that controls the regularization strength. We present several possible regularizers in Sections 2.2 and 2.3 that depend on the information available at test time. Note that for $\lambda \to 0$ we recover independent per-sample predictions, showing that per-example label selection can be thought of as a special case of this framework.

Equation (1) resembles the expressions occurring in the classical framework of *regularized risk minimization* [31]. The difference lies in the fact that we regularize in the output space (the space of all labelings), not in the space of classifier parameters. Therefore, we call the resulting approach *Co-Classification with output space regularization* (CoConut).

2.1 Theoretical Foundations

Statistical learning theory[1] studies the task of choosing a particular prediction function out of a (very large) set of hypotheses, using the information provided by a set of training examples. A central result is the following theorem:

Theorem 1 ([31]) *Let $\mathcal{H}$ be a set containing $|\mathcal{H}|$ hypotheses. Then the following inequality holds with probability at least $1-\delta$, uniformly for all $h \in \mathcal{H}$:*

$$\operatorname{err}(h) \leq \widehat{\operatorname{err}}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2n}}, \tag{2}$$

where $\operatorname{err}(h)$ is the expected error of the classifier h on future data, and $\widehat{\operatorname{err}}(h)$ is the empirical error of the classifier h on the training set of size n.

The theorem links the error on future data and the error on the training set. One consequence of the theorem is the *consistency* of empirical risk minimization: if $\log|\mathcal{H}|$ is small compared to the number of training examples, it is a good learning strategy to simply find the classifier that minimizes the training error [30]. Otherwise, this strategy might *overfit*: it might choose a classifier that works well on the training set, but not on the test set.

Co-classification fits the above framework when we interpret the classifier predictions $f_l(x_1),\ldots,f_l(x_n)$ as observed but potentially noisy labels. A joint labeling $y=(y_1,\ldots,y_n)$ corresponds to a discrete function, $g: X \to \mathcal{Y}$ with $g(x_i)=y_i$. Since all valid labeling are possible *a priori*, the hypothesis class is the set of all such functions, $\mathcal{G}=\mathcal{Y}^n$. Empirical risk minimization requires solving the following optimization problem

$$\min_{g\in\mathcal{G}} \mathcal{L}(g) \quad \text{with} \quad \mathcal{L}(g) = \sum_{i=1}^{n} \ell_i(g(x_i)), \tag{3}$$

where $\ell_i(l) = -f_l(x_i)$ is the loss of predicting the label y for the sample x_i. Unfortunately, for n (test) examples the hypothesis set $\mathcal{G}$ contains $|\mathcal{Y}|^n$ elements, so $\log|\mathcal{G}| = O(n)$, and the second term in the bound (2) cannot be neglected. Consequently, per-sample greedy prediction as in Equation (3) might *overfit*.

CoConut resolves the above situation by building on the concept of *regularization*. Instead of the hypothesis set $\mathcal{G}$ above, we search only labeling from a smaller set, $\mathcal{G}_C = \{g \in \mathcal{G} : \Omega(g) \leq C\}$ where Ω is a measure of regularity, and C is a task-dependent threshold. In the case where $|\mathcal{G}_C| \ll |\mathcal{G}|$, the bound (2) becomes tighter, and we can expect higher overall

[1] For simplicity of notation, we discuss the theory only for the case of binary classification, *i.e.* $\mathcal{Y}=\{\pm 1\}$.

classification accuracy when searching only for labelings that lie in the reduced hypothesis space.

