Support Vector Machine, SVM Loss and Softmax Loss

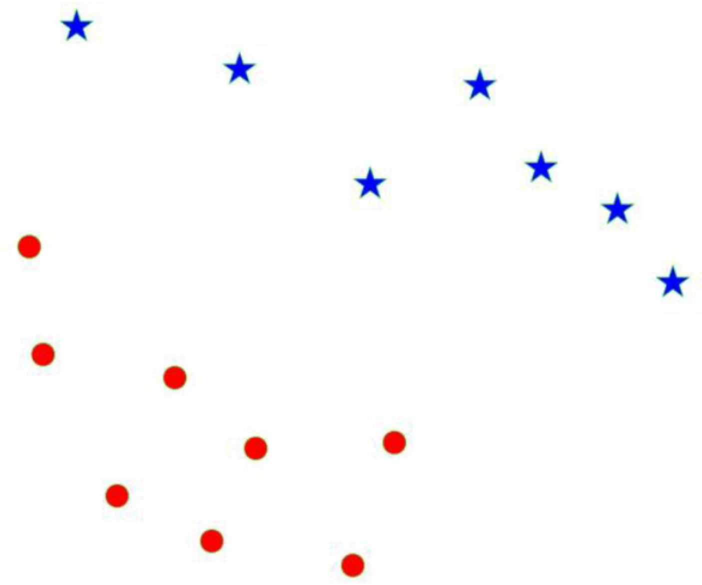
Binary Classification with a Linear Model

- Classification: Predict a discrete-valued target
- Binary classification: Targets $t \in \{-1, +1\}$
- Linear model:

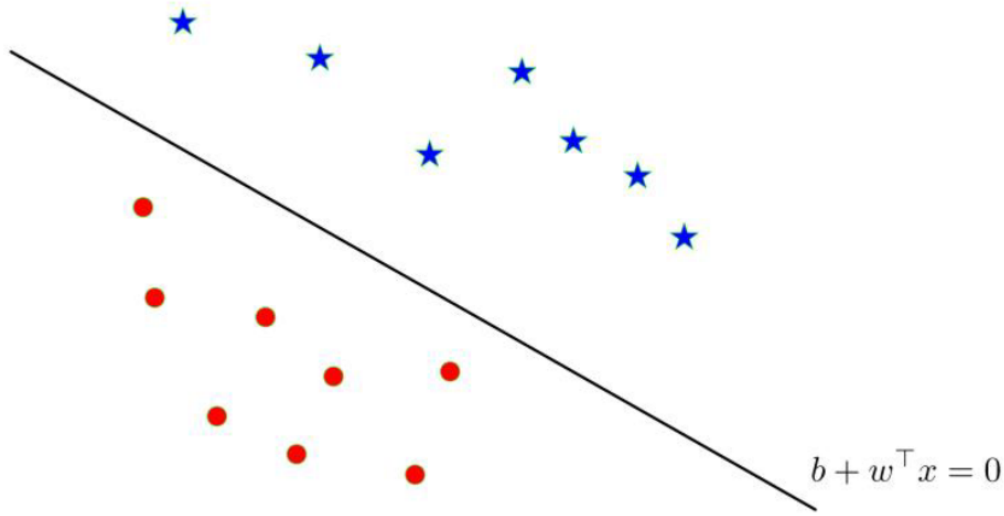
$$z = \mathbf{w}^\top \mathbf{x} + b$$
$$y = sign(z)$$

Separating Hyperplanes

Suppose we are given these data points from two different classes and want to find a linear classifier that separates them.

