Structured SVMs

Karl Stratos

This note closely follows Schmidt (2009) and Nowozin and Lampert (2011).

1 How (multi-class) SVMs are derived from log-linear models

Let X be our input space (can be anything) and let m be the number of possible labels. Assume a feature function $\phi : X \times [m] \to \mathbb{R}^d$ that maps an input-label pair to a vector.

1.1 A typical log-linear model

Consider a log-linear model. It has a parameter $w \in \mathbb{R}^d$ that defines a distribution over m labels for a given input $x \in X$ as follows:

$$p_w(y|x) := \frac{\exp(w^\top \phi(x, y))}{\sum_{y'=1}^{m} \exp(w^\top \phi(x, y'))} \quad \forall y \in [m]$$

Suppose we have M training samples $(x^{(1)}, y^{(1)}) \dots (x^{(M)}, y^{(M)}) \in X \times [m]$. A log-linear model is typically trained to maximize the log-likelihood of the data. That is, we calculate $w^* \in \mathbb{R}^d$ such that

$$w^* = \arg\max_{w \in \mathbb{R}^d} \sum_{i=1}^{M} \log p_w(y^{(i)}|x^{(i)}) - \frac{\lambda}{2} ||w||^2$$

where we use an l_2 regularizer to prevent overfitting and to ensure the uniqueness of w^*. The hyperparameter $\lambda \in \mathbb{R}$ constrols the strength of regularization.

1.2 A support-vector machine (SVM)

Now we consider a different training method. Instead of maximizing the probability of the samples under the model, we attack classification errors more directly (that's what we care about anyway). Define $I := \{(i, y) : i \in [M], y \in [m], y \neq y^{(i)}\}$, a set of $M(m-1)$ pairs. We seek the following $w^* \in \mathbb{R}^d$:

$$\begin{aligned} w^* &= \arg\min_{w \in \mathbb{R}^d} 0 \\ \text{s.t. for some } c > 1,\ & \frac{p_w(y^{(i)}|x^{(i)})}{p_w(y|x^{(i)})} \geq c \quad \forall (i, y) \in I \end{aligned} \tag{1}$$

Taking log on both sides of the constraints and defining $s_w(x, y) = w^\top \phi(x, y)$, we can rewrite this problem as:

$$\begin{aligned} w^* &= \arg\min_{w \in \mathbb{R}^d} 0 \\ \text{s.t. for some } c > 1,\ & s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq \log c \quad \forall (i, y) \in I \end{aligned}$$

Here, we see that the choice of $c > 1$ is arbitrary. If there exists *some* $c > 1$ that allows for some w satisfying the constraints, then *for all* $c > 1$ there is some w satisfying the constraints (simply by rescaling). So let's set c so that $\log c = 1$. The following is thus an equivalent problem:

$$w^* = \arg\min_{w \in \mathbb{R}^d} 0 \tag{2}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq 1 \quad \forall (i, y) \in I$$

The solution of problem (2) may not be unique, so we can use an l_2 regularizer to have the reformulation:

$$w^* = \arg\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} ||w||^2 \tag{3}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq 1 \quad \forall (i, y) \in I$$

where again the hyperparameter $\lambda \in \mathbb{R}$ controls the strength of regularization. However, a solution that satisfies the constraints might not exist. We can introduce slack variables $\xi_{(i,y)} \in \mathbb{R}$ for each $(i, y) \in I$ to ensure the feasibility of the problem:

$$w^*, \{\xi^*_{(i,y)}\}_{(i,y)\in I} = \arg\min_{w \in \mathbb{R}^d, \{\xi_{(i,y)}\}_{(i,y)\in I}} \frac{\lambda}{2} ||w||^2 + \sum_{(i,y)\in I} \xi_{(i,y)} \tag{4}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq 1 - \xi_{(i,y)} \quad \forall (i, y) \in I$$
$$\xi_{(i,y)} \geq 0 \quad \forall (i, y) \in I$$