In practice, solving Equation (3) over the reduced set $\mathcal{G}_C$ for fixed C results in a hard combinatoric optimization problem. We overcome this by reformulating the problem into one of *regularized risk minimization*. We first observe that finding $\min_{g\in\mathcal{G}_C} \mathcal{L}(g)$ is equivalent to the following optimization problem with side constraints,

$$\min_{g\in\mathcal{G}} \mathcal{L}(g) \quad \text{subject to } g \in \mathcal{G}_C, \tag{4}$$

which itself is equivalent to the unconstrained

$$\min_{g\in\mathcal{G}} \max_{\lambda\in\mathbb{R}^+} \quad \mathcal{L}(g) + \lambda(\Omega(g) - C) \tag{5}$$

where λ is a Lagrangian multiplier of the constraint (since $g \in \mathcal{G}_C \;\Leftrightarrow\; \Omega(g) \leq C$). Consequently, there is a value λ^* such that the same solution for g as of Eq. (4) is given by

$$\min_{g\in\mathcal{G}} \quad \mathcal{L}(g) + \lambda^*\Omega(g) + const. \tag{6}$$

This is equivalent to the optimization problem (1) we propose to solve for CoConut. Note that the dependence of λ^* on C is non-trivial and in general non-explicit. However, when C ranges over all of $\mathbb{R}$, so will λ^*, so it does not matter if we first perform model selection and then convert to the form of Equation (6), or vice versa.

2.2 Cluster Assumption

In our choice of regularizer we encode the *inductive bias* we have about the problem. Often this would be an assumption that the true labels vary smoothly with respect to the inputs, which means we should prefer smooth hypotheses $g \in \mathcal{G}$. The smoothness of a function is typically measured by (the integral over) its gradient norm. Since the elements of $\mathcal{G}$ are discrete functions, we use the following discrete analog of the gradient norm [33]. For any point x_i, let $N_i \subset X$ be the set of neighbors that are similar to x_i. Let w_{ij} denote the a measure of the similarity between two neighbors x_i and x_j. For any $x_j \in N_i$ the slope of g between x_i and x_j is $w_{ij}\delta_{ij}(g)$, where $\delta_{ij}(g) := [\![g(x_i) \neq g(x_j)]\!]$ indicates whether g changes value between x_i and x_j[2]. Averaging this quantity across all neighbors and all points, we obtain a measure for the average discontinuity (lack of smoothness) of any labeling function $g \in \mathcal{G}$:

$$\Omega_S(g) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|N_i|} \sum_{x_j\in N_i} w_{ij}\delta_{ij}(g). \tag{7}$$

Note that Ω_S coincides with a degree-normalized discrete Laplacian [1], defined by the weighted neighborhood graph of the samples in X. The measure of smoothness it provides is data-dependent: label changes in regions of high data density (neighboring points are similar to each other) are penalized strongly, whereas regions of low data density (neighboring points are less similar) are less constrained.

[2]The name slope stems from the case where $w_{ij} = \frac{1}{d_{ij}}$ with d_{ij} the distance between x_i and x_j. In this case, $w_{ij}\delta_{ij}$ is the geometric slope of the piecewise linear extension of g along the line connecting x_i with x_j.

Neighborhood Construction. We highlight three possible neighborhood system options: *a*) a k-NN graph with respect to the original feature representation, *b*) a k-NN graph with respect to an additional feature representation or data modality available only at test time, and c) a graph with neighborhood relations predetermined by side information. In all three situations, we define the weights using the radial basis function $w_{ij} = \exp(-\frac{1}{\sigma^2} d_{ij}^2)$ commonly used for graph Laplacians. where d_{ij}^2 is the distance between i-th and the j-th samples.

For the cases *a*) and *c*), we use $d_{ij}^2 = d_{\mathcal{X}}^2(x_i, x_j)$, *i.e.* we obtain the similarity between samples from their regular feature space distance. In case *b*), we assume that more information about the test examples is available than was the case at training time. We formalize this by assuming additional feature vectors, $z_i \in \mathcal{Z}$, for the test samples and define the distance between samples from the distance in the combined feature space $d_{ij}^2 = d_{\mathcal{X}}^2(x_i, x_j) + d_{\mathcal{Z}}^2(z_i, z_j)$.

