Machine Learning

A. G. Schwing & M. Telgarsky

University of Illinois at Urbana-Champaign, 2018

Review: Discriminative Methods

Goals of this lecture

Goals of this lecture

- Review of what we learned so far

Recap: Empirical risk minimization

Recap: Empirical risk minimization

$$\min_{\boldsymbol{w}} R(\boldsymbol{w}) + \sum_{i \in \mathcal{D}} \ell(\boldsymbol{y}^{(i)}, \hat{F}(\boldsymbol{w}, x^{(i)}))$$

where

$$\hat{F}(\boldsymbol{w}, x^{(i)}) =$$

Recap: Empirical risk minimization

$$\min_{\boldsymbol{w}} R(\boldsymbol{w}) + \sum_{i \in \mathcal{D}} \ell(\boldsymbol{y}^{(i)}, \hat{F}(\boldsymbol{w}, x^{(i)}))$$

where

$$\hat{F}(\boldsymbol{w}, x^{(i)}) = \arg\max_{\hat{\boldsymbol{y}}} F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})$$

So far:

So far:

- kNN

So far:

- kNN
- Least squares

So far:

- kNN
- Least squares
- Logistic regression

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines
- Multiclass classification

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines
- Multiclass classification
- Deep nets

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines
- Multiclass classification
- Deep nets
- Ensemble methods

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines
- Multiclass classification
- Deep nets
- Ensemble methods
- Structured prediction

So far:

- kNN
- Least squares
- Logistic regression
- Convex sets
- Convex optimization and duality
- Support vector machines
- Multiclass classification
- Deep nets
- Ensemble methods
- Structured prediction
- Learning theory

Nearest Neighbor

Nearest Neighbor

- Dataset:

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k =$$

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k = \arg\min_{i\in\{1,\ldots,N\}} \|x^{(i)} - x\|_2^2$$

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

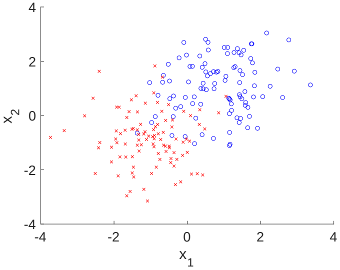
$$y = y^{(k)} \quad \text{where} \quad k = \arg \min_{i \in \{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg \min_{i \in \{1,\ldots,N\}} d(x^{(i)}, x)$$

Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k = \arg\min_{i\in\{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg\min_{i\in\{1,\ldots,N\}} d(x^{(i)}, x)$$

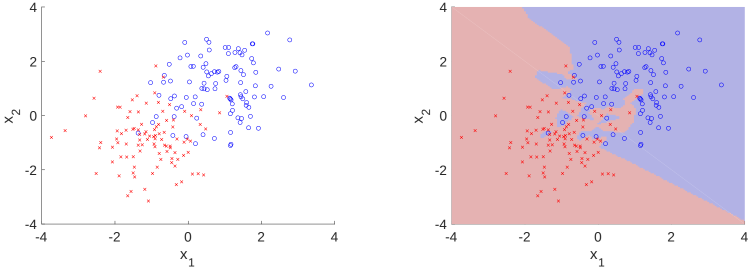


Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k = \arg\min_{i\in\{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg\min_{i\in\{1,\ldots,N\}} d(x^{(i)}, x)$$

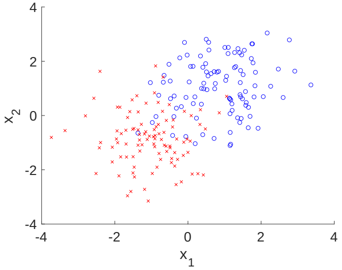


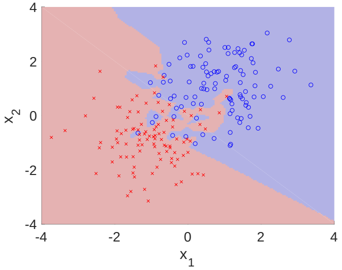
Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k = \arg\min_{i\in\{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg\min_{i\in\{1,\ldots,N\}} d(x^{(i)}, x)$$





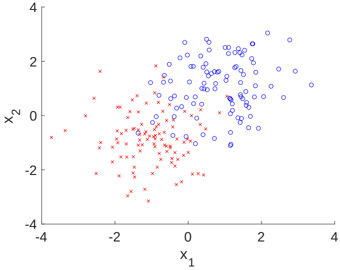
Shortcomings?

