Review: Linear models
Training linear classifiers

• **Given:** i.i.d. training data \((x_i, y_i), i = 1, \ldots, n\), \(y_i \in \{-1,1\}\)

• **Prediction function:** \(f_w(x) = \text{sgn}(w^T x)\)

• Classification with *bias*, i.e. \(f_w(x) = \text{sgn}(w^T x + b)\), can be reduced to the case w/o bias by letting \(w' = [w; b]\) and \(x' = [x; 1]\)
General recipe

- Find parameters $w$ that minimize the sum of a \textit{regularization loss} and a \textit{data loss}:

$$
\hat{L}(w) = \lambda R(w) + \frac{1}{n} \sum_{i=1}^{n} l(w, x_i, y_i)
$$

- Optimize by \textit{stochastic gradient descent (SGD)}: At each iteration, sample a single data point $(x_i, y_i)$ and take a step in the direction \textit{opposite} the gradient of the loss for that point:

$$
w \leftarrow w - \eta \nabla_w \left[ \frac{\lambda}{n} R(w) + l(w, x_i, y_i) \right]
$$
Model 1: Linear regression

- **Regularization**: none
- **Data loss**: \( l(w, x_i, y_i) = (w^T x_i - y_i)^2 \)
- **Interpretation**: negative log likelihood assuming \( y|x \) is normally distributed with mean \( w^T x \)
- **Pros**: convex loss, easy to optimize
- **Cons**: conceptually inappropriate for classification, sensitive to outliers
Model 2: Logistic regression
Model 2: Logistic regression

- Posterior label probability or confidence is given by the **sigmoid function**:

\[
P_w(y = 1|x) = \sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)}
\]

\[
P_w(y = -1|x) = 1 - \sigma(w^T x) = \sigma(-w^T x)
\]
Model 2: Logistic regression

- **Regularization**: none
- **Data loss**: 
  \[ l(w, x_i, y_i) = -\log P_w(y_i|x_i) = -\log \sigma(y_i w^T x_i) \]
- **Interpretation**: negative log likelihood assuming Gaussian class-conditional distributions \( P(x|y = 1) \)
Model 3: Perceptron training algorithm
Model 3: Perceptron training algorithm

- **Regularization:** none
- **Data loss:** hinge loss

\[ l(w, x_i, y_i) = \max(0, -y_i w^T x_i) \]
Model 4: Support vector machines
Model 4: Support vector machines

- **Regularization**: $R(w) = \frac{1}{2} \|w\|^2$
- **Data loss**: *hinge loss*

$$l(w, x_i, y_i) = \max(0, 1 - y_i w^T x_i)$$

[Diagram showing the hinge loss function for Perceptron and SVM compared to the correct classification boundary.]
Model 4: Support vector machines

- **Regularization:** \( R(w) = \frac{1}{2} \|w\|^2 \)
- **Data loss:** *hinge loss*

\[
l(w, x_i, y_i) = \max(0, 1 - y_i w^T x_i)\]

- **Interpretation:** maximize margin while minimizing constraint violations
SGD updates

- Linear regression:
  \[ w \leftarrow w + \eta (y_i - w^T x_i) x_i \]

- Logistic regression:
  \[ w \leftarrow w + \eta \sigma(-y_i w^T x_i) y_i x_i \]

- Perceptron:
  \[ w \leftarrow w + \eta \mathbb{I}[y_i w^T x_i < 0] y_i x_i \]

- SVM:
  \[ w \leftarrow \left(1 - \frac{\eta \lambda}{n}\right) w + \eta \mathbb{I}[y_i w^T x_i < 1] y_i x_i \]