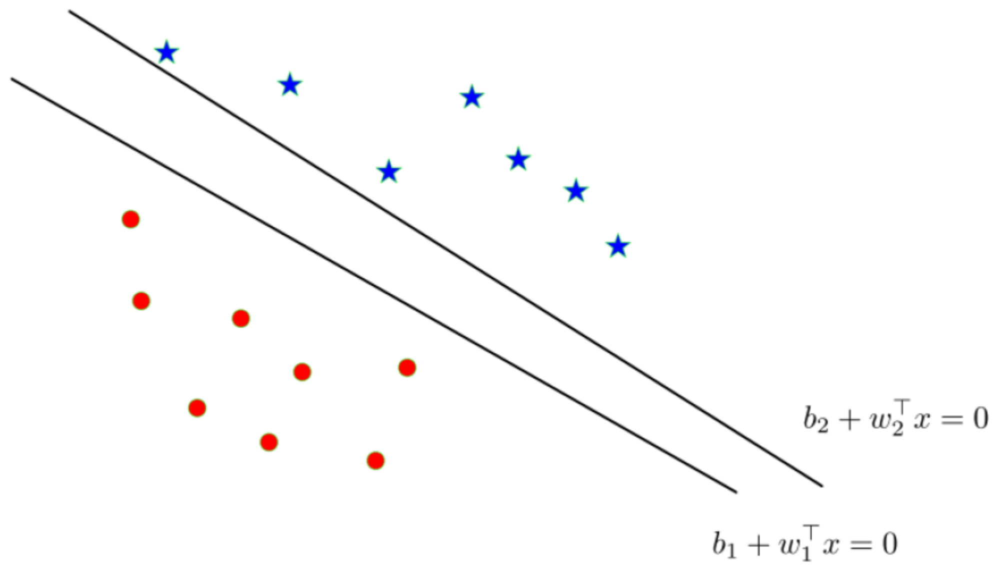


Find the hyperplane:



The decision boundary looks like a line because $\mathbf{x} \in \mathbb{R}^2$, but think about it as a $D-1$ dimensional hyperplane.

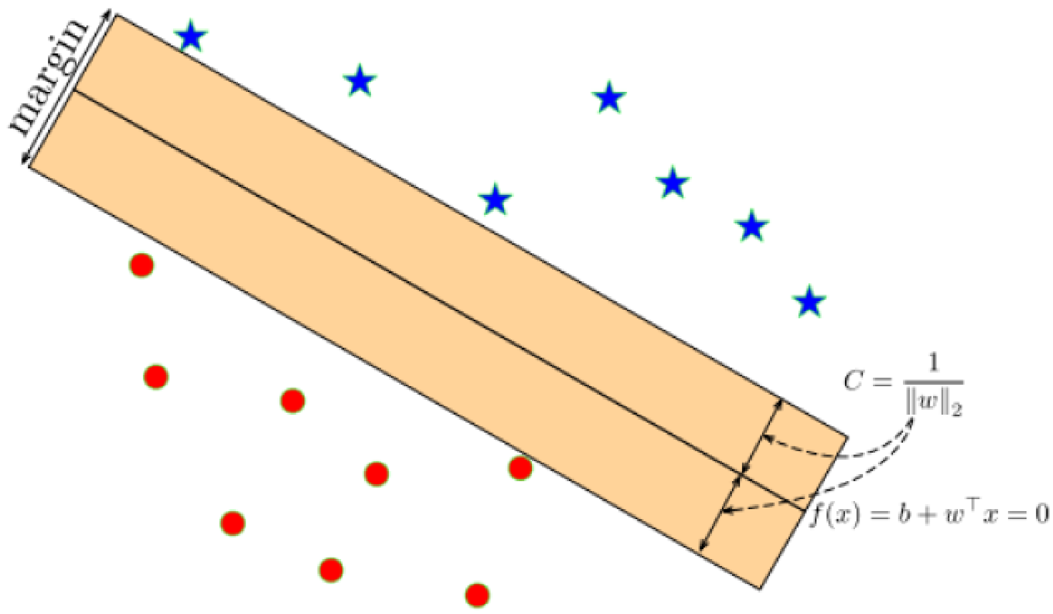
Recall that a hyperplane is described by points $\mathbf{x} \in \mathbb{R}^D$ such that $f(\mathbf{x}) = \mathbf{w}^\top x + b = 0$.