Optimization. From Equations (7) and Equation (1) we obtain a discrete optimization problem for predicting the optimal labeling, y^*:

$$y^* = \operatorname*{argmin}_{y \in \mathcal{Y}^n} \sum_{i=1}^{n} \ell_i(y_i) + \lambda \sum_{i=1}^{n} \frac{1}{|N_i|} \sum_{x_j \in N_i} w_{ij} [\![y_i \neq y_j]\!] \tag{8}$$

where we have absorbed the factor $\frac{1}{n}$ of Ω_S into the constant λ. Optimization problems of the same structure as Equation (8) are known well in area of computer vision as problem of *energy minimization*. To solve them, we can rely on well-developed theoretical as well as algorithmic results. For $L = 2$, the objective function is submodular [13]. We can find the exact global minimizer using the mincut/maxflow duality [8] or LP relaxations [6]. For $L > 2$, the objective (8) coincides with the energy function of a generalized Potts model [32]. Convex relaxations or iterative techniques like α-expansion [4] find solutions that are guaranteed to be constant factor approximation of the optimum, but in practice are much better [11, 27].

Complexity. The regularizer Ω_S has been analyzed for binary labeling problem in the machine learning literature. In particular, it is proved in [22] that the size of the hypothesis set $\mathcal{G}_C$ is bounded by the value $\sum_{d=0}^{k} \binom{n}{d}$, where k is a constant depending on C and the eigenvalues of the graph Laplacian of the neighborhood graph. For small enough k this expression is exponentially smaller than 2^n (which is its value for $k = n$ and the size of $\mathcal{G}$ in this case).

2.3 Class Label Distribution

A regularizer can also encode a preference for a certain class label distribution at test time. This can counter the effect of the bias introduced by training with imbalanced class distributions. We assume that the target (expected) class label proportion for class l is Q_l, where $\sum_{l=1}^{L} Q_l = 1$. We define a measure for the disparity between the class label proportion $p_l(g)$ induced by labeling function g and the target proportion for each class l:

$$\Omega_D(g) = \sum_{l=1}^{L} |p_l(g) - Q_l| \tag{9}$$

where $p_l(g) = \frac{1}{n} \sum_{i=1}^{n} [\![g(x_i) = l]\!]$ are the label proportions the hypothesis g. The regularizer Ω_D penalizes the deviation from the target distribution, and this penalty is linear in the amount of deviation.

Optimization. We express the regularizer Ω_D in term of the labeling y and insert it into the CoConut objective. This results in the following discrete optimization problem.

$$y^* = \operatorname*{argmin}_{y \in \mathcal{Y}^n} \sum_{i=1}^{n} \ell_i(y_i) + \sum_{l=1}^{L} \xi_l \left(p_l(g) - Q_l\right), \tag{10}$$

where $\xi_1, \ldots, \xi_L$ are the Lagrangian multipliers of the label proportion constraints of the L classes. Model selection over all these parameters would be impractical. Instead, we infer their values during the optimization using *Lagrangian Relaxation* [2] as in [17] At each step, we alternatingly update the two variable sets y and ξ. To update y, we solve Equation (10) with fixed ξ, which corresponds to a per-variable minimization of the modified loss terms $\ell'_i(l) = -f_l(x_i) + \frac{1}{n}\xi_l$. To update ξ we take a subgradient step with step size α_t at iteration t. Given initial values for ξ_l, the minimization proceeds as follows:

- With fixed ξ_l, minimize loss terms ℓ'_i
- With fixed y, update $\xi_l \leftarrow \xi_l + \alpha_t(p_l(y) - Q_l)$

It is also possible to combine the two regularizers, Ω_D and Ω_S. In this case, the update step for y requires solving Equation (8) with the modified loss terms. The second step remains as it is. In this case the efficiency of the algorithm can be significantly improved by warmstarting the optimization for y with the solution from the previous iteration [12].

Complexity. The power of the regularizer Ω_D depends on the label proportions. For example, in the binary situation the size of the hypothesis space $\{g \in \mathcal{G} : \Omega_D(g) = 0\}$ is $\binom{n}{Q_1 n}$. This is significantly smaller than 2^n, if $Q_1 \ll Q_2$ or $Q_2 \ll Q_1$. Consequently, we expect the effect of label proportions regularizer to be most effective if the desired class distribution is imbalanced or in combination with another regularizer, such as Ω_S.