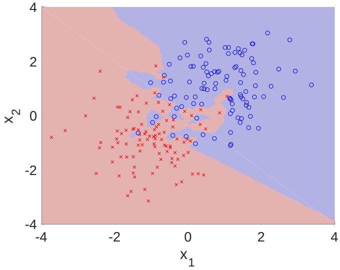
Nearest Neighbor

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

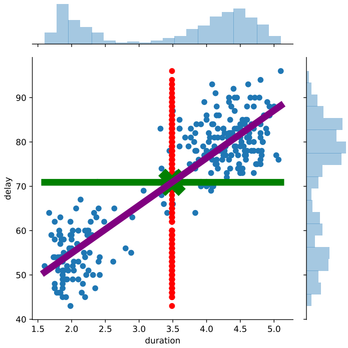
$$y = y^{(k)} \quad \text{where} \quad k = \arg\min_{i \in \{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg\min_{i \in \{1,\ldots,N\}} d(x^{(i)}, x)$$





Shortcomings?
k-Nearest Neighbors

Least Squares



Least squares problem

$\arg\min_{\boldsymbol{w}\in\mathbb{R}^{d+1}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$.

OLS solution

$(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$.

Question:

still "why this line" ?

Three justifications/interpretations.

- Geometric interpretation.
- Probabilistic model.
- Loss minimization.

Logistic regression

Model:

$$p(y^{(i)}|x^{(i)}) = \frac{1}{1+\exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))}$$

Task:

$$\arg\max_{\boldsymbol{w}} \prod_{(x^{(i)},y^{(i)})\in\mathcal{D}} p(y^{(i)}|x^{(i)}) =$$

Logistic regression

Model:

$$p(y^{(i)}|x^{(i)}) = \frac{1}{1 + \exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))}$$

Task:

$$\arg\max_{\boldsymbol{w}} \prod_{(x^{(i)},y^{(i)})\in\mathcal{D}} p(y^{(i)}|x^{(i)}) = \arg\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} -\log p(y^{(i)}|x^{(i)})$$

Logistic regression

Model:

$$p(y^{(i)}|x^{(i)}) = \frac{1}{1 + \exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))}$$

Task:

$$\arg\max_{\boldsymbol{w}} \prod_{(x^{(i)},y^{(i)})\in\mathcal{D}} p(y^{(i)}|x^{(i)}) = \arg\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} -\log p(y^{(i)}|x^{(i)})$$

Combined:

Logistic regression

Model:

$$p(y^{(i)}|x^{(i)}) = \frac{1}{1 + \exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))}$$

Task:

$$\arg\max_{\boldsymbol{w}} \prod_{(x^{(i)},y^{(i)})\in\mathcal{D}} p(y^{(i)}|x^{(i)}) = \arg\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} -\log p(y^{(i)}|x^{(i)})$$

Combined:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))\right)$$

Logistic Regression

Linear regression

Program:

Logistic regression

Program:

Logistic Regression

Linear regression

Program:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \frac{1}{2}\underbrace{(1-y^{(i)}\underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})}_{F(x^{(i)},w,y^{(i)})}^2$$

Logistic regression

Program:

Logistic Regression

Linear regression

Program:

$$\min_{\mathbf{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \underbrace{\frac{1}{2}(1-y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2}_{F(x^{(i)},w,y^{(i)})}$$

Logistic regression

Program:

$$\min_{\mathbf{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1+\exp(-\underbrace{y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)}}_{F(x^{(i)},w,y^{(i)})})\right)$$

Logistic Regression

Linear regression

Program:

$$\min_{\mathbf{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \underbrace{\frac{1}{2}(1-y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2}_{F(x^{(i)},w,y^{(i)})}$$

