Learning Diverse Models: The Coulomb Structured Support Vector Machine

Martin Schiegg[1,2], Ferran Diego[1], Fred A. Hamprecht[1]

[1] *University of Heidelberg, IWR/HCI, 69120 Heidelberg, Germany*
firstname.lastname@iwr.uni-heidelberg.de
[2] *Robert Bosch GmbH, 70465 Stuttgart, Germany*

July 22, 2016

Abstract

In structured prediction, it is standard procedure to discriminatively train a single model that is then used to make a single prediction for each input. This practice is simple but risky in many ways. For instance, models are often designed with tractability rather than faithfulness in mind. To hedge against such model misspecification, it may be useful to train multiple models that all are a reasonable fit to the training data, but at least one of which may hopefully make more valid predictions than the single model in standard procedure.
We propose the Coulomb Structured SVM (CSSVM) as a means to obtain at training time a full ensemble of different models. At test time, these models can run in parallel and independently to make diverse predictions. We demonstrate on challenging tasks from computer vision that some of these diverse predictions have significantly lower task loss than that of a single model, and improve over state-of-the-art diversity encouraging approaches.

1 Introduction

The success of large margin methods for structured output learning, such as the structured support vector machine (SSVM) [1], is partly due to their good generalization performances achieved on test data, compared to, *e.g.* maximum likelihood learning on structured models [2]. Despite such regularization strategies, however, it is not guaranteed that the model which optimizes the learning objective function really generalizes well to unseen data. Reasons include wrong model assumptions, noisy data, ambiguities in the data, missing features, insufficient training data, or a task loss which is too complex to model directly.
To further decrease the generalization error, it is beneficial to either *(i)* generate

multiple likely solutions from the model [3, 4, 5] or, *(ii)* learn *multiple* models which generate diverse predictions [6, 7, 8]. The different predictions for a given structured input may then be analyzed to compute robustness/uncertainty measures, or may be the input for a more complex model exploiting higher-order dependencies, as is done in re-ranking models, *e.g.* Yadollahpour *et al.* [9] augment their features with global ones for automatic re-ranking. Other successful applications include prediction of diverse hypotheses for machine translation [10], on-demand feature computation [11], or active learning methods [12, 13]. Furthermore, an oracle may choose amongst all predictions that one which is closest to the ground truth. This becomes handy for proof-reading tasks in order to keep manual interactions at a minimum. It is particularly beneficial in structured output spaces to present to the user not only similarly likely, but also *diverse* proposal solutions. The set of diverse predictions may still contain a low-loss solution, even if the most likely prediction of the single model has a large loss. As a consequence, instead of minimizing the expected generalization error of a *single* model in structured learning, (cf. Fig. 1(a)), it is favorable to minimize the expected generalization error *amongst multiple* models, see Fig. 1(b,c).

Our main contribution is an algorithm termed the *Coulomb structured support vector machine* (CSSVM) which learns an ensemble of M models with different parameters, thanks to a corresponding diversity-encouraging prior. This is qualitatively different from previous work which requires that the *outputs* of the M models are diverse. In particular, we allow the M models in the ensemble to make identical predictions (and hence perfectly fit the data) at training time. Another benefit is that CSSVM can learn diverse models even if only a single structured training example is available. In Sec. 3.4, we generalize our algorithm to allow for structured clustering.

2 Related Work

One major research avenue is to generate at prediction time multiple (possibly diverse) solutions from a single previously trained structured model [3, 4, 5]. In order to find M similarly likely solutions, Yanover *et al.* [3] propose a message passing scheme to iteratively add constraints forbidding the previous solutions. Batra *et al.* [5] build on the same idea but incorporate these constraints directly into the objective function. This yields a deterministic framework which tries to find *diverse* solutions by requiring a minimum distance to the previous solutions. Their idea is extended in [14] to jointly infer diverse predictions at test time. Papandreou *et al.* [4], instead, perturb model parameters repeatedly with noise from a Gumbel distribution, and subsequently solve for the maximum-a-posteriori (MAP) solution to sample M plausible solutions. Their idea of perturbing the data term is natural when data is assumed to be noisy.
Sampling M solutions could of course also be achieved using Gibbs sampling or other MCMC techniques, however with very slow mixing time on general graphs; more efficient sampling strategies have been proposed recently [15]. Recent work aims at finding the M best modes of the probability distribution (local maxima)

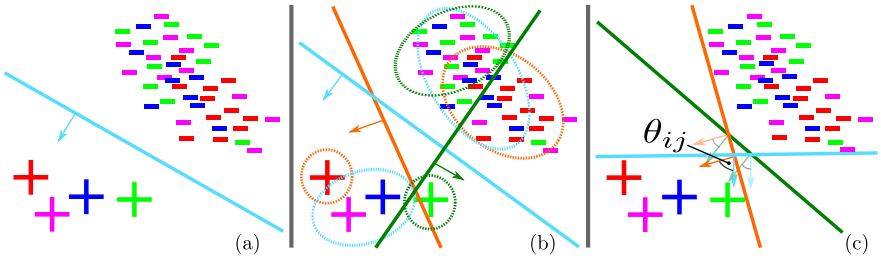

Figure 1: **Structured SVM learning.** "+" indicates a structured training example whereas "−" in the same color are the corresponding structured outputs with task loss $\Delta(+,-) > 0$. (a) A standard linear SSVM maximizes the margin between positive and all "negative" examples (decision boundary with its normal vector in cyan). (b) Multiple choice learning [6] learns M SSVMs (here: 3) which cluster the space (clusters for positive and negative examples are depicted in the same color) to generate M outputs. (c) We propose the Coulomb Structured SVM which learns an ensemble of M SSVMs through a diversity term which maximizes the pairwise angles θ_{ij} between their (linear) decision boundaries, while seeking to best fit all training data.

directly [16, 17]. While promising, their algorithms are yet not applicable to general graphs. Another recently discussed approach to sample diverse predictions at test time are determinantal point processes [18].