3 Related Work

To our knowledge, we are the first to propose a method for co-classification in a regularized risk minimization framework. However, the concrete techniques of graph-based regularization that we rely on have previously found application in machine learning, in particular transductive and semi-supervised learning [34]. In transductive learning, the test samples are known at training time. This allows integrating them in a neighborhood graph together with the training samples. Label information is propagated in this joint graph [3, 35]. We, on the other hand, do not assume that we have access to the test samples at training time. Semi-supervised learning typically learns a parametric classifier, but does so while taking the location of unlabeled examples into account. The cluster assumption, that we also rely on, is the most common method for this: a prediction function is preferable if it assigns similar values to samples that have a small distance from each other in feature space [1]. However, at test time predictions are made individually without further regularization of the outputs.

It is possible to combine multiple data modalities within a semi-supervised learning framework, but all of them must be present already at training time [26]. Christoudias *et al.* [6] recently proposed a technique for handling new modalities at test time. They train a classifier on the testing data to "hallucinate" the missing modality on the training data, and then train a classifier on both modalities. CoConut does not require any additional training and avoids estimating the missing modality on the training data.

Making joint predictions across multiple objects is a core component of structured prediction techniques, like CRFs [14] and structured SVMs [29]. However, such approaches typically model a set of objects as one example, and training then requires multiple sets of objects. A graph of fixed structure is typically required, whereas for CoConut we form a data-dependent graph at test time. Collective classification takes a similar route by using known relational properties between objects. Taskar *et al*. [28] models the sample relations using a Markov network trained using structured prediction. Lu and Getoor [19] train a classifier that makes per-sample prediction, but augment it with another classifier trained on additional neighborhood features. Our approach, on the other hand, does not require additional training and can complement the accuracy of pre-trained classifiers at little cost.

CoConut combines supervised inductive classifiers with an unsupervised similarity measure. This aspect resembles earlier work on combining supervised and unsupervised learning techniques. For example, Gao *et al*. [9] combines classifiers and clustering methods, by clustering the test set and forming a majority vote of the classifier outputs over the samples within each cluster. However, in the case where samples are erroneously clustered together with samples of a different class, errors are unavoidable. The regularization-based approach we take avoids hard decisions. It allows any incorrect neighbor relation to be overruled by confident classifiers scores, and vice versa.

4 Experiments

We evaluated CoConut on six different datasets: four image and two network datasets. For all datasets we take on the features provided by the original authors. The Robotics dataset consists of low-resolution (webcam) and high-resolution (DSLR) images. We use the former as original representation for classifier training and the latter as an additional representation available only at test time. For the other three image datasets, Flowers, Birds, and Butterflies, we follow the same setup using shape and texture features as the original modalities and color features as the additional modality. For the network data, Cora and Citeseer, the base representations are bag-of-word histograms [25], and additionally we impose a structural link between two documents whenever one of them cites the other. All features are reduced in dimension using Kernel PCA [24] to the dimensionality that minimizes the nearest neighbor error rate on the training set. We then train one-versus-rest linear SVMs for all classes using *liblinear* [7] and use their outputs as confidence values. To find the nearest neighbors, we compute the Euclidean distance matrices and normalize them to the $[0,1]$ range.

Supervised classification techniques typically adjust their parameters using a separate validation set. In the co-classification setup that we work in, such a validation set is not readily available, so for determining the single parameter we have, λ, we use part of the training set, essentially simulating co-classification on this set of examples instead of the test set. Clearly, this strategy is not ideal, since the output of the classifiers on the training set is not fully representative of their output on a test set, and in the case where additional modalities are available only at test time, we cannot integrate them at training time already. Nevertheless, we found the adopted model selection approach to work well in practice.

