Linear classifiers – Which line is better?
Pick the one with the largest margin!

Margin for point $j$:

$$\gamma^j = y^j (w \cdot x^j + w_0)$$

Max Margin:

$$\max_{\gamma, w, w_0} \gamma$$

$$\forall j. y^j (w \cdot x^j + w_0) > \gamma$$

$$y = w \cdot x + w_0$$
How many possible solutions?

Any other ways of writing the same dividing line?

- $w \cdot x + b = 0$
- $2w \cdot x + 2b = 0$
- $1000w \cdot x + 1000b = 0$
- ....
- Any constant scaling has the same intersection with $z=0$ plane, so same dividing line!

Do we really want to max $\gamma$, $w$, $w_0$?
Review: Normal to a plane

$$w \cdot x + w_0 = 0$$

Key Terms

-- projection of $x^j$ onto $w$

-- unit vector normal to $w$

$$x^j = \overline{x^j} + \lambda \frac{w}{\|w\|_2}$$

$$\|w\|_2 = \sqrt{\sum_i w_i^2}$$
\[ x^j = \bar{x}^j + \lambda \frac{w}{\|w\|_2} \]

\[ \|w\|_2 = \sqrt{\sum_i w_i^2} \]

Assume: \( x^+ \) on positive line \((y=1)\) intercept, \( x^- \) on negative \((y=-1)\)

\[
x^+ = x^- + 2\gamma \frac{w}{\|w\|_2}
\]

\[
w \cdot x^+ + w_0 = 1
\]

\[
w \cdot (x^- + 2\gamma \frac{w}{\|w\|_2}) + w_0 = 1
\]

\[
w \cdot x^- + w_0 + 2\gamma \frac{w \cdot w}{\|w\|_2} = 1
\]

\[
\gamma \frac{w \cdot w}{\|w\|_2} = 1 \quad w \cdot w = \sum_i w_i^2 = \|w\|_2^2
\]

\[
\gamma = \frac{\|w\|_2}{w \cdot w} = \frac{1}{\|w\|_2}
\]

Final result: can maximize constrained margin by minimizing \( \|w\|_2 \).
Max margin using canonical hyperplanes

The assumption of canonical hyperplanes (at +1 and -1) changes the objective and the constraints!
Support vector machines (SVMs)

- Solve efficiently by quadratic programming (QP)
  - Well-studied solution algorithms
  - Not simple gradient ascent, but close
- Decision boundary defined by support vectors

**Support Vectors:**
- data points on the canonical lines

**Non-support Vectors:**
- everything else
- moving them will not change w

\[
\begin{align*}
\min_{w,w_0} & \frac{1}{2}||w||^2_2 \\
\forall j, y^j(w \cdot x^j + w_0) & \geq 1
\end{align*}
\]
\[
\min \frac{1}{2} \|W\|_2^2
\]
\[
\begin{align*}
\begin{cases}
wx_1 + b > 1 \\
w_2x_2 + b > 1 \\
w_3x_3 + b > 1
\end{cases}
\end{align*}
\]
\[
-1 + \langle wx_1 + b \rangle > 0 \\
-1 + \langle wx_2 + b \rangle > 0 \\
-1 + \langle wx_3 + b \rangle > 0
\]
\[
\min \frac{1}{2} \|W\|_2^2 + \lambda_1 (wx_1 + b - 1) + \lambda_2 (wx_2 + b - 1) + \lambda_3 (wx_3 + b - 1)
\]
\[
\frac{d}{dw}
\]
What if the data is not linearly separable?

Add More Features!!!

\[
\phi(x) = \begin{pmatrix}
  x_1 \\
  \ldots \\
  x_n \\
  x_1 x_2 \\
  x_1 x_3 \\
  \ldots \\
  e^{x_1} \\
  \ldots 
\end{pmatrix}
\]

Can use Kernels… (more on this later)

What about overfitting?
What if the data is still not linearly separable?

\[
\min_{w, w_0} \frac{1}{2} \|w\|_2^2 + C \#(\text{mistakes})
\]

\[\forall j, y^j (w \cdot x^j + w_0) \geq 1\]

- First Idea: Jointly minimize \(\|w\|_2^2\) and number of training mistakes
  - How to tradeoff two criteria?
  - Pick C on development / cross validation

- Tradeoff \(\#(\text{mistakes})\) and \(\|w\|_2^2\)
  - 0/1 loss
  - Not QP anymore
  - Also doesn’t distinguish near misses and really bad mistakes
  - NP hard to find optimal solution!!!
Slack variables – Hinge loss

For each data point:
- If margin $\geq 1$, don’t care
- If margin $< 1$, pay linear penalty

$$\min_{w, w_0} \frac{1}{2} \|w\|_2^2 + C \sum_j \xi_j$$

$$\forall j. y^j (w \cdot x^j + w_0) \geq 1 - \xi_j, \quad \xi_j \geq 0$$

Slack Penalty $C > 0$:
- $C = \infty \rightarrow$ have to separate the data!
- $C = 0 \rightarrow$ ignore data entirely!
- Select on dev. set, etc.