Rather than learning one model and then sampling successively (possibly diverse) solutions from the model, recent developments [6, 7, 8] allow to *train* multiple diverse models, *i.e.* diversity is already considered at training time. Typically, only one ground truth solution is provided per training sample rather than a diverse set, and thus diversity amongst the models can not be directly measured by means of training data. There are multiple works which tackle this challenge successfully: Gane *et al.* [8] learn (multi-modal) distributions over the perturbations in Perturb-and-MAP models using latent variable models which include inverse convex programs to determine relations between the model parameters and the MAP solution. Most similar to our work is [6, 7], where a set of M SSVMs is optimized while trading diversity versus data fit. In the former, diversity is encouraged through clustering: Each structured training example is assigned to the learner which achieves the lowest task loss for this sample in the current iteration. Their idea builds on the assumption that there are M clusters present in the training samples, thus requiring at least M (implicitly) diverse training samples. This requirement may be a crucial problem on small training sets. Our approach, in contrast, can learn M diverse models even if only one training example is present, as is often the case in CRF learning, *e.g.* co-segmentation (Sec. 4i), [19, 20]. In their more recent work, Guzman-Rivera *et al.* [7] extend their idea by augmenting the learning objective directly with a convex term which explicitly rewards diversity in the *outputs* of different learners, as also done in [21]. In our approach, in contrast, the diversity prior is

posed on the *parameters* of the M models, and thus, all learners might achieve the same loss on the training samples while still providing diverse predictions on test data, *cf.* Fig. 1(b,c).

3 Coulomb Structured Support Vector Machine

The goal of this work is to learn M mappings from one structured input to M possibly *diverse* structured outputs from a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,...,N}$.

3.1 Problem Description and Diversity Prior

For this purpose, we propose to learn an ensemble of M concurrent structured SVMs, which amounts to the following optimization problem:

$$\underset{\mathbf{w}_1,...,\mathbf{w}_M}{\arg\min} \underbrace{\alpha\Gamma(W)}_{\text{diversity}} + \underbrace{\Omega(W)}_{\text{generalization}} + \underbrace{C \cdot R_M(W, \mathcal{D})}_{\text{data term}}, \tag{1}$$

where $R_M(W, \mathcal{D}) = \frac{1}{MN} \cdot \sum_{m=1}^{M} \left(\sum_{i=1}^{N} L(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}_m) \right)$ is the empirical risk with $L(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}_m)$ being the structured loss of the i-th training example evaluated by the m-th learner. $\Omega(W)$ is the regularization term on the parameters $W = [\mathbf{w}_1, ..., \mathbf{w}_M]$ (in SSVMs typically an $L2$ regularizer is used on each single $\mathbf{w}_i$), and a bias term is omitted since it does not have an influence on the optimization problem [22]. Diversity amongst the M learners is encouraged by the diversity prior $\Gamma(W)$ on the parameters W, where α regulates the degree of diversity. In this way, $\alpha = 0$ reveals the standard SSVM formulation, since all M weights converge to the same optimum.

For the ease of argument, let us now assume the training set is linearly separable[1] as in Fig. 1. Moreover, assume that feature selection yielded independent features. Our illustration of the structured learning problem in Fig. 1 is analogous to representations of flat classification problems where we regard the ground truth labeling of the structured training samples as the single positive examples and all other (exponentially many) labelings as corresponding negative examples. The objective in Fig. 1(a) is to find a weight vector $\mathbf{w}$ which separates the positive from the negative examples and maximizes the margin [1].

We define the *version space* $V(\mathcal{D})$ analogously as in flat classification [24, 25], as

$$V(\mathcal{D}) = \{\mathbf{w} \in \mathcal{W} \mid R_1(\mathbf{w}, \mathcal{D}) = 0\}, \tag{2}$$

where R_1 is the empirical risk as in Eq. (1) with $M = 1$, and $\mathcal{W}$ is the space of feasible weight vectors. In other words, the version space is the set of all feasible weight vectors which yield zero loss on the training set $\mathcal{D}$. For linear classifiers, the weight vectors $\mathbf{w} \in \mathcal{W}$ are linear combinations of the training points $\mathbf{x}_i$ [25], *i.e.* $\mathbf{w} = \sum_{i=1}^{N} c_i \mathbf{x}_i$ for coefficients c_i, and the version space may be restricted

[1]Note that this is almost always true once we have a sufficient number of independent features, see the function counting theorem [23].

appropriately. Note that the error of a structured model induced from a weight vector in version space may still be large for randomly chosen query points (*i.e.* high generalization error) in spite of achieving zero loss on the training set.
Typically, version space is only summarized by a single point such as the center of the largest inscribed sphere (the hard-margin SVM) or the center of mass of the version space (the Bayes point machine [26]). To learn an ensemble of classifiers, our goal is to distribute M weight vectors $\mathbf{w}_m \in \mathcal{W}$, $m = 1, ..., M$, in version space such that the most diverse predictions on unseen points are obtained. To this end, it is sufficient for structured models with energy functions linear in $\mathbf{w}$ – similar to flat linear classification [27] – to only investigate weight vectors on the unit sphere (*i.e.* $\|\mathbf{w}\|_2 = 1$): At prediction time, labelings are scored by the energy function of the structured model $E(\mathbf{x}, \mathbf{y}) = \mathbf{w}^\top f(\mathbf{x}, \mathbf{y})$, where $f(\mathbf{x}, \mathbf{y})$ is the joint feature function. Replacing $\mathbf{w}$ by $\lambda\mathbf{w}$, $\lambda > 0$, still yields the same ordering of the labelings.
We hence have to solve an experimental design problem on parts of the unit sphere to get an ensemble of *diverse* structured models, in other words – disregarding training data – we want to evenly distribute M points on the unit sphere. The goal of experimental design [28, 29] is to select from a set of possible experiments / configurations / parameter settings the subset with greatest expected merit. In our case, the set of experiments to choose from is the sphere $\|\mathbf{w}\|_2 = 1$. In other words, rather than sample the sphere uniformly, we need to bias our experimental design towards parameters that produce low empirical loss. Hence, we next introduce the repulsive diversity energy term $\Gamma(W)$ which makes Eq. (1) a non-convex optimization problem, wich we optimize approximately.

