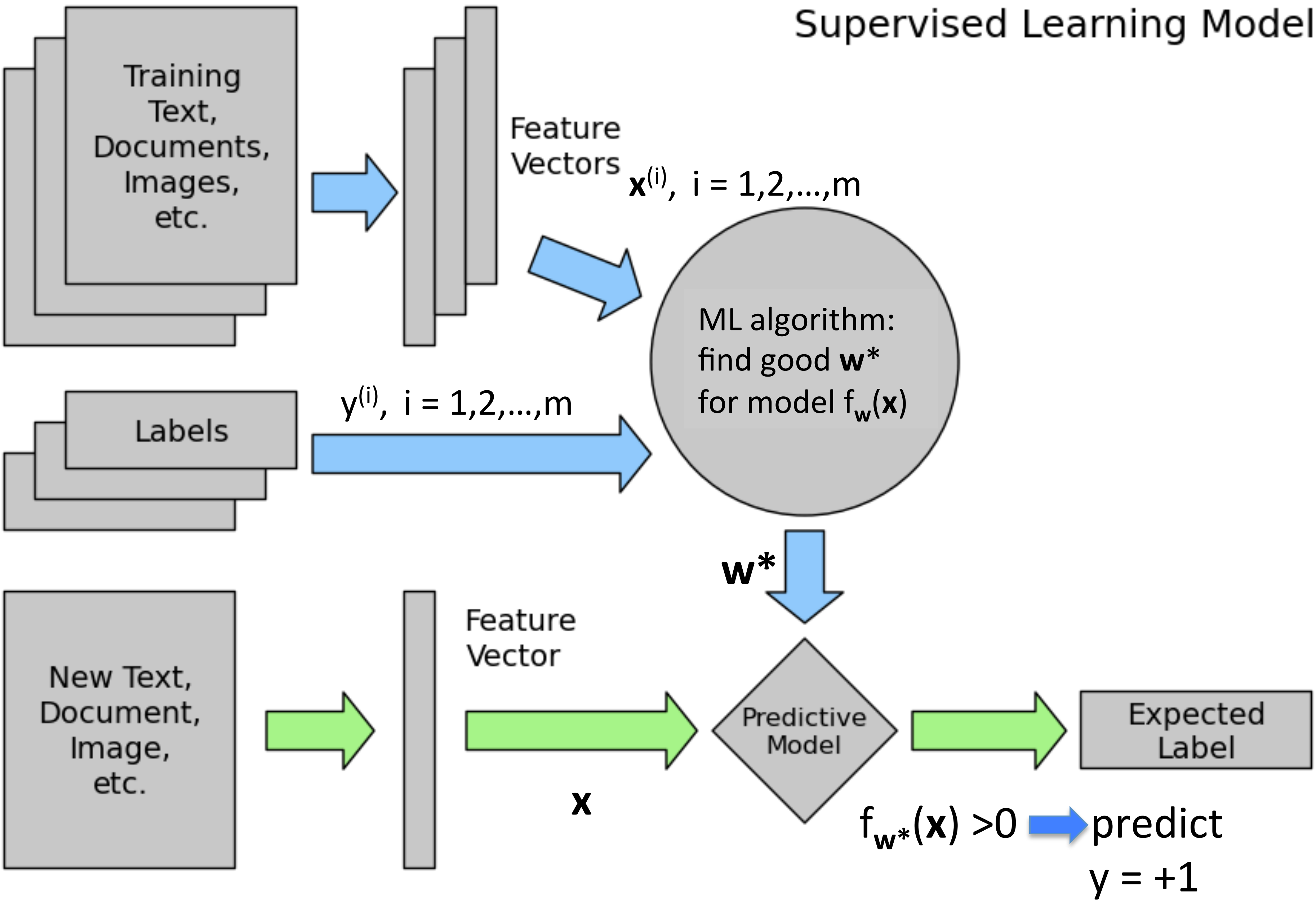
Supervised Machine Learning

An Overview of what
we’ve seen so far



Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
$\mathbf{x}^{(i)}$, i = 1,2,…,m
ML algorithm: find good $\mathbf{w}^*$ for model $f_w(\mathbf{x})$
Labels
$y^{(i)}$, i = 1,2,…,m
$\mathbf{w}^*$
New Text, Document, Image, etc.
Feature Vector
$\mathbf{x}$
Predictive Model
Expected Label
$f_{\mathbf{w}^*}(\mathbf{x}) > 0$ predict $y = +1$

Models so far

- Perceptron:
  - Linear model $h_{\mathbf{w}}(\mathbf{x})$
  - Requires data to be separable.
  - Finds good, not optimal $\mathbf{w}^*$ using a simple algorithm.
- Hard-margin SVM:
  - Linear model $h_{\mathbf{w}}(\mathbf{x})$
  - Requires data to be separable.
  - Finds "optimal" $\mathbf{w}^*$ by solving a constrained optimization problem to maximize margin
- Soft-margin SVM:
  - Linear model $h_{\mathbf{w}}(\mathbf{x})$
  - General data
  - Finds optimal $\mathbf{w}^*$ by solving a constrained optimization problem
- Soft-margin SVM with kernels.
  - Non-linear model $h_{\mathbf{w}}(\mathbf{x})$
  - General data
  - Finds optimal $\mathbf{w}^*$ by solving dual optimization problem. Uses SMO (Sequential Minimal Optimization)