This is known as the primal form of a soft-margin SVM. This is a constrained quadratic programming problem with $d + M(m-1)$ variables and $2M(m-1)$ linear constraints. An unconstrained formulation can be derived by observing that the constraints can be rearranged as $(\xi_{(i,y)} \geq 0) \wedge (\xi_{(i,y)} \geq 1 + s_w(x^{(i)}, y) - s_w(x^{(i)}, y^{(i)}))$, or

$$\xi_{(i,y)} \geq \max\{0, 1 + s_w(x^{(i)}, y) - s_w(x^{(i)}, y^{(i)})\}$$

for all $(i, y) \in I$. Since we are minimizing each $\xi_{(i,y)}$, we can bake these constraints into the objective, resulting in:

$$w^* = \arg\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} ||w||^2 + \sum_{(i,y)\in I} \max\{0, 1 + s_w(x^{(i)}, y) - s_w(x^{(i)}, y^{(i)})\} \tag{5}$$

This is an unconstrained, non-differentiable (but still convex) problem with d variables that is equivalent to (4). But there are many terms in the objective: the second summation is over $M(m-1)$ values.

1.3 Preparing SVMs for cases where m is large

With our formulation (1), the size of the problem grows linearly with the number of labels m. This is problematic when m is large. A slight modification allows us to dodge this problem entirely:

$$w^* = \arg\min_{w \in \mathbb{R}^d} 0 \tag{6}$$
$$\text{s.t. for some } c > 1, \frac{p_w(y^{(i)}|x^{(i)})}{\max_{y\in[m], y\neq y^{(i)}} p_w(y|x^{(i)})} \geq c \quad \forall i \in [M]$$

In other words, we are still attacking classification errors directly by ensuring that the true label is picked over all other alternatives, but we aren't bothering to compare with every alternative—only the best (wrong) alternative. For each $i \in [M]$, define

$$\tilde{y}^{(i)} = \underset{y \in [m], y \neq y^{(i)}}{\arg\max}\ p_w(y|x^{(i)})$$

The final soft-margin formulation in this "lazy" formulation, analogous to (4), is:

$$w^*, \{\xi_i^*\}_{i\in[M]} = \underset{w\in\mathbb{R}^d, \{\xi_i\}_{i\in[M]}}{\arg\min} \frac{\lambda}{2}||w||^2 + \sum_{i\in[M]} \xi_i \tag{7}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, \tilde{y}^{(i)}) \geq 1 - \xi_i \quad \forall i \in [M]$$
$$\xi_i \geq 0 \quad \forall i \in [M]$$

This is a constrained quadratic programming problem with $d+M$ variables and $2M$ linear constraints:[1] note the independence from m. An unconstrained version can be similarly derived, as

$$w^* = \underset{w\in\mathbb{R}^d}{\arg\min} \frac{\lambda}{2}||w||^2 + \sum_{i\in[M]} \max\{0, 1 + s_w(x^{(i)}, \tilde{y}^{(i)}) - s_w(x^{(i)}, y^{(i)})\} \tag{9}$$

2 SVMs for structured labels

Now we consider a case in which our labels are structured. For instance, a label is a possible binary tree over a sequence of words. Here, the labels (synonymous to structures) depend on the input x, so we write $Y(x)$ to denote the space of possible labels for the given input x.

We assume a convex scoring function $s_w(x, y) \in \mathbb{R}$ governed by model parameter $w \in \mathbb{R}^d$ that maps an input $x \in X$ paired with a label $y \in Y(x)$ to a real value. We can follow a similar derivation shown in section 1.2. We want to compute w^* that will let the model pick the correct label over incorrect ones, on every sample we have:

$$w^* = \underset{w\in\mathbb{R}^d}{\arg\min} 0 \tag{10}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq 1 \quad \forall y \in Y(x^{(i)}), y \neq y^{(i)}, i \in [M]$$

But now that labels are structured, they are not independent. Some incorrect labels are "more incorrect" than other incorrect labels. To capture the interdependence of labels, we introduce a *structured loss* function $\triangle(y, y') \in \mathbb{R}$ that measures the difference between labels y and y' (the higher, the more different). We also require that $\triangle(y, y) = 0$ and