4.1 Results

We randomly split each dataset into equal training and test parts. On the former part we train linear classifiers in a standard supervised setting. We use their output for independent

Accuracy/Dataset	Robotics	Flowers	Birds	Butterflies
Baseline (transductive)	5.51 ± 0.00	5.88 ± 0.00	23.33 ± 1.83	21.75 ± 0.00
Baseline (per-sample)	63.86 ± 0.41	72.82 ± 0.23	54.40 ± 0.34	53.90 ± 0.24
CoConut (unseen)	64.56 ± 0.48	74.88 ± 0.23	**54.87 ± 0.35**	54.03 ± 0.27
CoConut (structural)	–	–	–	–
CoConut (proportions)	**64.76 ± 0.40**	**75.15 ± 0.22**	54.53 ± 0.29	**54.42 ± 0.34**

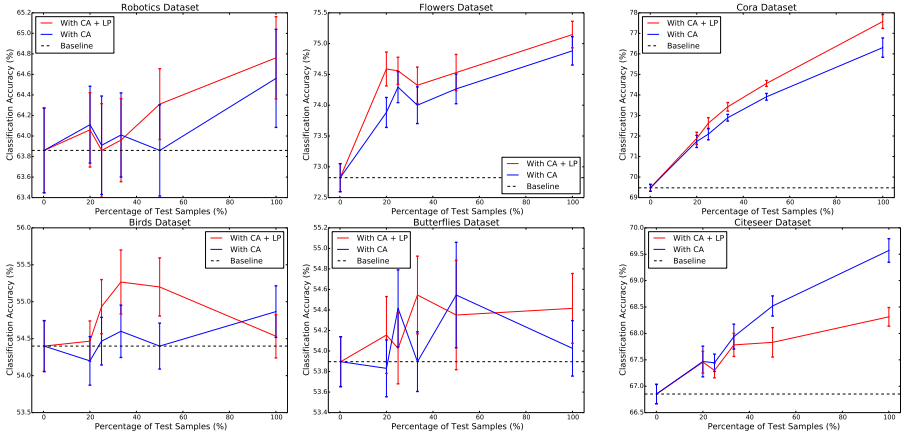
Accuracy/Dataset	Cora	Citeseer
Baseline (transductive)	30.23 ± 0.00	20.18 ± 0.00
Baseline (per-sample)	69.05 ± 0.23	66.74 ± 0.18
CoConut (unseen)	–	–
CoConut (structural)	76.30 ± 0.47	**69.57 ± 0.22**
CoConut (proportions)	**77.58 ± 0.33**	68.31 ± 0.18

Table 1: Classification accuracies on the six datasets (mean and standard error over five random splits). Baseline (transductive): label propagation across neighbor links. Baseline (per sample): independent classification. CoConut (unseen): with the cluster assumption on the unseen modality, CoConut (proportions): with both the cluster assumption and label proportions information, CoConut (structural): with the cluster assumption on structural links.

per-sample decisions, that serve as baseline, and as inputs to CoConut, which we test with the cluster assumption regularizer, based on the similarity of samples with respect to the new data modalities (for the image datasets) and from the structural relations (for the network datasets). We also evaluate CoConut with the label distribution prior on all six datasets. As a second baseline, we implemented label propagation (a transductive classifier) along a neighborhood graph constructed in the same way as for CoConut, but on all data samples, *train* and *test*, not just the *test* part.

Table 1 reports the results as mean and standard error of the mean over five random splits. A first observation is that the transductive classifier achieves clearly worse results than the other methods, even though it assumes that the training examples are available at test time. Presumably this is because it does not have the ability to distinguish between relevant and irrelevant dimensions in the feature space, while the discriminatively trained inductive classifiers do. One also sees that CoConut with the cluster assumption improves the classification accuracy consistently for all datasets. This is noteworthy, considering that the classifiers are not retrained, but only their output confidences are modulated using the cluster assumption at test time. Imposing a preference for the label distribution yields an even larger improvement across all datasets, except for Birds and Citeseer where CoConut with just the cluster assumption is the better performer.