3.2 Diversity through Coulomb Potential

Distributing M points evenly on the unit sphere is much studied in information theory and is known as a spherical code [30]: Different variants include *sphere packing* (maximize the minimal angle between any two parameter vectors) and *covering problems* (minimize the distance between any point on the sphere and the closest parameter vector). In three dimensions, the problem is known as the *Thomson problem*[2]: The goal is to minimize the energy configuration of M charges on a unit sphere while the charges repel each other with forces determined by Coulomb's law. While yet unsolved exactly, approximate solutions have been proposed in the literature, including spiral approximations [31], subdivisions of polyhedrons [32], or gradient descent methods [33, 34, 35] which correspond to electrostatic repulsion simulations exploiting Coulomb's law: Particles of equal charge repel each other with a force proportional to the square of their pairwise distance, the Coulomb force. More generally, in the equilibrium

[2]Note that we want to approximate this problem in a high dimensional space instead of only 3 dimensions.

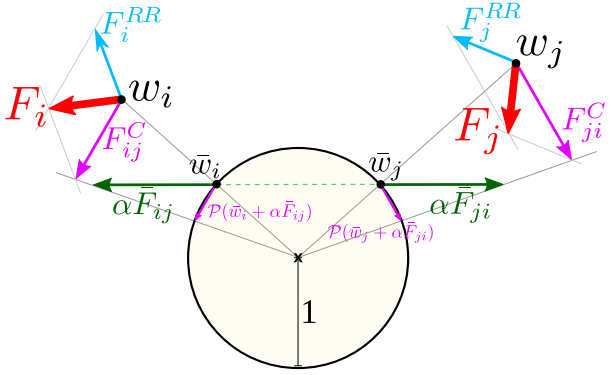

Figure 2: **Optimization** In each iteration of the subgradient algorithm, the current weights w of the competing M learners (here: 2) are projected to the unit sphere, $\bar{w}$, their Coulomb forces (green) are computed, and the resultant weight updates $\mathcal{P}(\bar{w} + \alpha\bar{F})$ are projected from the unit sphere to the original weight vectors w, yielding F^C (pink). Independently, the negative gradient of the regularized risk determines forces F^{RR} (blue). Added together, F^{RR} and F^C yield the update F of the weight vector (red).

state of the M particles $\mathbf{p}_1, ..., \mathbf{p}_M$ on the unit sphere, the Riesz energy,

$$E_s(\mathbf{p}_1, ..., \mathbf{p}_M) = \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} \frac{1}{\|\mathbf{p}_i - \mathbf{p}_j\|_2^s} \quad \text{s.t. } \|\mathbf{p}_i\|_2^2 = 1 \ \forall i \tag{3}$$

is minimal. In the following, we set $s = 1$ which yields the Coulomb energy $E_C = E_1$. The Coulomb force which affects particle $\mathbf{p}_i$ amounts to the negative gradient vector of Eq. (3) w.r.t. $\mathbf{p}_i$ [33, 36, 35] and is given by

$$\bar{F}_i^C = -\frac{\partial E_C}{\partial \mathbf{p}_i}(\mathbf{p}_1, ..., \mathbf{p}_M) = -\sum_{j=1, j\neq i}^{N} \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_i - \mathbf{p}_j\|_2^3} = \sum_{j=1, j\neq i}^{N} \frac{\mathbf{e}_{ij}}{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}, \tag{4}$$

where $\mathbf{e}_{ij}$ is the unit vector from $\mathbf{p}_i$ to $\mathbf{p}_j$. Projecting the resultant of force $\bar{F}_i^C$ on $\mathbf{p}_i$ back to the unit sphere by the projection $\mathcal{P}(\mathbf{p}) = \frac{\mathbf{p}}{\|\mathbf{p}\|}$ yields the projected gradient descent update on $\mathbf{p}_i$, namely $\mathbf{p}_i' = \mathcal{P}(\mathbf{p}_i + \bar{F}_i^C)$.

3.3 Optimization by an Electrostatic Repulsion Model

In the following, we will specify the diversity term $\Gamma(W)$ in Eq. (1) and minimize it by utilizing the electrostatic repulsion simulation from the previous section. As derived in Sec. 3.1, the magnitudes of vectors $\mathbf{w}_m$ do not contribute to the diversity term $\Gamma(W)$. Thus, we project the weight vectors to the unit sphere, *i.e.*

$\bar{\mathbf{w}}_m = \frac{\mathbf{w}_m}{\|\mathbf{w}_m\|}$, and use the Coulomb energy E_C as the diversity term[3] in Eq. (1),

$$\Gamma(\mathbf{w}_1, ...\mathbf{w}_M) = E_C(\bar{\mathbf{w}}_1, ..., \bar{\mathbf{w}}_M). \tag{5}$$

Note that the weights in both the regularizer $\Omega(W)$ and the risk $R_M(W, \mathcal{D})$ are *not* constrained to the unit sphere.