[1] Notice that the formulation (7) is actually equivalent to the problem:

$$w^*, \{\xi_i^*\}_{i\in[M]} = \underset{\substack{w\in\mathbb{R}^d,\\ \{\xi_i\}_{i\in[M]}}}{\arg\min} \frac{\lambda}{2}||w||^2 + \sum_{i\in[M]} \xi_i \tag{8}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq 1 - \xi_i \quad \forall (i, y) \in I$$
$$\xi_i \geq 0 \quad \forall i \in [M]$$

which is the same as (4) but with the difference that there is only one slack variable ξ_i per sample $i \in [M]$.

$\triangle(y, y') > 0$ for $y \neq y'$. Now the problem can be written as:

$$w^* = \arg\min_{w \in \mathbb{R}^d} 0 \tag{11}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq \triangle(y^{(i)}, y) \quad \forall y \in Y(x^{(i)}), i \in [M]$$

The space of labels $Y(x^{(i)})$ is large—too many constraints! Let us prefer the "lazy" formulation in section 1.3:

$$w^* = \arg\min_{w \in \mathbb{R}^d} 0 \tag{12}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, \hat{y}^{(i)}) \geq \triangle(y^{(i)}, \hat{y}^{(i)}) \quad \forall i \in [M]$$

where

$$\hat{y}^{(i)} = \arg\max_{y \in Y(x^{(i)})} \; s_w(x^{(i)}, y) + \triangle(y^{(i)}, y) \tag{13}$$

It turns out that (13) can be computed efficiently using dynamic programming, under suitable definitions of $s_w(x, y)$ and $\triangle(y, y')$ (they must factor into local parts). Following the same steps in section 1.2, we arrive in the primal form of a "lazy" soft-margin structured SVM:

$$w^*, \{\xi_i^*\}_{i \in [M]} = \arg\min_{w \in \mathbb{R}^d, \{\xi_i\}_{i \in [M]}} \frac{\lambda}{2} ||w||^2 + \sum_{i \in [M]} \xi_i \tag{14}$$
$$s_w(x^{(i)}, y^{(i)}) - s_w(x^{(i)}, y) \geq \triangle(y^{(i)}, y) - \xi_i \quad \forall y \in Y(x^{(i)}), i \in [M]$$
$$\xi_i \geq 0 \quad \forall i \in [M]$$

The unconstrained version of (14) is thus

$$w^* = \arg\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} ||w||^2 + \sum_{i \in [M]} \max\{0, \triangle(y^{(i)}, \hat{y}^{(i)}) + s_w(x^{(i)}, \hat{y}^{(i)}) - s_w(x^{(i)}, y^{(i)})\} \tag{15}$$

Conveniently, because $\hat{y}^{(i)}$ is the maximizing $y \in Y(x)$ of $s_w(x^{(i)}, y) + \triangle(y^{(i)}, y) \geq s_w(x^{(i)}, y)$, we can discard the max operation and obtain

$$w^* = \arg\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} ||w||^2 + \sum_{i \in [M]} \triangle(y^{(i)}, \hat{y}^{(i)}) + s_w(x^{(i)}, \hat{y}^{(i)}) - s_w(x^{(i)}, y^{(i)}) \tag{16}$$

2.1 Subgradient-based training

We optimize the objective function in (16):

$$f(w) = \frac{\lambda}{2} ||w||^2 + \sum_{i \in [M]} \triangle(y^{(i)}, \hat{y}^{(i)}) + s_w(x^{(i)}, \hat{y}^{(i)}) - s_w(x^{(i)}, y^{(i)}) \tag{17}$$