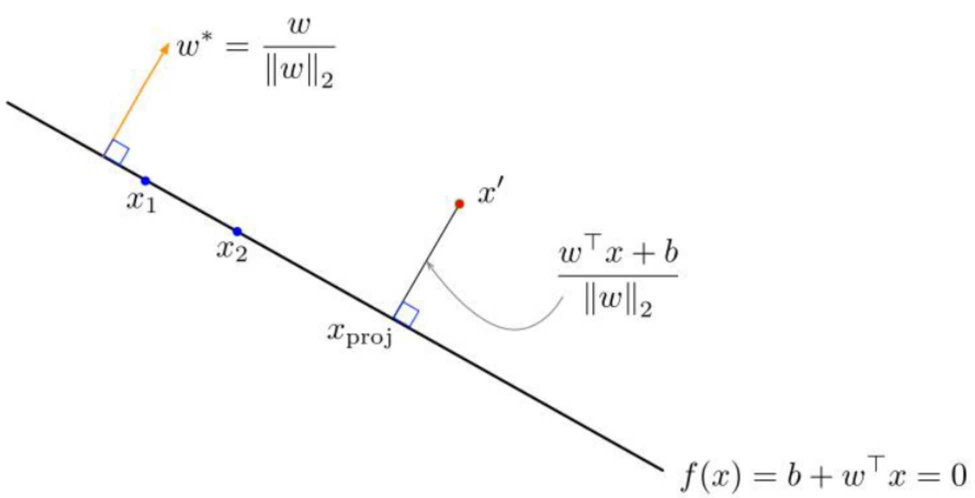
There are multiple separating hyperplanes, described by different parameters $(\mathbf{w}, b)$.

Optimal Separating Hyperplane

Optimal Separating Hyperplane: A hyperplane that separates two classes and maximizes the distance to the closest point from either class, i.e., maximize the **margin** of the classifier.



Geometry of Points and Planes



- Recall that the decision hyperplane is orthogonal (perpendicular) to $\mathbf{w}$.
- The vector $\mathbf{w}^* = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$ is a unit vector pointing in the same direction as $\mathbf{w}$.
- The same hyperplane could equivalently be defined in terms of $\mathbf{w}^*$.

注：为什么 [w 垂直于 decision hyperplane]。

The (signed) distance of a point $\mathbf{x}'$ to the hyperplane is

$$\frac{\mathbf{w}^\top \mathbf{x}' + b}{\|\mathbf{w}\|_2}$$

Maximizing Margin as an Optimization Problem

The classification for the i-th data point is correct when

$$\operatorname{sign}\left(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b\right)=t^{(i)}$$

This can be rewritten as

$$t^{(i)}\left(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b\right)>0$$

注：$t^{(i)}\in\{-1,+1\}$，$\mathbf{w}^{\top}\mathbf{x}^{(i)}+b$ 和 $t^{(i)}$ 正负号一致，下同。

Enforcing a margin of C :

$$t^{(i)}\cdot\underbrace{\frac{\left(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b\right)}{\|\mathbf{w}\|_2}}_{\text{signed distance}}\geq C$$

Max-margin objective:

$$\begin{aligned}&\max_{\mathbf{w},b} C\\ &\text{s.t.}\ \frac{t^{(i)}(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b)}{\|\mathbf{w}\|_2}\geq C \qquad i=1,\ldots,N\end{aligned}$$

Plug in $C=1/\|\mathbf{w}\|_2$ and simplify:

$$\underbrace{\frac{t^{(i)}(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b)}{\|\mathbf{w}\|_2}\geq\frac{1}{\|\mathbf{w}\|_2}}_{\textbf{geometric margin constraint}} \qquad\Longleftrightarrow\qquad \underbrace{t^{(i)}(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b)\geq 1}_{\textbf{algebraic margin constraint}}$$