$C$
Slack variables – Hinge loss

\[
\min_{w,w_0} \frac{1}{2} \|w\|^2 + C \sum_j \xi_j
\]

\[\forall j. y^j (w \cdot x^j + w_0) \geq 1 + \xi_j, \xi_j \geq 0\]

\[[x]^+_+ = \max(x,0)\]

\[
\min_{w,w_0} \frac{1}{2} \|w\|^2 + C \sum_{j=1}^{N} [1 - y^j (w \cdot x^j + w_0)]^+_+
\]

Solving SVMs:
- Differentiate and set equal to zero!
- No closed form solution, but quadratic program (top) is concave
- Hinge loss is not differentiable, gradient ascent a little trickier…
Logistic Regression as Minimizing Loss

Logistic regression assumes:

\[ P(Y = 1 | X = x) = \frac{\exp(f(x))}{1 + \exp(f(x))} \]

And tries to maximize data likelihood, for \( Y = \{-1, +1\} \):

\[ P(y^i | x^i) = \frac{1}{1 + \exp(-y^i f(x^i))} \]

\[ \ln P(D_Y | D_X, w) = \sum_{j=1}^{N} \ln P(y^j | x^j, w) \]
\[ = - \sum_{i=1}^{N} \ln(1 + \exp(-y^i f(x^i))) \]

Equivalent to minimizing log loss:

\[ \sum_{i=1}^{N} \ln(1 + \exp(-y^i f(x^i))) \]
SVMs vs Regularized Logistic Regression

**SVM Objective:**

$$f(x) = w_0 + \sum_i w_i x_i$$

$$\arg\min_{w, w_0} \frac{1}{2} \|w\|_2^2 + C \sum_{j=1}^N [1 - y^j f(x^j)]_+$$

Tradeoff: same $l_2$ regularization term, but different error term

**Logistic regression objective:**

$$\arg\min_{w, w_0} \lambda \|w\|_2^2 + \sum_{j=1}^N \ln(1 + \exp(-y^j f(x^j)))$$
Logistic regression:

\[
\ln(1 + \exp(-y^j f(x^j)))
\]

Hinge loss:

\[
[1 - y^j f(x^j)]^+
\]

0-1 Loss:

\[
\delta(f(x^j) \neq y^j)
\]

We want to smoothly approximate 0/1 loss!
What about multiple classes?
One against All

Learn 3 classifiers:
• \(+\) vs \(\{0,-\}\), weights \(w_+\)
• \(-\) vs \(\{0,+,+\}\), weights \(w_-\)
• 0 vs \(\{+,+-\}\), weights \(w_0\)

Output for \(x\):
\[
y = \text{argmax}_i \ w_i \cdot x
\]

Any problems?
Could we learn this \(\rightarrow\) dataset?
\[
\text{Min } \sum_{i}^{1} \frac{1}{2} ||w_i||_2^2 + \sum_{i}c_i \\
\rightarrow w_i \cdot x_i + w_0 \geq 1 - \xi_i \\
w_i \cdot x_i + w_0 \geq 0 \\
\max \left\{ \frac{w_i \cdot x}{w_i \cdot x} \right\} \\
w_i \cdot x > w_i \cdot x^+ + \xi_i \\
w_i \cdot x > w_i \cdot x^- + \xi_i \\
\]
Learn 1 classifier: Multiclass SVM

Simultaneously learn 3 sets of weights:

- How do we guarantee the correct labels?
- Need new constraints!

For each class:

\[
w^{y^j} \cdot x^j + w_0^{y^j} \geq w^{y'} \cdot x^j + w_0^{y'} + 1, \quad \forall y' \neq y^j, \quad \forall j
\]
Learn 1 classifier: Multiclass SVM

Also, can introduce slack variables, as before:

$$\min_{w,w_0} \sum_{y} \|w^y\|_2^2 + C \sum_{j} \xi^j$$

$$w^{y_j} \cdot x^j + w_0^{y_j} \geq w^{y'} \cdot x^j + w_0^{y'} + 1 - \xi^j, \quad \forall y' \neq y^j, \quad \xi^j > 0 \quad \forall j$$

Now, can we learn it?
What you need to know

• Maximizing margin
• Derivation of SVM formulation
• Slack variables and hinge loss
• Tackling multiple class
  – One against All
  – Multiclass SVMs