Logistic regression

Program:

$$\min_{\mathbf{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1+\exp(-\underbrace{y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})}_{F(x^{(i)},w,y^{(i)})}\right)$$

Empirical risk minimization:

Logistic Regression

Linear regression

Program:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \underbrace{\frac{1}{2}(1-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2}_{F(x^{(i)},w,y^{(i)})}$$

Logistic regression

Program:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1+\underbrace{\exp(-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})}_{F(x^{(i)},w,y^{(i)})}\right)$$

Empirical risk minimization:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \ell(y^{(i)}, F(x^{(i)}, w))$$

Logistic Regression

Linear regression:

Logistic regression:

Logistic Regression

Linear regression:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2$$

Logistic regression:

Logistic Regression

Linear regression:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)}\underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2$$

Logistic regression:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)}\underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})\right)$$

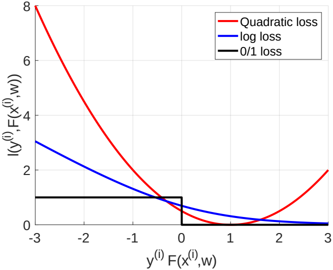
Logistic Regression

Linear regression:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})^2$$

Logistic regression:

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},w)})\right)$$



Optimization:

Optimization:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & f_0(\boldsymbol{w}) \\ \text{s.t.} & f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\} \end{array}$$

Optimization:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & f_0(\boldsymbol{w}) \\ \text{s.t.} & f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\} \end{array}$$

Solution:

Optimization:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & f_0(\boldsymbol{w}) \\ \text{s.t.} & f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\} \end{array}$$

Solution:

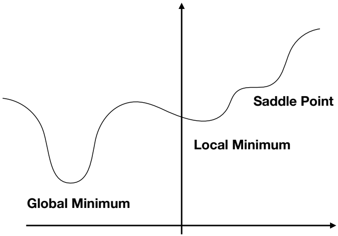
Solution $\boldsymbol{w}^*$ has smallest value $f_0(\boldsymbol{w}^*)$ among all values that satisfy constraints

Optimization:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & f_0(\boldsymbol{w}) \\ \text{s.t.} \quad & f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\} \end{aligned}$$

Solution:

Solution $\boldsymbol{w}^*$ has smallest value $f_0(\boldsymbol{w}^*)$ among all values that satisfy constraints



Optimization: Descent algorithm

Optimization: Descent algorithm

- Start with some guess $\boldsymbol{w}$

Optimization: Descent algorithm

- Start with some guess $\boldsymbol{w}$
- Iterate k = 1, 2, 3, ...

Optimization: Descent algorithm

- Start with some guess $\boldsymbol{w}$
- Iterate k = 1, 2, 3, ...
  - Select direction $\boldsymbol{d}_k$ and stepsize α_k

Optimization: Descent algorithm

- Start with some guess $\boldsymbol{w}$
- Iterate k = 1, 2, 3, ...
  - Select direction $\boldsymbol{d}_k$ and stepsize α_k
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_k \boldsymbol{d}_k$

Optimization: Descent algorithm

- Start with some guess $\boldsymbol{w}$
- Iterate k = 1, 2, 3, ...
  - Select direction $\boldsymbol{d}_k$ and stepsize α_k
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_k \boldsymbol{d}_k$
  - Check whether we should stop (e.g., if $\nabla f(\boldsymbol{w}) \approx 0$)

Duality: Recipe for computing dual program

Duality: Recipe for computing dual program

- Bring primal program into standard form

Duality: Recipe for computing dual program

- Bring primal program into standard form
- Assign Lagrange multipliers to a suitable set of constraints

Duality: Recipe for computing dual program

- Bring primal program into standard form
- Assign Lagrange multipliers to a suitable set of constraints
- Subsume all other constrains in $\mathcal{W}$

Duality: Recipe for computing dual program

- Bring primal program into standard form
- Assign Lagrange multipliers to a suitable set of constraints
- Subsume all other constrains in $\mathcal{W}$
- Write down the Lagrangian L