Equivalent optimization objective:

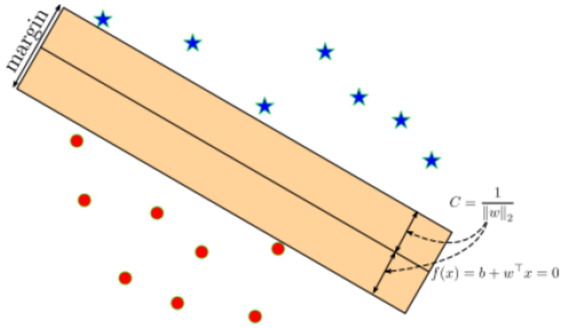
$$\begin{aligned}&\min \|\mathbf{w}\|_2^2\\ &\text{s.t.}\ t^{(i)}(\mathbf{w}^{\top}\mathbf{x}^{(i)}+b)\geq 1 \qquad i=1,\ldots,N\end{aligned}$$

SVM

Algebraic max-margin objective:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2$$

$$\text{s.t. } t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 \qquad i = 1, \ldots, N$$

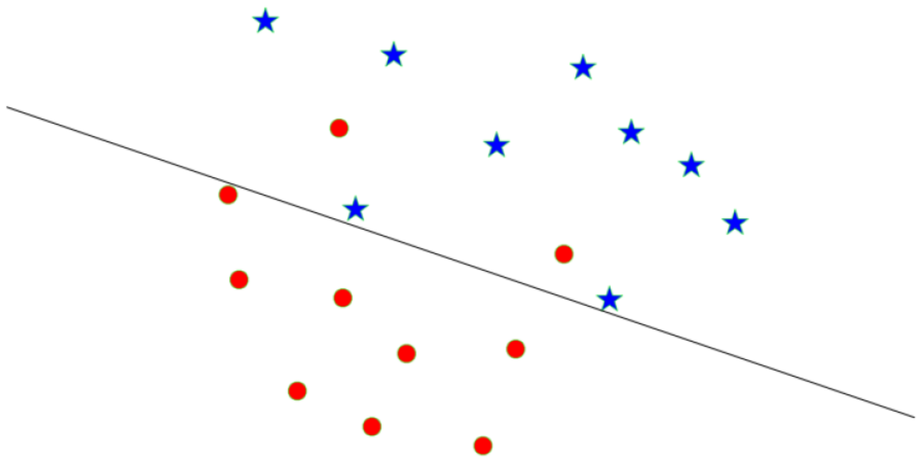


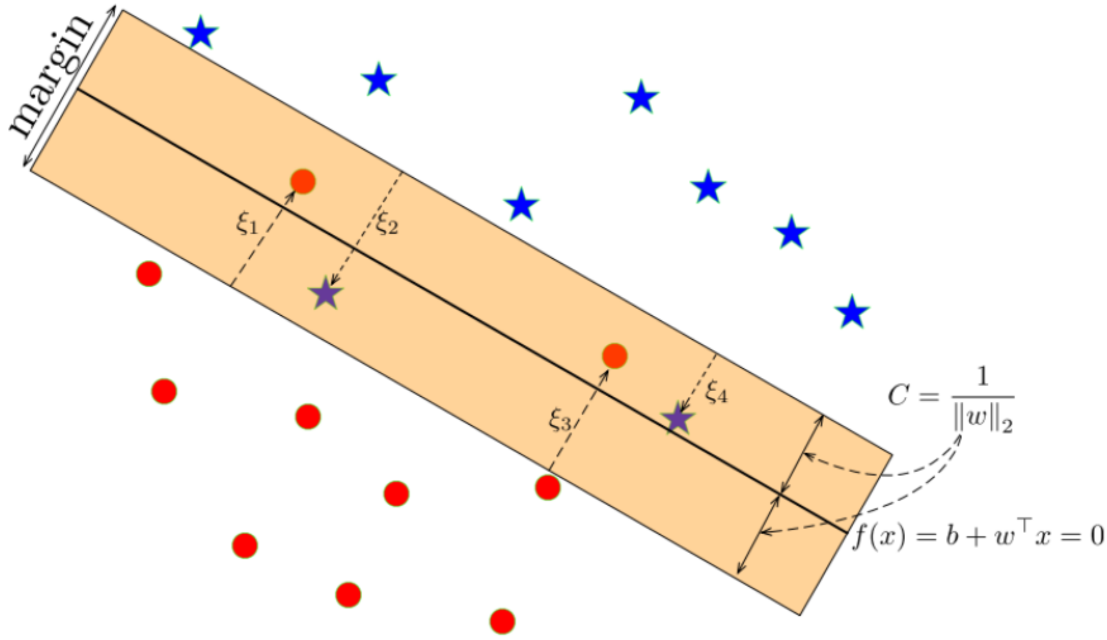
这就是 SVM (Support Vector Machine) 的原理：找到一个 hyperplane 使得类之间的距离最大。SVM-like algorithms are often called **max-margin** or **large-margin**.