We further evaluate the improvement with respect to the amount of test data available. The intuition behind this is that, *e.g.*, the cluster assumption can be expected to only be helpful if sufficiently many data points are available. Figures 2 shows the average results when splitting on the test set in non-overlapping groups of 20%, 25%, 33%, 50%, and 100% of test examples, and performing co-classification on each of the groups independently. The results confirm our hypothesis: for very small test sets, imposing a regularizer based on the cluster assumption can in some cases be disadvantageous. One possible reason for this is that the few data points are not sufficient to reliably estimate the cluster structure of the data. This effect, however, depends on the quality of the distance function. This is visible in the results for the Robotics and Flowers datasets, where the additional modality improves



(a) CA on a neighborhood graph (b) CA on a link structure

Figure 2: Classification accuracy (mean accuracy and standard deviation of the mean) of CoConut versus the accuracy of per sample independent prediction (baseline). The results for all datasets are reported using the Cluster Assumption (CA) and also using it in association with the Label Proportion information (LP). The right-most values in each figure corresponds to the results in Table 1. The rest of the values indicate problems made artificially harder by reducing the amount of test data available for co-classification.

the classification performance even for small test sets. This is even more apparent for the Cora and Citeseer datasets, where the relations are based on structural links. Similarly, the label distribution prior consistently improved the results across all test sizes. It was however slightly damaging to the results on Birds and Citeseer, where the cluster assumption was superior in isolation, while still clearly outperforming the baseline.

5 Summary

We presented CoConut, a technique for classifying multiple samples jointly. It combines pre-trained classifiers with a regularizer to impose favorable priors on the output space. Our method is general and independent of the choice of the classifiers. It is also efficient since it does not re-train those classifiers and only optimizes a regularized risk functional using scalable and efficient techniques. We can integrate additional information that was not present or useful at training time, such as measurement from additional sensors on a robot or timestamp and GPS information from a camera. We performed experiments on six different datasets in three different scenarios to evaluate CoConut, achieving consistent improvements over the baselines. In the future we plan to investigate imposing regularization on subsets of samples, whether given or obtained through clustering, as well as further applications of our model.

Acknowledgments

This work was in parts funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 308036.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[5] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1): 120–145, 2011.

[6] C. M. Christoudias, R. Urtasun, M. Salzmann, and T. Darrell. Learning to recognize objects from unseen modalities. In *ECCV*, 2010.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[8] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

[9] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, 2009.

[10] C. L. Giles, K. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Digital libraries*, 1998.

[11] Jörg H. Kappes, Bjoern Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Jan Lellmann, Nikos Komodakis, and Carsten Rother. A comparative study of modern inference techniques for discrete energy minimization problem. In *CVPR*, 2013.

[12] Pushmeet Kohli and Philip H.S. Torr. Efficiently solving dynamic Markov random fields using graph cuts. In *ICCV*, 2005.

[13] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[14] J. Lafferty, A. McCallum, and F Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, 2004.

[16] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *CVPR*, 2005.

[17] Yongsub Lim, Kyomin Jung, and Pushmeet Kohli. Energy minimization under constraints on label counts. In *ECCV*, 2010.

[18] W. Liu and S.-F. Chang. Robust multi-class transductive learning with graphs. In *CVPR*, 2009.

[19] Q. Lu and L. Getoor. Link-based classification. In *ICML*, 2003.

[20] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[21] M. E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006.

[22] K. Pelckmans, J. Shawe-Taylor, J. A. K. Suykens, and B. De Moor. Margin-based transductive graph cuts using linear programming. In *AISTATS*, 2007.

[23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[24] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[25] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[26] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *ICML Workshop on Learning with Multiple Views*, 2005.

[27] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008.

[28] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.

[29] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[30] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[31] V. N. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.

[32] F.-Y. Wu. The Potts model. *Reviews of modern physics*, 54(1), 1982.

[33] D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *DAGM*, 2005.

[34] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, 2005.

[35] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.