Duality: Recipe for computing dual program

- Bring primal program into standard form
- Assign Lagrange multipliers to a suitable set of constraints
- Subsume all other constrains in $\mathcal{W}$
- Write down the Lagrangian L
- Minimize Lagrangian w.r.t. primal variables s.t. $\boldsymbol{w} \in \mathcal{W}$

Binary SVM:

Binary SVM:

- Linear regression:

Binary SVM:

- Linear regression:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})})^2$$

Binary SVM:

- Linear regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})^2$$

- Logistic regression:

Binary SVM:

- Linear regression:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})})^2$$

- Logistic regression:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})})\right)$$

Binary SVM:

- Linear regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})^2$$

- Logistic regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})\right)$$

- Binary SVM:

Binary SVM:

- Linear regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})^2$$

- Logistic regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})\right)$$

- Binary SVM:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \max\{0, \underbrace{1}_{\text{taskloss}} - y^{(i)} \underbrace{\boldsymbol{w}^\top\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})}\}$$

Binary SVM:

- Linear regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})^2$$

- Logistic regression:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\boldsymbol{w}^T\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})})\right)$$

- Binary SVM:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \max\{0, \underbrace{1}_{\text{taskloss}} - y^{(i)} \underbrace{\boldsymbol{w}^\top\phi(x^{(i)})}_{F(x^{(i)},\boldsymbol{w})}\}$$

- General binary classification:

Binary SVM:

- Linear regression:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})})^2$$

- Logistic regression:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)} \underbrace{\mathbf{w}^T\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})})\right)$$

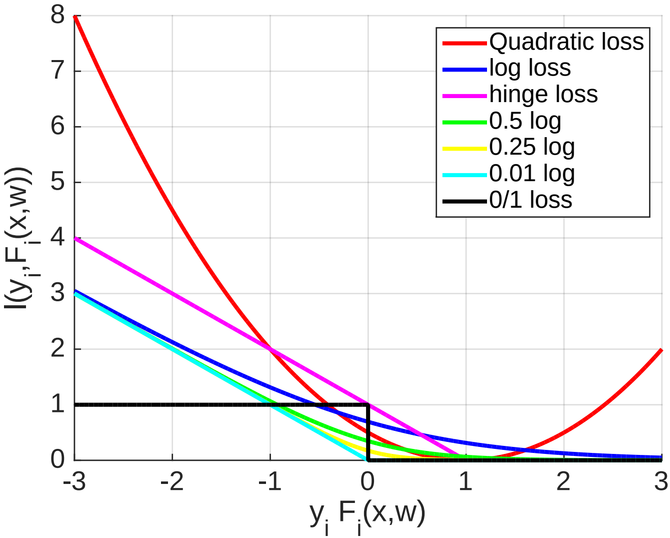
- Binary SVM:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \max\{0, \underbrace{1}_{\text{taskloss}} - y^{(i)} \underbrace{\mathbf{w}^\top\phi(x^{(i)})}_{F(x^{(i)},\mathbf{w})}\}$$

- General binary classification:

$$\min_{\mathbf{w}} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \log\left(1 + \exp\left(\frac{L - y^{(i)}\mathbf{w}^T\phi(x^{(i)})}{\epsilon}\right)\right)$$

Binary SVM



Quadratic loss
log loss
hinge loss
0.5 log
0.25 log
0.01 log
0/1 loss
l(y_i,F_i(x,w))
y_i F_i(x,w)
8
7
6
5
4
3
2
1
0
-3
-2
-1
0
1
2
3

Multiclass classification: How to classify between K classes?