找到这个 hyperplane 实际上只需参考距离最近的几个 training examples (or closest point)，而 closest point 到 hyperplane 的向量就是 support vector（closest point is the one with algebraic margin 1）。

Non-Separable Data Points

How can we apply the max-margin principle if the data are **not** linearly separable?





Main Idea:

- Allow some points to be within the margin or even be misclassified; we represent this with **slack variables** ξ_i.
- But constrain or penalize the total amount of slack.

- **Soft margin constraint**:

$$\frac{t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}{\|\mathbf{w}\|_2} \geq C(1 - \xi_i),$$

  for $\xi_i \geq 0$.

- Penalize $\sum_i \xi_i$

注：上式中分为两种情况：

1. point 被正确分类：

   这种情况下，ξ_i 为 0，式子还是 $\frac{t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)}+b)}{\|\mathbf{w}\|_2} \geq C$。

2. point 在 margin 内 或 被误分类：

   假如被误分类，在这种情况下，ξ_i > 1 (or $\xi_i > \frac{1}{\|w\|_2}$)。假如 $x^{(i)}$ 的实际类别为 1，那么 $t^{(i)}$ = 1；但被误分类后，$\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right)$ = -1，因此式子的左边现在变成了 $-\frac{(\mathbf{w}^\top \mathbf{x}'+b)}{\|\mathbf{w}\|_2}$。现在看式子右边，$1 - \xi_i$ 是点到决策面的距离（现在我们把 $\frac{1}{\|w\|_2}$ 当作 1 来处理是为了方便，实际上 $\frac{1}{\|w\|_2}$ 肯定不是 1，所以可以把它看作一个比例），$1 - \xi_i$ 为负号，然后 $C(1 - \xi_i)$ 就是点到决策面的距离。

Soft-margin SVM objective:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \sum_{i=1}^{N} \xi_i \\
\text{s.t.} \ & t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \qquad i = 1, \ldots, N \\
& \xi_i \geq 0 \qquad i = 1, \ldots, N
\end{aligned}$$

注：我们一直的目标是 max margin，因为 margin = C = $\frac{1}{\|\mathbf{w}\|_2}$，所以要 min $\frac{1}{\|\mathbf{w}\|_2}$。我们当然不希望有一个被误分类的点离决策面很远，因此要 min $\sum_{i=1}^{N} \xi_i$。

- γ is a hyperparameter that trades off the margin with the amount of slack.
  - For $\gamma = 0$, we'll get $\mathbf{w} = 0$. (Why?)
  - As $\gamma \to \infty$ we get the hard-margin objective.