A subgradient of $f(w)$ is

$$g(w) = \lambda w + \sum_{i \in [M]} s'_w(x^{(i)}, \hat{y}^{(i)}) - s'_w(x^{(i)}, y^{(i)}) \tag{18}$$

where $s'_w(x,y) = \frac{\partial}{\partial w} s_w(x,y)$. For instance, if $s_w(x,y) = w^\top \Phi(x,y)$, then $s'_w(x,y) = \Phi(x,y)$. This immediately gives a way to a stochastic online subgradient descent algorithm, such as the one below:

Input: training samples $(x^{(1)}, y^{(1)}) \dots (x^{(M)}, y^{(M)})$
Output: model parameter $w \in \mathbb{R}^d$

1. Initialize $w^0 \leftarrow 0 \in \mathbb{R}^d$ and $t \leftarrow 0$.
2. Loop:
   (a) Randomly pick $i \in [M]$:
      i. $w^t \leftarrow w^{t-1} - \eta^t g_i(w^{t-1})$ where η^t is an appropriate step size and
      $$g_i(w) = \lambda w + s'_w(x^{(i)}, \hat{y}^{(i)}) - s'_w(x^{(i)}, y^{(i)})$$
      ii. $t \leftarrow t + 1$
3. Return $w \leftarrow w^t$.

In this online version, we take a subgradient at each example i. We recover the structured perceptron algorithm of Collins (2002) if

$$\begin{aligned} s_w(x,y) &:= w^\top \Phi(x,y) \\ \lambda &:= 0 \\ \eta^t &:= 1 \ \ \forall t \\ \triangle(y,y') &:= 0 \ \ \forall y, y' \end{aligned}$$

and we set the return value w to be an average of w_t for all t. The update in step 2(a)i is

$$w^t \leftarrow w^{t-1} + \Phi(x^{(i)}, y^{(i)}) - \arg\max_{y \in Y(x^{(i)})} \Phi(x^{(i)}, y)$$

Thus the structured perceptron of Collins (2002) is simply a stochastic online gradient descent method for learning a degenerate structured SVM with no regularization (though parameter averaging somewhat compensates for the lack of regularization) and no structural loss between structures.

3 Motivating structured SVMs from a risk minimization point of view

In (12), we motivated structured SVMs from an angle to achieve the following goal: attack the classification errors but take into account structural penalties. They can also be motivated from a risk minimization point of view. This view places the structured loss $\triangle(y,y')$ at the center of the stage.

Let $f_w(x) := \arg\max_{y \in Y(x)} s_w(x,y)$ denote the model prediction. We seek to minimize the Bayes risk over the distribution of samples (x,y),

$$\mathbf{E}_{(x,y)}\big[\triangle(y, f_w(x))\big]$$

or more realistically, the regularized empirical risk on M actual samples

$$\frac{\lambda}{2}||w||^2 + \frac{1}{M}\sum_{i=1}^{M} \triangle(y^{(i)}, f_w(x^{(i)})) \tag{19}$$

But it's unclear how to optimize this objective (19) with respect to w because for a given $x \in X$ and $y \in Y(x)$, $\triangle(y, f_w(x))$ is discontinuous in w (looks like steps). Instead, we tighten a convex upper bound on $\triangle(y, f_w(x))$. In the context of structured SVMs, this bound can take the form of a *hinge loss*:

$$\text{hinge}(x, y; w) := \max_{y' \in Y(x)} \triangle(y, y') + s_w(x, y') - s_w(x, y) \tag{20}$$

This is

1. Continuous
2. Convex: max of convex functions
3. Upper bound on $\triangle(y, f_w(x))$:

$$\begin{aligned}\triangle(y, f_w(x)) &\leq \triangle(y, f_w(x)) + \underbrace{s_w(x, f_w(x)) - s_w(x, y)}_{\geq 0} \text{ since } f_w(x) := \arg\max_{y \in Y(x)} s_w(x, y) \\ &\leq \max_{y' \in Y(x)} \triangle(y, y') + s_w(x, y') - s_w(x, y)\end{aligned}$$

So we minimize the following convex upper bound to (19):

$$\frac{\lambda}{2}||w||^2 + \frac{1}{M}\sum_{i=1}^{M} \max_{y \in Y(x^{(i)})} \triangle(y^{(i)}, y) + s_w(x^{(i)}, y) - s_w(x^{(i)}, y^{(i)}) \tag{21}$$

The solution of (21) is the same as the solution of (16).

