In the typical setting for SVM's, we aim to optimize a linear objective over a set of constraints. There are many alernatives to solve that problem, the most popular ones being gradient-based methods. One of the issues with this type of algorithm is that they cannot be applied to matchings and min-cuts. In addition, variants that aim at generating constraints on the fly and incrementally solving new QP's do not typically scale well. The method proposed in this paper, which is in essence a modification of the extra-gradient method by Nesterov, leverages the min-max formulation and takes the dua ; the exponential number of constraints in then transformed an exponential number of variables. However, the paper highlights that we can use only a tractable number of these dual variables to solve the optimization problem. The method prososed in this paper also generalizes the applicability of the usual extra-gradient method to non-euclidean, Bergman projections. We thus end up with a framework that is more flexible and also efficient to implememt.

Since estimating the maximum likelihood over graphical models is often impractical or infeasible for a wide class of problems, we focus our attention on large margin estimation. For a dataset $S = \{(x_i, y_i)\}_{i=1}^m$, where each x_i is an object with a structure (e.g. sequence of words in french), we attempt to find the optimal parameter w of a linear classifier:

$$y_i = \arg\max_{y_i' \in \mathcal{Y}} w^T f(x_i, y_i) \tag{1}$$

The function f gives a feature mapping of a structured object with its corresponding label y_i. The error of a prediction is mesured by a loss function l. To make the loss convex, another term is introduced in the form a hinge loss. Since this gives an upper bound to the loss, it is natural to minimize it. We end up with a problem of the form:

$$\min_{w \in \mathcal{W}} \sum_i \max_{y_i' \in \mathcal{Y}_i} \left[w^T f_i(y_i') + l_i(y_i') \right] - w^T f_i(y_i) \tag{2}$$

The parameters w are also regularized with parameter λ. Since we are optimizing over y_i', we can drop the term from equation 1 and we end up with a loss-augmented inference problem inside the min function. The three types of structure that are presented in the paper have a general formulation that can better be expressed as:

$$\min_{w \in \mathcal{W}} \max_{z \in \mathcal{Z}} \sum_i \left(w^T F_i z_i + c_i^T z_i - w^T f_i(y_i) \right) \tag{3}$$

where the z_i's can be identified with the edge and node potentials of a markov network and satisfy the constraints of the structured problem. The terms F_i correspong to the feature mapping for over all labels y_i when multiplied by z_i's. The c_i's correspond to the costs of a z_i and can be identified with the loss l for a label y_i'. Taking the dual, we end up with the following:

$$\begin{aligned} &\min_{w \in \mathcal{W}, (\lambda, \mu) \geq 0} && \sum_i \left(b_i^T \lambda_i + \mathbf{1}^T \mu_i - w^T f_i(y_i) \right) \\ &\text{s.t.} && F_i^T w + c_i \leq A_i^T \lambda_i + \mu_i \quad i = 1, \dots, m \end{aligned} \tag{4}$$

The number of variables and constraints is linear in the number of paramters and training data. We already see that this formulation is much more efficient. We do have a set of constraints that is tractable, as is the number of parameters to update. In equation 3, the term that is opitmized is defined as:

$$\mathcal{L}(w, z) \triangleq \sum_i w^T F_i z_i + c_i^T - w^T f_i(y_i) \tag{5}$$

It is bilinear in w and z. We can then imagine two players represented by w and z that play a zero-sum game. They perform updates using gradients of the objective w.r.t. their parameters. They then project the result to the set of feasible points given by the constraints imposed on the structure. We usually consider Euclidean projections, as there are well-known problems where they are efficient to compute. However, as seen later, this is not the case for all problem. This is why Bergman projections will be introduced. Going back to the zero-sum game, we have the following operator that is used to perform the updates for both players at the same time.

$$\begin{pmatrix} \nabla_w \mathcal{L}(w,z) \\ -\nabla_{z_1} \mathcal{L}(w,z) \\ \vdots \\ -\nabla_{z_m} \mathcal{L}(w,z) \end{pmatrix} = \begin{pmatrix} 0 & F_1 & \dots & F_m \\ -F_1^T & & & \\ \vdots & & 0 & \\ -F_m^T & & & \end{pmatrix} \begin{pmatrix} w \\ z_1 \\ \vdots \\ z_m \end{pmatrix} - \begin{pmatrix} \sum_i f_i(y_i) \\ c_1 \\ \vdots \\ c_m \end{pmatrix} = Fu - a \tag{6}$$