In Sec. (3.2), we derived the projected Coulomb forces which act on the point $\bar{\mathbf{w}}_m$ on the unit sphere. This update step can be projected to $\mathbf{w}_m$ utilizing the intercept theorem (*cf.* Fig. 2),

$$F_m^C = \|\mathbf{w}_m\|_2^2 \cdot \mathcal{P}(\bar{\mathbf{w}}_m + \alpha \bar{F}_m^C). \tag{6}$$

Next, let us derive force F_m^{RR} which acts on particle $\mathbf{w}_m$ according to the regularized risk $\Omega(W) + C \cdot R_M(W, \mathcal{D})$ in Eq. (1). The regularized risk in a structured SVM can be minimized using subgradient methods [38] and the negative subgradient for the learner m amounts to the force F_m^{RR}, *i.e.* the direction to go in the next optimization step when only considering the regularized risk. The $L2$ regularized risk of *one* learner is given by $R_1(\mathbf{w}_m, \mathcal{D}) = \frac{1}{2}\|\mathbf{w}_m\|_2^2 + \frac{C}{N}\sum_{k=1}^N L(\mathbf{x}_k, \mathbf{y}_k; \mathbf{w}_m)$. When choosing the structured hinge loss $L(\mathbf{x}_k, \mathbf{y}_k; \mathbf{w}_m) = \max_{\mathbf{y}\in\mathcal{Y}} \big(\Delta(\mathbf{y}_k, \mathbf{y}) - \mathbf{w}_m^\top f(\mathbf{x}_k, \mathbf{y})\big) + \mathbf{w}_m^\top f(\mathbf{x}_k, \mathbf{y}_k)$, where $\Delta(\mathbf{y}_k, \mathbf{y})$ is the task loss, f the feature function, and $(\mathbf{x}_k, \mathbf{y}_k)$ are the training examples; then the subgradient $\mathbf{g}_k^m$ for training example k is given by

$$\begin{aligned} \hat{\mathbf{y}}^m &= \max_{\mathbf{y}\in\mathcal{Y}} \big(\Delta(\mathbf{y}_k, \mathbf{y}) - \mathbf{w}_m^\top f(\mathbf{x}_k, \mathbf{y})\big) + \mathbf{w}_m^\top f(\mathbf{x}_k, \mathbf{y}_k), \\ \mathbf{g}_k^m &= f(\mathbf{x}_k, \mathbf{y}_k) - f(\mathbf{x}_k, \hat{\mathbf{y}}^m), \end{aligned} \tag{7}$$

i.e. the regularized risk force on particle $\mathbf{w}_m$ is $F_m^{RR} = -\frac{1}{N}\sum_{k=1}^N \mathbf{g}_k^m$.

Finally, all forces acting on $\mathbf{w}_m$ can be summed to the total force F_m which determines the next update of $\mathbf{w}_m$: $F_m = F_m^{RR} + F_m^C$. In other words, defining η_t as the step size at iteration t and $\mathbf{g}_l^m$ as in Eq. (7), then the update of $\mathbf{w}_m$ is given by

$$\mathbf{w}'_m \leftarrow \mathbf{w}_m - \eta_t\big(\mathbf{w}_m + \frac{C}{N}\sum_{k=1}^N \mathbf{g}_k^m - F_m^C\big), \quad \text{or:} \tag{8}$$

$$\mathbf{w}'_m \leftarrow \mathbf{w}_m - \eta_t\left(\mathbf{w}_m + C\mathbf{g}_l^m - F_m^C\right), \tag{9}$$

where the latter is the update in the *stochastic* subgradient algorithm with a random $l \in \{1, ..., N\}$. Note that element $[\mathbf{w}'_m]_i$ may be projected to zero to guarantee submodular energies during training as proved in [39]. For initialization of the CSSVM, we train one SSVM to get the optimum $\mathbf{w}_*$. Then M random perturbations of $\mathbf{w}_*$ give starting points for $\mathbf{w}_1, ...\mathbf{w}_M$.

[3]Note that we assume electrostatic charges on the *parameters*, and not the *training samples* as done in [37].

3.4 Extension: Structured Clustering

Our model suggests a straightforward extension to structured clustering: In the stochastic subgradient update given in Eq. (8), a random training sample is chosen for each learner to update the weight vector. Instead of random selection, a steered selection of training samples for each individual learner would increase diversity. Similarly to the structured K-means block-coordinate descent algorithm proposed in [6], we assign training examples to individual learners: After each subgradient iteration in Sec. 3.3, the task losses $\Delta(\mathbf{y}^m, \mathbf{y}_i; \mathbf{w}_m)$ between prediction $\mathbf{y}^m$ and ground truth $\mathbf{y}_i$ are computed for each learner m, $m \in \{1, ..., M\}$, and normalized over all learners, *i.e.* $\pi_i^m = \frac{\Delta(y^m, y_i; \mathbf{w}_m)}{\sum_{k=1}^{M} \Delta(y^k, y_i; \mathbf{w}_k)}$, $\sum_m \pi_i^m = 1$.

Training example i is then assigned to any of the M learners according to some indicator vector $\sigma(\boldsymbol{\pi}_i)$, where $[\sigma(\boldsymbol{\pi}_i)]_m = 1$ if training sample i is assigned to learner m, 0 otherwise. In Table 1, we propose different alternatives for the mapping $\sigma(\cdot)$. The subgradient update step in Eq. (8) is then modified accordingly:

$$\mathbf{w}'_m \leftarrow \mathbf{w}_m - \eta_t \big(\mathbf{w}_m + \frac{C}{\sum_{j=1}^{N} [\sigma(\boldsymbol{\pi}_j)]_m} \sum_{j=1}^{N} [\sigma(\boldsymbol{\pi}_j)]_m \cdot \mathbf{g}_j^m - F_m^C \big). \tag{10}$$

4 Experiments and Results

To evaluate the performance of our approach, we run experiments on three challenging tasks from computer vision: *(i)* co-segmentation, *(ii)* foreground/background segmentation, and *(iii)* semantic segmentation. We use the iCoseg [40] database for *(i)* and *(ii)* and PASCAL VOC 2010 [41] for *(iii)*. Note that for clearer comparison with previous work, we focus on the evaluation of our first stage model usually used in a two stage pipeline. The proposed method can be combined with any second stage model [10, 11, 42, 12, 13, 43, 44].