4 Appendix: motivating classic binary SVMs from a margin maximization point of view

In the case of binary classification, SVMs can be motivated through a geometric argument of finding a hyperplane that maximally separates the training data.

Given the training data $(x^{(1)}, y^{(1)}) \ldots (x^{(1)}, y^{(1)}) \in \mathbb{R}^d \times \{-1, +1\}$, our goal is to find $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

$$w^\top x^{(i)} + b = \begin{cases} > 0 & \text{if } y^{(i)} = +1 \\ < 0 & \text{if } y^{(i)} = -1 \end{cases} \tag{22}$$

for $i = 1 \ldots M$. Because of our label encoding scheme, we can write (22) compactly as

$$y^{(i)}(w^\top x^{(i)} + b) > 0 \;\; \forall i = 1 \ldots M \tag{23}$$

If you assume $d = 1$, the equation $w^\top x + b = 0$ can be visualized as a line on the plane. Our goal is to find such a line that perfectly separtes the two classes for the given training data. For the sake of argument, let's assume for now that the data is separable.

Here is a central step for the max margin derivation: *scale* the training data so that

$$y^{(i)}(w^\top x^{(i)} + b) \geq 1 \;\; \forall i = 1 \dots M \tag{24}$$

Assuming we have found w and b such that (23) holds, we can surely find w and b such that (24) holds.

Visually speaking, (24) gives a hyperplane $w^\top x + b = 0$ sandwiched between two hyperplanes $w^\top x + b = -1$ and $w^\top x + b = 1$ such that there is no $x^{(i)}$ lying between the sandwiching hyperplanes. The idea is to maximize this "sandwich void", in hope that this gives a separating hyperplane (among many possible separating hyperplanes, assuming one exists) that has the best generalization properties.

How do we implement this idea? We first need to express the "sandwich void" more concretely. We will express it as the closest distance between $w^\top x+b = -1$ and $w^\top x+b = 1$. Let x_1 be a point such that $w^\top x_1 + b = -1$. Since these hyperplanes are parallel and w has a direction orthogonal to both, the closest point x_2 such that $w^\top x_2 + b = 1$ can be expressed as

$$x_2 = x_1 + \beta w$$

where $\beta \in \mathbb{R}$ corresponds to the "sandwich void". Using the fact the points are on the hyperplanes, we have $\beta w^\top w = 2$ or $\beta = 2/w^\top w$. Thus we want to maximize $2/w^\top w$ under the constraints (24). Equivalently, we want to minimize $w^\top w/2$ under the constraints (24):

$$\begin{aligned} w^* &= \arg\min_{w \in \mathbb{R}^d} \frac{1}{2} ||w||^2 \\ y^{(i)}(w^\top x^{(i)} + b) &\geq 1 \;\; \forall i = 1 \dots M \end{aligned} \tag{25}$$

This is an instance of the (hard) SVM objective in (3) for the case of two labels (without the regularization parameter λ). Extending this formulation with slack variables is also straightforward:

$$\begin{aligned} w^*, \{\xi_i^*\}_{i \in [M]} = \arg\min_{w \in \mathbb{R}^d, \{\xi_i\}_{i \in [M]}} & \frac{1}{2} ||w||^2 + \sum_{i \in [M]} \xi_i \\ y^{(i)}(w^\top x^{(i)} + b) &\geq 1 - \xi_i \qquad \forall i = 1 \dots M \\ \xi_i &\geq 0 \qquad \forall i \in [M] \end{aligned} \tag{26}$$

which is an instance of the soft SVM objective in (4).

References

Nowozin, S. and Lampert, C. H. (2011). *Structured learning and prediction in computer vision*, volume 6. Now Publishers Inc.

Schmidt, M. (2009). A note on structural extensions of svms.