注：1. 当 $\gamma = 0$，意为着不对 ξ_i 进行 min。那么假如有一个离决策面无限远的点，它到决策面的距离为 $\frac{\mathbf{w}^\top \mathbf{x}' + b}{\|\mathbf{w}\|_2} = \infty$，解为 w = 0。

2. 当 $\gamma = \infty$，意味着将所有 ξ_i 都 min。假如我们已经将所有 ξ_i 都变成了 0，那么所有的点都不在 margin 内，也没有被误分类，这就是 hard-margin objective。

From Margin Violation to Hinge Loss

Let's simplify the soft margin constraint by eliminating ξ_i. Recall:

$$\begin{aligned}
& t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \qquad i = 1, \ldots, N \\
& \xi_i \geq 0 \qquad i = 1, \ldots, N
\end{aligned}$$

- Rewrite as $\xi_i \geq 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)$.
- **Case 1:** $1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \leq 0$
  - The smallest non-negative ξ_i that satisfies the constraint is $\xi_i = 0$.
- **Case 2:** $1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 0$
  - The smallest ξ_i that satisfies the constraint is $\xi_i = 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)$.
- Hence, $\xi_i = \max\{0, 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\}$.
- Therefore, the slack penalty can be written as

$$\sum_{i=1}^{N} \xi_i = \sum_{i=1}^{N} \max\{0, 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\}.$$

注：在 case 2 中，假如点被误分类到了 1 中，那么 $t^{(i)} = -1$，它的作用是控制符号。$\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right)$ 是一个正值，它代表着到决策面的距离，所以 $1 - t^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right)$ 代表 γ
。

If we write $y^{(i)}(\mathbf{w}, b) = \mathbf{w}^\top \mathbf{x} + b$, then the optimization problem can be written as

$$\min_{\mathbf{w},b,\xi} \sum_{i=1}^{N} \max\{0, 1 - t^{(i)} y^{(i)}(\mathbf{w}, b)\} + \frac{1}{2\gamma} \|\mathbf{w}\|_2^2$$

- The loss function $\mathcal{L}_{\mathrm{H}}(y, t) = \max\{0, 1 - ty\}$ is called the **hinge** loss.
- The second term is the L_2-norm of the weights.
- Hence, the soft-margin SVM can be seen as a linear classifier with hinge loss and an L_2 regularizer.

Multiclass SVM loss

Given an example (x_i, y_i) where $\boldsymbol{x}_i$, is the image and where y_i is the (integer) label, and using the shorthand for the scores vector: $s = f(x_i, W) = Wx_i$.

The SVM loss has the form:

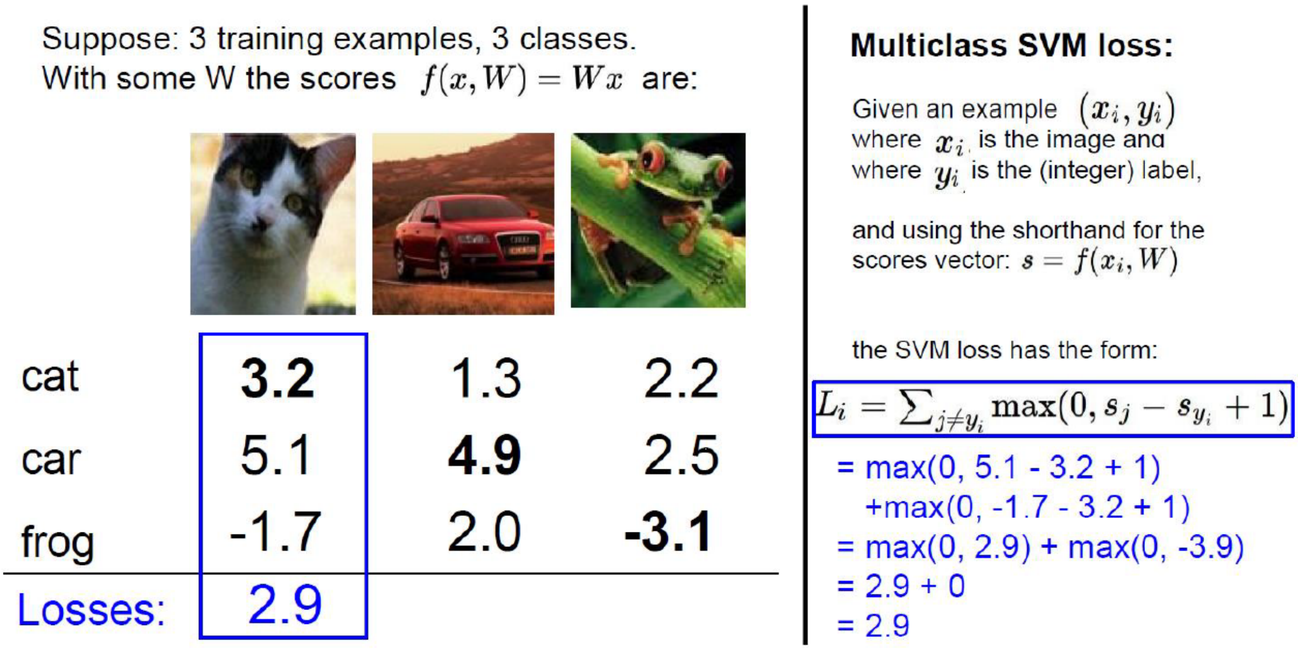
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