We implemented our algorithm in Python using the PyStruct [45] framework. The code is made available on `https://github.com/martinsch/coulomb_ssvm`. On all three tasks, we are comparing our results with the state-of-the-art diversity inducing methods Multiple Choice Learning [6] (MCL) and Diverse Multiple Choice Learning [7] (DivMCL), the Matlab implementations of which as well as their features/splitting criterions for the iCoseg dataset in task *(ii)* were kindly provided by the authors. The energies for tasks *(i)* and *(ii)* are submodular, and we thus use graph-cut as inference method; for the multi-label problem in *(iii)*, we utilize TRWS [46].

Generating M diverse outputs is particularly useful in early stages of cascaded approaches, where at a later stage, *e.g.* a human or a second complex model may choose the best of M predictions according to a higher-order loss function. The goal of our approach is, hence, to generate M diverse predictions some of which ought to achieve better task loss than the prediction of the single max-margin model. We therefore stick to the evaluation criterion as applied in

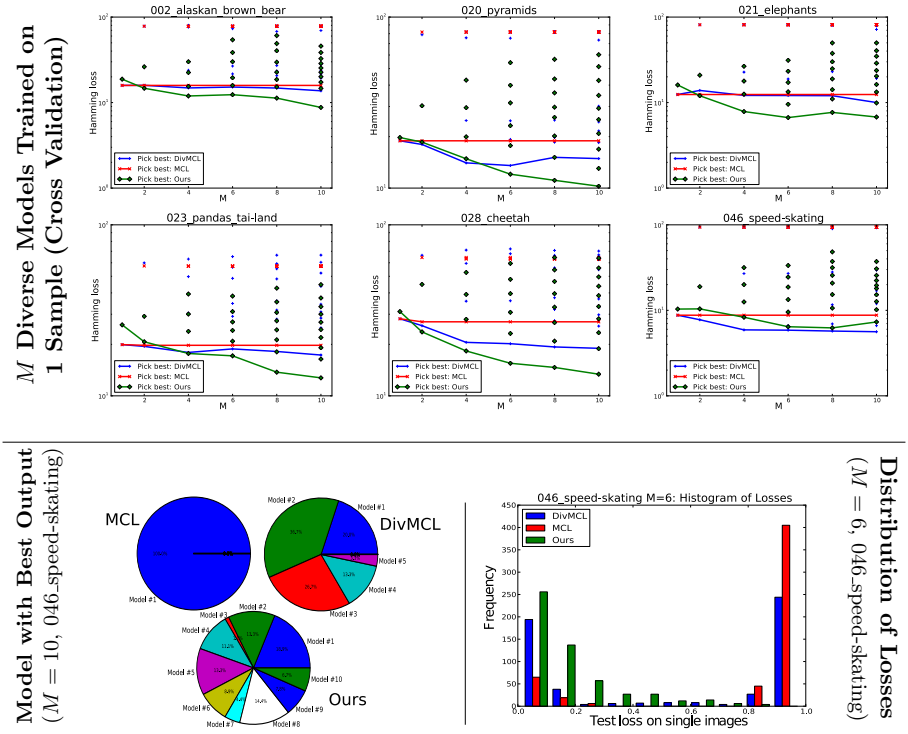

Figure 3: **Top:** Hamming losses on the respective datasets of the iCoseg database averaged after cross-validation (lower is better): Each fold consists of exactly one image. We train our model, MCL [6], and DivMCL [7] on one fold, validate on three other folds, and take the remaining $N_c - 4$ folds as test folds, the errors of which we report. For each test example, we compute the M task losses of the predictions to the ground truth, report the minimum as the pick best error (line), and mark the averages of the second, third, etc. best errors in the graphs. In other words, the line represents the losses which an oracle achieves when selecting always the best out of the M predictions. Note that the average error when always selecting the prediction with highest task error (*i.e.* the worst prediction), is constantly lower in our model than in the competing MCL and DivMCL. **Bottom left:** Frequency of how often model $\#i$, $i \in \{1, ..., M\}$, generates the best test prediction; here $M = 10$, speed-skating dataset. Note that in our algorithm, there is no dominant model and each of the M models achieves the pick-best error on a reasonable number of test samples, whereas in MCL and DivMCL the pick-best losses are attributed to only one or few models, respectively. **Bottom right:** Frequencies of task losses achieved among all test folds and models. All models in our CSSVM ensemble yield predominantly low losses whereas in Div-/MCL many predictions are useless.

prior works, where an oracle chooses the best out of M predictions. In this way, we can evaluate the usefulness of such an approach for cascade models. We relate to this loss as *pick best* error, *i.e.* the lowest task loss among the M predictions.

(i) Co-Segmentation The design of the proposed CSSVM allows to learn an ensemble of diverse models on very small training sets, in fact, even on training sets which consist of one structured training example only. To demonstrate the usefulness of our approach on such tasks, we run experiments on a co-segmentation dataset. The goal in co-segmentation in general is the simultaneous segmentation of two images each containing similar objects [47]. In our experiments, we assume that a model can be learned on the annotations of one image to predict the segmentation of similar images. We choose six categories from the iCoseg database and use the superpixels and features from [7], their 12-dim. color features for the nodes and a contrast-sensitive and -insensitive Potts term for the edges.

The results for MCL, DivMCL, and our model are depicted in Fig. 3. For each category, we vary the number of models in the ensemble M from 1 to 10, where $M = 1$ may be viewed as the baseline and corresponds to the training of a standard SSVM. We perform a full N_c-fold crossvalidation on each category, where N_c is the number of images in category c, and report the test losses of all M models. We choose the regularization and diversity trade-off parameters of each method on a hold-out validation set consisting of three images per category. Note that these losses are computed on superpixel level rather than pixel level which makes for a fair comparison since all three models are using the same superpixels and features. In these datasets, N_c's are in the range of 10 to 33, dependent on the dataset. Obviously, we use strategy "all" from Tab. 1 for these experiments.

It should be noted that, if we took the same implementations, exactly the same losses for all three competing models for $M = 1$ would be obtained (since all three models are direct generalizations of SSVM). The deviations here are probably due to different optimization strategies, *e.g.* different minima on a plateau or not enough iterations for the subgradient method (Div-/MCL use cutting-plane optimization instead).

