

Linear models for classification (cont'd)

Perceptron is an example of an error-correction algorithm:

1. Initialize $\theta = 0$
2. Iterate over training data (x, y) :
 - (a) If $y\theta^\top x \leq 0$, set $\theta = \theta + yx$

Very similar to gradient for LR, but *online*.

Theorem (Block & Novikoff): if the training data is linearly separable with margin γ with weights u , then Perceptron will terminate in $\leq 1/\gamma^2$ iterations (assuming $\|x\| \leq 1$ for all x).

Proof sketch: Let θ_k be the weights before the k th update, so $\theta_1 = 0$ and if the k th error is on example n , then $y_n(\theta_k^\top x_n) \leq 0$. Then, $\theta_{k+1}^\top u = \theta_k^\top u + y_n(u^\top x_n) \geq \theta_k^\top u + \gamma$, so $\theta_{k+1}^\top u \geq k\gamma$. Also, $\|\theta_{k+1}\|^2 = \|\theta_k\|^2 + 2y_n(\theta_k^\top x_n) + \|x_n\|^2 \leq \|\theta_k\|^2 + 1$ so $\|\theta_{k+1}\|^2 \leq k$. Thus $\sqrt{k} \geq \|\theta_{k+1}\| \geq \theta_{k+1}^\top u \geq k\gamma$; then algebra.

For the inseparable case, just force it to be separable (add a feature).

Voting: keep a copy of each value of θ and make predictions by voting ($\text{sign}[\sum_i \text{sign}[\theta_i^\top x]]$). *Averaging*: use averaged weights to make prediction ($\text{sign}[(\sum_i \theta_i)^\top x]$).

Support Vector Machines

Large margin principle: want θ that separates the classes maximally. Compute margin as a function of normalized weight vector u , for positive point x^+ and negative point x^- (margin 1 on both):

$$\begin{aligned} \gamma &= \frac{1}{2} \left[\frac{u}{\|u\|}^\top x^+ - \frac{u}{\|u\|}^\top x^- \right] \\ &= \frac{1}{2\|u\|} [u^\top x^+ - u^\top x^-] \\ &= \frac{1}{\|u\|} \end{aligned}$$

Write the learning problem as an optimization problem:

$$\begin{aligned} &\text{minimize}_w \quad \frac{1}{2} \|w\|^2 \\ &\text{subject to} \quad y_n [w^\top x_n + b] - 1 \geq 0 \quad (\forall n) \end{aligned}$$

Introduce Lagrange-multipliers $\alpha_{1:N}$, one for each constraint. Leads to Lagrangian:

$$L(w, \alpha) = \frac{1}{2} \|w\|^2 - \sum_n \alpha_n \{y_n [w^\top x_n + b] - 1\}$$

Now, we want to minimize $L(w, \alpha)$ with respect to *both* w and α . Take derivatives with respect to w :

$$\begin{aligned}\nabla_w L &= w - \sum_n \alpha_n y_n x_n = 0 \\ \implies w &= \sum_n \alpha_n y_n x_n \\ \nabla_b L &= \sum_n y_n \alpha_n = 0\end{aligned}$$

So, given α , w is deterministic... plug back in to L :

$$\begin{aligned}L(\alpha) &= \frac{1}{2} \left\| \sum_n \alpha_n y_n x_n \right\|^2 - \sum_n \alpha_n \left\{ y_n \left[\left(\sum_m \alpha_m y_m x_m \right)^\top x_n + b \right] - 1 \right\} \\ &= \frac{1}{2} \sum_m \sum_n \alpha_m \alpha_n y_m y_n x_m^\top x_n - \sum_m \sum_n \alpha_m \alpha_n y_m y_n x_m^\top x_n - b \sum_n \alpha_n y_n + \sum_n \alpha_n \\ &= -\frac{1}{2} \sum_m \sum_n \alpha_m \alpha_n y_m y_n x_m^\top x_n + \sum_n \alpha_n\end{aligned}$$