注：s_j 是对 x_i 的预测值，s_{y_i} 是数据 x_i 的正确 label。

现在我们有 3 个 training example，和 3 个类别，它们的预测结果如下：

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

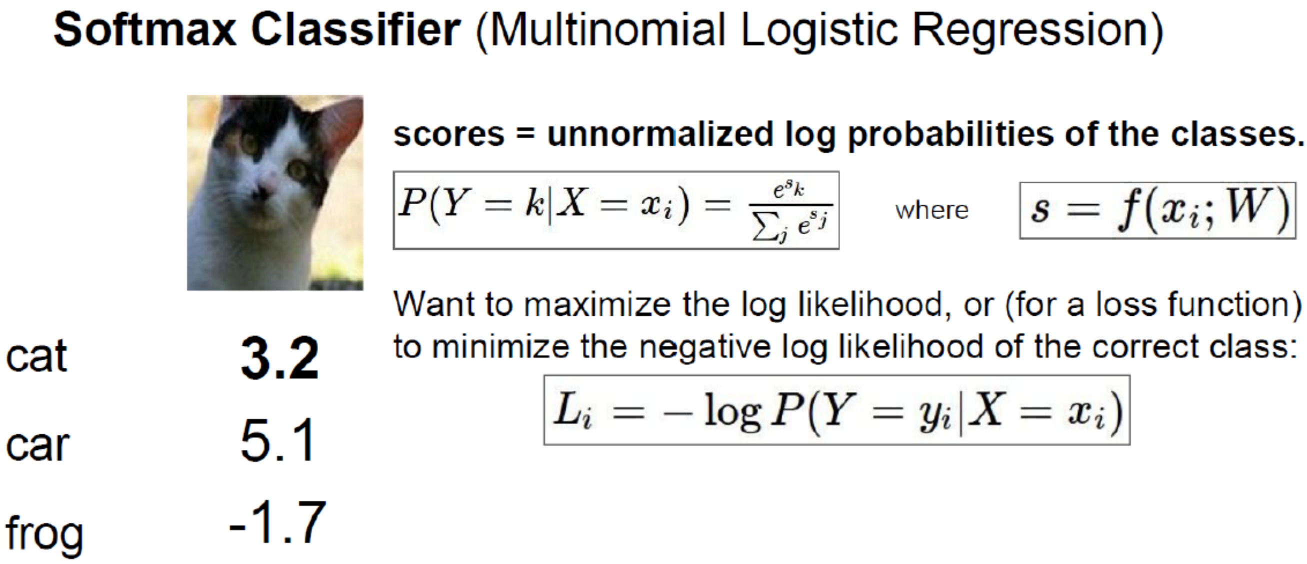
然后计算出它们的 SVM loss：



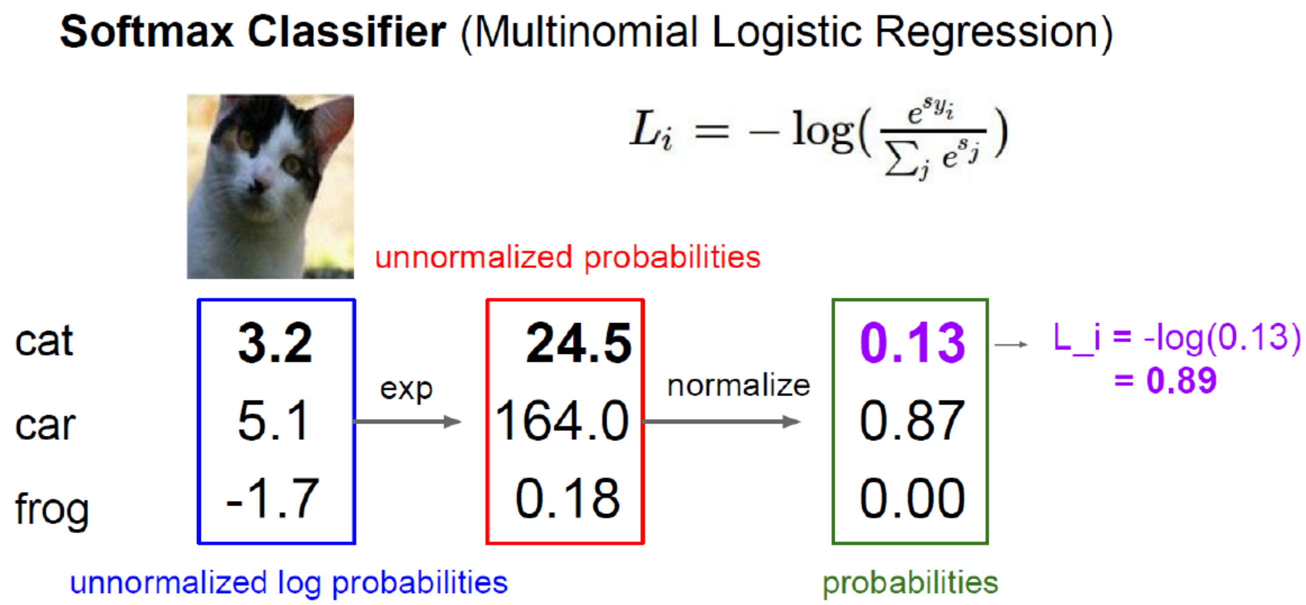
注：计算 SVM loss 时，不算正确的那一类（即 $j \neq y_i$）

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	10.9

Softmax Classifier (Multinomial Logistic Regression)



注：$P\left(Y=k \mid X=x_i\right)$ 意为在 $X=x_i$ 的条件下，$Y=k$ 的概率。



注：以 3.2 为例，从 unnormalized log probabilities 到 unnormalized probabilities，要进行 $e^{3.2}$ = 24.5，即公式分子的操作。然后是 24.5 / (24.5 + 164.0 + 0.18) = 0.13，这是公式括号里面的操作。最后得到 Li 作为 loss。

Softmax loss vs. SVM loss

Softmax vs. SVM

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right) \qquad L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

assume scores:
[10, -2, 3]
[10, 9, 9]
[10, -100, -100]
and $\boxed{y_i = 0}$

Q: Suppose I take a datapoint and I jiggle a bit (changing its score slightly). What happens to the loss in both cases?

注：上面的后两个，SVM loss 都是 0，因此 SVM 无法很好的表现优化的情况。