On all six datasets, our method clearly improves over the baseline of only one SSVM ($M = 1$) and achieves better pick-best errors for large M than MCL and DivMCL do, with the exception of the *speed-skating* category. We show for this category exemplarily, however, that our algorithm learns M models which are all performing similarly well while in DivMCL only few models are strong, and in MCL, there exists only one strong model since diversity is only encouraged by assigning the training samples (here: 1) to specific models (shown for $M = 10$ in bottom left of Fig. 3). The phenomenon that our method yields significantly better average errors across all predictors in the ensemble is also reflected in the histogram of all losses from the full cross validation, as provided in Fig. 3 bottom right. The fact that most of the predictions achieve low loss in the proposed CSSVM is a strong advantage when the model is used in a cascade model since

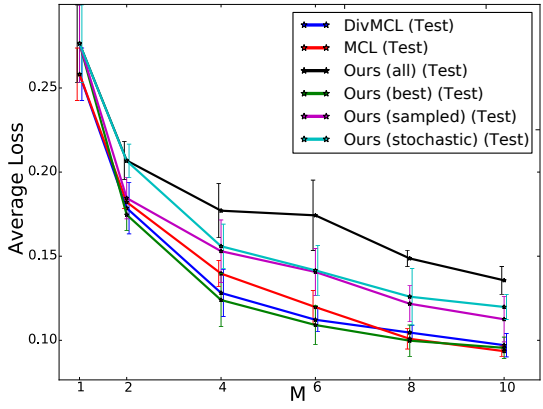

Figure 4: **Foreground/background Segmentation (iCoseg).** Average pick-best error (Hamming distance, lower is better) on the set of all categories. Shown are the test errors with one standard deviation (error bars are slightly perturbed on the x-axis for illustration purposes). Our training sample assignment strategies are denoted as in Tab. 1.

all predictions are good candidates to be selected as the best solution.
Example images for $M = 10$ are presented in Fig. 5. Note that for CSSVM, all models in the ensemble achieve similar training performances while yielding high diversity on the test images. By design, diversity on the training samples is *not* rewarded but models are distributed diversely in version space as argued in Sec. 3.1 in order to achieve a low generalization error on unseen data when the predictions of all M models are considered jointly. This is in contrast to the competing methods, where diversity among the models is also enforced on the training set.

(ii) Foreground/background Segmentation In this experiment, we use all these categories together (166 images in total) and use the same split criterion for the 5-fold cross validation as in [7]. We train the models on one fold, select regularization and diversity trade off parameters on two validation folds and report the test error on the remaining two folds. Fig. 4 presents the results for MCL, DivMCL, and our model with different sample assignment strategies as in Tab. 1. Since this dataset consists of different categories, it seems natural that the models which cluster the training data by assigning training instances to distinct models (as in Div-/MCL, Ours-sampled, and Ours-best) perform better than the models which try to fit all M models to the *entire* dataset (Ours-all, Ours-stochastic). Our model achieves similar accuracies as the state-of-the-art method DivMCL in this experiment.

$[\sigma(\boldsymbol{\pi}_i)]_m =$	**Description**	**Abbrev.**
1	Assign the sample i to every learner $m \in \{1, ..., M\}$, *i.e.* Eq. (8).	**all**
$\mathbb{1}\left[m = \arg\min_{m'}\{\pi_i^{m'}\}\right]$	Assign the sample i to the learner m which achieves the best task loss.	**best**
$\mathbb{1}\left[m = \hat{m}(\pi_i^1, ..., \pi_i^M)\right]$	Sample a learner index $\hat{m}$ from the distribution defined by $q_i^1, ...q_i^M$ and assign the sample i to learner $\hat{m}$; here, $q_i^m = \frac{1-\pi_i^m}{\sum_j (1-\pi_j^m)}$, $\sum_m q_i^m = 1$.	**sampled**
$\mathbb{1}\left[i = \hat{j}_m\right]$	Sample one training example index $\hat{j}_m \in \{1, ..., N\}$ for each learner $m \in \{1, ..., M\}$, *i.e.* Eq. (9).	**stochastic**

Table 1: Possible mappings for the assignment of training samples to individual learners

(iii) Semantic Segmentation We also evaluate our algorithm on the PASCAL VOC 2010 benchmark dataset for object class segmentation (challenge 5). The dataset consists of an official training set and validation set comprising 964 images each, which contain 21 object classes. We use the SLIC superpixels and Textonboost potentials [48] publicly available from [45]. Due to the lack of a publicly available test set, we are selecting the parameters of all three models on the official validation set and report these validation errors in Tab. 2 using the PASCAL VOC evaluation criterion, the Jaccard index. For structured learning, all models use a loss weighted by the inverse class frequency present in the training data. The baselines for this experiment are given by an arg max operation on our features ("unaries only"), a linear SVM on the unary features, and a structured SVM ($M = 1$). With these publicly available features, these baselines achieve average accuracies of 21.6%, 27.4%, and 29.1% which is much lower than the current best results reported on this challenge. In this experiment, however, we want to focus on how much a baseline algorithm can be improved thanks to a diverse ensemble, and not indulge in feature and pipeline tuning.

By training $M = 6$ diverse models and selecting the best predictions amongst them according to the ground truth, all three competing methods yield significantly higher pick-best accuracies than a single SSVM. We can even improve the accuracy from 29.1% to 37.6% with the assignment strategy "best" (*cf.* Tab. 1). This massive relative improvement underlines the usefulness of a diverse ensemble approach. MCL (35.0%) and DivMCL (34.5%) yield inferior performance.