Models so far (Part 2)

- Logistic Regression with MSE loss function:
  - Non-linear model $h_{\mathbf{w}}(\mathbf{x})$
    - But linear decision boundary
  - General data
  - Non-convex
  - Finds local optimal **w*** with gradient descent
- Logistic Regression with MLE loss function:
  - Convex
  - Finds optimal **w*** with gradient descent

- Logistic Regression with MLE loss function:
  - Convex
  - Finds optimal **w*** with gradient descent

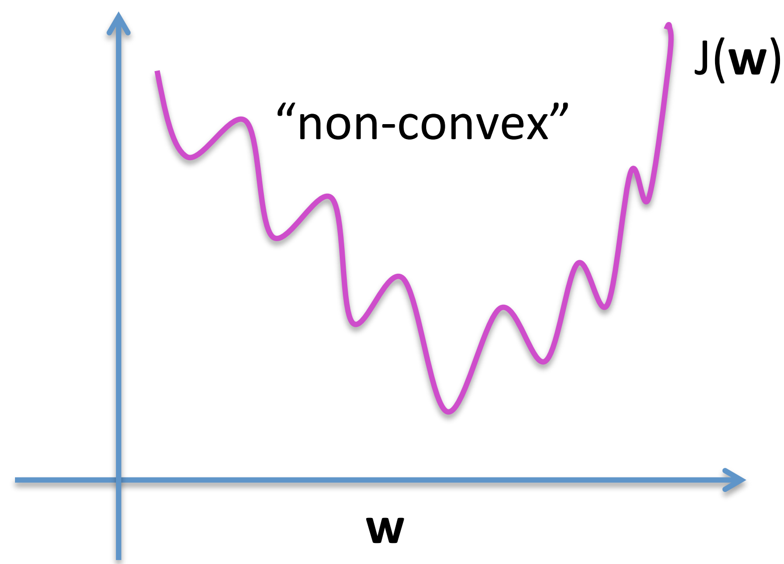
Logistic Regression: MLE

- Training set: $\{ (\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)}) \}$
- How about choosing w to minimize MSE (as usual)?

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_{\boldsymbol{w}}(\boldsymbol{x}^{(i)}), y^{(i)})$$

$$Cost(h_{\boldsymbol{w}}(\boldsymbol{x}), y) = \frac{1}{2}(h_{\boldsymbol{w}}(\boldsymbol{x}) - \mathrm{y})^2$$

$$h_{\boldsymbol{w}}(x) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$



Gradient Descent

$$J(\boldsymbol{w}) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) + (1 - y^{(i)}) \log\left(1 - h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right)\right)\right]$$

Want $\min_{\mathbf{w}} J(\mathbf{w})$:

Repeat {

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\boldsymbol{w})$$

} (simultaneously update all $\mathbf{w}_j$)

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} (h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) - y^{(i)}) x_j^{(i)}$$

$$w_j := w_j - \alpha \sum_{i=1}^{m} (h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) - y^{(i)}) x_j^{(i)}$$

Algorithm looks identical to linear regression! But it is not!

Gradient Descent Algorithm

Initialize $\mathbf{w}$

Repeat:

For j = 0,1,...,n

$$\widetilde{w}_j = w_j - \alpha \sum_{i=1}^{m} [f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}] x_j^{(i)}$$

$$\mathbf{w} \leftarrow \widetilde{\mathbf{w}}$$

Stochastic Gradient Descent (SGD)

Initialize $\mathbf{w}$

Repeat:

For i = 1,2,...,m

For j = 0,1,...,n

$$\widetilde{w}_j = w_j - \alpha[f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}]x_j^{(i)}$$

$\mathbf{w} \leftarrow \widetilde{\mathbf{w}}$

Each update of **w** uses a single data point $\mathbf{x}^{(i)}$!

General Gradient Descent

$$\mathbf{w} = \mathbf{w} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$$

$$J(w) = \Sigma_i \text{ cost } (h_{\mathbf{w}}(\mathbf{x}^{(i)}), y^{(i)})$$

What is $\text{cost}(h_{\mathbf{w}}(\mathbf{x}), y)$ for MSE? For MLE?

What is a general expression for the partial derivative?

Models so far (3)

- Neural networks with MLE loss function:
  - Looking at binary classification for now
  - Use same MLE cost function as logistic regression
  - Finds **w*** with gradient descent
  - To evaluate $h_{\mathbf{w}}(\mathbf{x})$ use forward propagation
  - To evaluate partial derivatives, use back propagation
    - They are different from logistic regression. Why?
  - Let's look into the partial derivatives.