So now just solve:

$$\begin{aligned}\text{minimize}_{\alpha} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{n,m} y_n y_m \alpha_n \alpha_m x_n^\top x_m \\ \text{subject to} \quad & \sum_n y_n \alpha_n = 0 \\ & \alpha_n \geq 0 \quad , \quad (\forall n)\end{aligned}$$

Then compute the bias:

$$b = -\frac{1}{2} \left[\max_{n:y_n=-1} \mathbf{w}^\top \mathbf{x}_n + \min_{n:y_n=+1} \mathbf{w}^\top \mathbf{x}_n \right]$$

This leads to a *sparse* solution: most α are zero. Why? The Karush-Kuhn-Tucker conditions say that at the optimum:

$$\alpha_n [y_n (\mathbf{w}^\top \mathbf{x}_n + b) - 1] = 0 \quad (\forall n)$$

This means that $\alpha_n \neq 0$ only when the point is right on the margin. These points are the *support vectors*.

Not linearly-separable data...

Introduce slack parameters: ξ_n is how far "on the wrong side" $y_n x_n$ is from the margin. Then:

$$\begin{aligned} & \text{minimize}_w \quad \frac{1}{2} \|w\|^2 + C \sum_n \xi_n \\ & \text{subject to} \quad y_n [w^\top x_n + b] - 1 + \xi_n \geq 0 \quad , \quad (\forall n) \\ & \quad \quad \quad \xi_n \geq 0 \quad , \quad (\forall n) \end{aligned}$$

High C means “fit data” while low C means “have a large margin.”

Following the same dual formulation, we get:

$$L(w, b, \xi, \alpha, r) = \frac{1}{2} w^\top w + C \sum_n \xi_n - \sum_n \alpha_n [y_n (w^\top x_n + b) - 1 + \xi_n] - \sum_n r_n \xi_n$$

Differentiate:

$$\begin{aligned} \nabla_w L &= w - \sum_n y_n \alpha_n x_n = 0 \\ \implies w &= \sum_n y_n \alpha_n x_n \\ \nabla_b L &= \sum_n y_n \alpha_n = 0 \\ \nabla_{\xi_n} L &= C - \alpha_n - r_n = 0 \end{aligned}$$

This:

$$L(w, b, \xi, \alpha, r) = \sum_n -\frac{1}{2} \sum_{n,m} y_n y_m \alpha_n \alpha_m x_n^\top x_m$$

Which is the same as before, but now we have constraints: $\alpha_n \geq 0$ and $r_n \geq 0$ and $C - \alpha_n - r_n = 0$. This means: (1) $\alpha_n \leq C$. (2) if $r_n = 0$ then $\alpha_n = C$. Geometrically, support vectors are now also the “noisy” points.

Why large margins? Because they mean simple solutions.

Standard result: if selecting from a finite set of hypotheses, then one can achieve low error with high probability in time proportional to the inverse of the size of the set. (Easy proof.) How to extend to infinite hypothesis classes?

VC dimension: measure of complexity of a class of functions.

Based on principle of *shattering*: Let \mathcal{C} be a class of functions. We say that \mathcal{C} shatters \mathcal{X} if \mathcal{C} can achieve perfect classification on *any* labeling of \mathcal{X} .

Then, the VC-dimension of \mathcal{C} is the smallest set d for which no set of $d+1$ examples is shattered by \mathcal{C} . IOW, the VC-dimension is the cardinality of the largest finite set of points that is shattered by \mathcal{C} .

Example: linear models in two dimensions shatter three points. Proof by handwaving. Hence, the VC-dimension is 3. In three dimensions, four points, etc.

Example: axis-aligned rectangles. Can always shatter four. Cannot shatter five. So VC is 4.

Important result: if $\|w\| \leq A$ and $\|x\| \leq R$, then the VC-dimension is at most $A^2 B^2 + 1$.