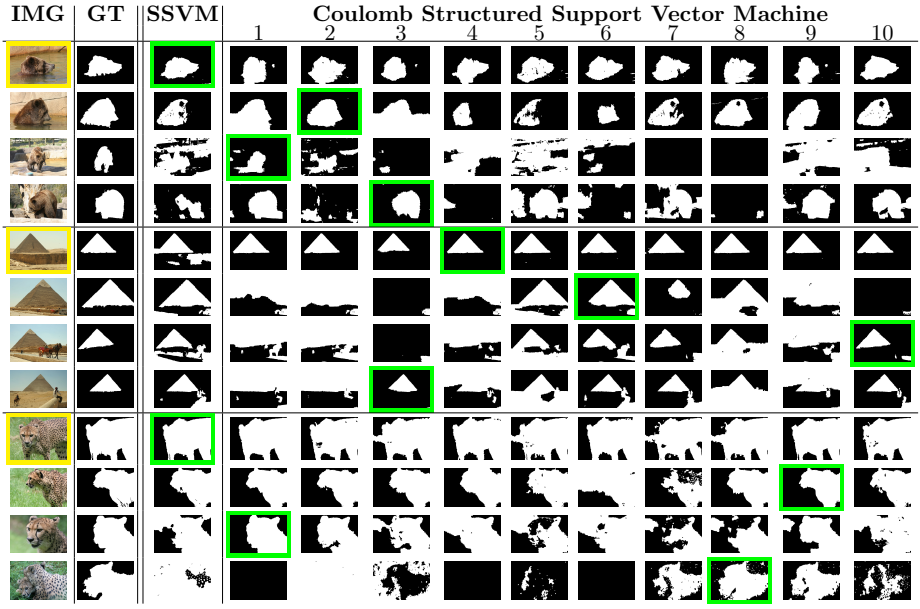

Figure 5: **Foreground/background Co-segmentation** (white/black, respectively). The single training image in each dataset is marked in yellow, the best prediction is framed in green. Note that all $M = 10$ models of CSSVM fit the training images similarly well, whereas high diversity amongst the M models is present in the predictions of the test set. GT stands for ground truth.

5 Conclusion

We propose an algorithm termed the *Coulomb structured support vector machine* which learns an ensemble of multiple models in order to yield *diverse* predictions on *test data*. The diversity prior is imposed on the set of model weights rather than on the outputs of training samples as in previous approaches. This allows for the training of diverse models even on a single structured training example. The CSSVM trades off diversity, large margins, and a data term during training in order to optimize the minimum expected generalization error of the *entire* ensemble. The coupling between the M models is effective only at training but not at test time. As a consequence, predictions can be made in parallel without communication overhead in contrast to [5]. Our algorithm learns multiple *strong* predictors in an ensemble on the entire dataset, other than [6, 7] where predictors 'focus' on the different clusters in the data, if present. We demonstrate on numerous real world datasets that the M diverse outputs of the proposed ensemble method include predictions with significantly lower task loss compared to only one model. Moreover, our approach of inducing diversity significantly improves over state-of-the-art methods on very small training sets while staying on par with the state-of-the-art methods on bigger training sets. The usefulness

Method	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	plant	sheep	sofa	train	tvmonitor	Average Accuracy
Unaries only	80.2	25.0	0.1	10.6	14.3	13.8	32.1	44.0	30.0	4.9	9.5	4.4	11.9	15.4	27.5	35.5	10.5	19.8	12.0	28.6	22.3	21.6
Linear SVM	80.0	36.6	2.8	17.3	23.0	25.6	40.4	48.7	27.6	8.3	19.5	10.5	13.3	21.9	34.4	36.4	16.9	22.8	17.0	37.0	34.3	27.4
SSVM (M=1)	79.9	39.9	2.1	18.5	27.5	28.4	43.2	49.2	28.7	8.4	21.6	12.3	14.1	23.7	35.2	37.2	22.0	23.6	18.3	38.9	39.4	29.1
MCL (M=6)	82.0	49.1	1.0	31.5	21.2	31.4	55.3	57.7	37.0	12.0	33.0	27.9	28.0	28.8	40.9	39.4	15.1	32.1	23.2	42.4	46.5	35.0
DivMCL (M=6)	82.2	30.3	0.5	25.7	26.4	30.4	51.1	56.3	42.7	7.9	33.5	22.9	45.4	27.3	45.6	43.3	21.3	39.5	17.4	42.9	32.2	34.5
Ours (M=6, all)	83.4	44.4	1.7	37.4	34.1	34.2	47.7	54.9	42.8	8.9	34.4	22.8	40.4	24.7	33.2	44.5	25.7	29.1	20.3	40.1	41.3	35.5
Ours (M=6, stochastic)	83.5	40.1	2.4	25.1	23.0	28.5	57.4	51.8	35.3	8.4	33.7	18.9	31.3	24.9	37.9	42.0	22.7	37.8	24.3	44.4	51.3	34.5
Ours (M=6, best)	83.2	48.8	3.2	38.3	28.4	33.3	58.1	60.3	51.1	7.7	34.5	21.6	34.6	32.0	39.3	43.4	17.7	27.7	26.6	48.3	51.3	**37.6**
Ours (M=6, sampled)	83.9	42.4	1.7	27.6	27.5	33.1	55.9	53.0	46.6	7.6	34.1	25.6	34.6	26.1	41.6	45.6	27.4	32.6	25.8	46.5	48.4	36.6

Table 2: **Pascal VOC 2010 Validation Accuracy** (higher is better). We tune a popular conditional random field [45] as baseline structured models (top rows). We here focus on the relative improvement that different diversity strategies can achieve (bottom rows), rather than tweaking the baseline model itself.

for machine learning tasks beyond computer vision is evident.

Acknowledgements

We would like to thank Abner Guzman-Rivera for making the (Div)MCL source code available.

References

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

[2] Nowozin, S., Lampert, C.H.: Structured learning and prediction in computer vision. Foundations and Trends in Computer Graphics and Vision **6**(3–4) (2011) 185–365

[3] Yanover, C., Weiss, Y.: Finding the m most probable configurations using loopy belief propagation. NIPS **16** (2003)

[4] Papandreou, G., Yuille, A.L.: Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In: ICCV. (2011)

[5] Batra, D., Yadollahpour, P., Guzman-Rivera, A., Shakhnarovich, G.: Diverse M-best solutions in Markov random fields. In: ECCV. (2012)

[6] Guzman-Rivera, A., Batra, D., Kohli, P.: Multiple choice learning: Learning to produce multiple structured outputs. In: NIPS. (2012) 1808–1816

[7] Guzman-Rivera, A., Kohli, P., Batra, D., Rutenbar, R.A.: Efficiently enforcing diversity in multi-output structured prediction. In: AISTATS. (2014)

[8] Gane, A., Hazan, T., Jaakkola, T.: Learning with maximum a-posteriori perturbation models. In: AISTATS. (2014) 247–256

[9] Yadollahpour, P., Batra, D., Shakhnarovich, G.: Discriminative re-ranking of diverse segmentations. In: CVPR. (2013)

[10] Gimpel, K., Batra, D., Dyer, C., Shakhnarovich, G.: A systematic exploration of diversity in machine translation. EMNLP (2013)

[11] Roig, G., Boix, X., de Nijs, R., Ramos, S., Kühnlenz, K., Van Gool, L.: Active MAP inference in CRFs for efficient semantic segmentation. In: ICCV. (2013)

[12] Maji, S., Hazan, T., Jaakkola, T.: Active boundary annotation using random map perturbations. In: AISTATS. (2014)

[13] Premachandran, V., Tarlow, D., Batra, D.: Empirical minimum bayes risk prediction: How to extract an extra few% performance from vision models with just three more parameters. In: CVPR. (2014)

[14] Kirillov, A., Savchynskyy, B., Schlesinger, D., Vetrov, D., Rother, C.: Inferring m-best diverse labelings in a single one. In: ICCV. (2015) 1814–1822

[15] Hazan, T., Maji, S., Jaakkola, T.: On sampling from the Gibbs distribution with random maximum a-posteriori perturbations. In: NIPS. (2013) 1268–1276

[16] Chen, C., Kolmogorov, V., Zhu, Y., Metaxas, D., Lampert, C.: Computing the M most probable modes of a graphical model. In: AISTATS. (2013) 161–169

[17] Chen, C., Liu, H., Metaxas, D., Zhao, T.: Mode estimation for high dimensional discrete tree graphical models. In: NIPS. (2014) 1323–1331

[18] Kulesza, A., Taskar, B.: Determinantal point processes for machine learning. arXiv preprint arXiv:1207.6083 (2012)

[19] Lucchi, A., Li, Y., Smith, K., Fua, P.: Structured image segmentation using kernelized features. In: ECCV. (2012) 400–413

[20] Lou, X., Hamprecht, F.A.: Structured Learning for Cell Tracking. NIPS (2011)

[21] Li, Y.F., Zhou, Z.H.: Towards making unlabeled data never hurt. IEEE Trans. on PAMI **37**(1) (2015) 175–188

[22] Lampert, C.H.: Maximum margin multi-label structured prediction. In: NIPS. (2011) 289–297

[23] Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. Electronic Computers, IEEE Transactions on **EC-14**(3) (1965) 326–334

[24] Mitchell, T.M.: Machine Learning. 1 edn. McGraw-Hill, Inc., New York, NY, USA (1997)

[25] Herbrich, R., Graepel, T., Williamson, R.C.: The structure of version space. Technical Report MSR-TR-2004-63, Microsoft Research (July 2004)

[26] Herbrich, R., Graepel, T., Campbell, C.: Bayes point machines. JMLR **1** (September 2001)

[27] Graepel, T., Herbrich, R.: The kernel gibbs sampler. In: NIPS. (2001) 514–520

[28] Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Statistical science (1989) 409–423

[29] Hardin, R., Sloane, N.: A new approach to the construction of optimal designs. Journal of statistical planning and inference **37**(3) (1993) 339–369

[30] Conway, J.H., Sloane, N.J.A.: Sphere-packings, Lattices, and Groups. Springer (1987)

[31] Saff, E.B., Kuijlaars, A.B.: Distributing many points on a sphere. The Mathematical Intelligencer **19**(1) (1997) 5–11

[32] Katanforoush, A., Shahshahani, M.: Distributing points on the sphere, i. Experimental Mathematics **12**(2) (2003) 199–209

[33] Claxton, T., Benson, G.: Stereochemistry and seven coordination. Canadian Journal of Chemistry **44**(2) (1966) 157–163

[34] Erber, T., Hockney, G.: Equilibrium configurations of n equal charges on a sphere. Journal of Physics A: Mathematical and General **24**(23) (1991) L1369

[35] Lakhbab, H., Bernoussi, S.E., Harif, A.E.: Energy minimization of point charges on a sphere with a spectral projected gradient method. Internat. Journal of Scientific & Engineering Research **3** (2012)

[36] Neubauer, Schilling, Watkins, Zeitlin: An algorithm for finding potential minimizing configurations of points on a sphere. (1998)

[37] Hochreiter, S., Mozer, M.C., Obermayer, K.: Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems. In: NIPS. (2003) 561–568

[38] Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: (online) subgradient methods for structured prediction. AISTATS (2007)

[39] Prasad, A., Jegelka, S., Batra, D.: Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. In: NIPS. (2014) 2645–2653

[40] Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR. (2010)

[41] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2010 (2010)

[42] Tsai, Y.H., Yang, J., Yang, M.H.: Decomposed learning for joint object segmentation and categorization, BMVC (2013)

[43] Lee, T., Fidler, S., Dickinson, S.: Learning to combine mid-level cues for object proposal generation. In: CVPR. (2015) 1680–1688

[44] Wang, S., Fidler, S., Urtasun, R.: Lost shopping! monocular localization in large indoor spaces. In: ICCV. (2015)

[45] Müller, A.C., Behnke, S.: PyStruct - learning structured prediction in python. JMLR (2014)

[46] Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. on PAMI **28**(10) (2006) 1568–1583

[47] Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In: CVPR. (2006) 993–1000

[48] Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: NIPS. (2011)