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$
- Use K weight vectors $\boldsymbol{w}_{(y)}$

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$
- Use K weight vectors $\boldsymbol{w}_{(y)}$
- How to parameterize the multinomial distribution?

$$p(y = k|x^{(i)}) = \frac{\exp \boldsymbol{w}_{(k)}^\top \phi(x^{(i)})}{\sum_{j\in\{0,1,\ldots,K-1\}} \exp \boldsymbol{w}_{(j)}^\top \phi(x^{(i)})}$$

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$
- Use K weight vectors $\boldsymbol{w}_{(y)}$
- How to parameterize the multinomial distribution?

$$p(y = k|x^{(i)}) = \frac{\exp \boldsymbol{w}_{(k)}^\top \phi(x^{(i)})}{\sum_{j\in\{0,1,\ldots,K-1\}} \exp \boldsymbol{w}_{(j)}^\top \phi(x^{(i)})}$$

- We are using one parameter vector $\boldsymbol{w}_{(y)}$ per class

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$
- Use K weight vectors $\boldsymbol{w}_{(y)}$
- How to parameterize the multinomial distribution?

$$p(y = k|x^{(i)}) = \frac{\exp \boldsymbol{w}_{(k)}^\top \phi(x^{(i)})}{\sum_{j\in\{0,1,\ldots,K-1\}} \exp \boldsymbol{w}_{(j)}^\top \phi(x^{(i)})}$$

- We are using one parameter vector $\boldsymbol{w}_{(y)}$ per class
- Maximizing the likelihood as before

Multiclass classification: How to classify between K classes?

- Use a multinomial distribution over $y \in \{0, 1, \ldots, K-1\}$
- Use K weight vectors $\boldsymbol{w}_{(y)}$
- How to parameterize the multinomial distribution?

$$p(y = k | x^{(i)}) = \frac{\exp \boldsymbol{w}_{(k)}^\top \phi(x^{(i)})}{\sum_{j \in \{0,1,\ldots,K-1\}} \exp \boldsymbol{w}_{(j)}^\top \phi(x^{(i)})}$$

- We are using one parameter vector $\boldsymbol{w}_{(y)}$ per class
- Maximizing the likelihood as before

$$\arg\max_{\boldsymbol{w}} \prod_{(x^{(i)},y^{(i)}) \in \mathcal{D}} p(y = y^{(i)} | x^{(i)}) = \arg\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)}) \in \mathcal{D}} -\log p(y = y^{(i)} | x^{(i)})$$

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2+\sum_{i\in\mathcal{D}}\left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)},\hat{y}) + \boldsymbol{w}^\top \psi(x^{(i)},\hat{y})}{\epsilon} - \boldsymbol{w}^\top \psi(x^{(i)},y^{(i)})\right)$$

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

- **Logistic regression**

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

- **Logistic regression**
- **Binary SVM**

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

- **Logistic regression**
- **Binary SVM**
- **Multiclass regression**

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

- **Logistic regression**
- **Binary SVM**
- **Multiclass regression**
- **Multiclass SVM**

Deep Learning:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \left(\epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\boldsymbol{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, y^{(i)}) \right)$$

How to get to

- **Logistic regression**
- **Binary SVM**
- **Multiclass regression**
- **Multiclass SVM**
- **Deep Learning**

Deep Learning:

What function $F(\boldsymbol{w}, x, y) \in \mathbb{R}$ to choose? ($y \in \{1, \ldots, K\}$)

Deep Learning:

What function $F(\boldsymbol{w}, x, y) \in \mathbb{R}$ to choose? ($y \in \{1, \ldots, K\}$)

- Choose any differentiable composite function

$$F(\boldsymbol{w}, x, y) = f_1(\boldsymbol{w}_1, y, f_2(\boldsymbol{w}_2, f_3(\ldots f_n(\boldsymbol{w}_n, x)\ldots))) \in \mathbb{R}$$

Deep Learning:

What function $F(\boldsymbol{w}, x, y) \in \mathbb{R}$ to choose? ($y \in \{1, \ldots, K\}$)

- Choose any differentiable composite function

$$F(\boldsymbol{w}, x, y) = f_1(\boldsymbol{w}_1, y, f_2(\boldsymbol{w}_2, f_3(\ldots f_n(\boldsymbol{w}_n, x)\ldots))) \in \mathbb{R}$$

- More generally: functions can be represented by an acyclic graph (computation graph)

Deep Learning:

$$F(\boldsymbol{w}, x, y) = f_1(w_1, y, f_2(w_2, f_3(\ldots)))$$

Deep Learning:

$$F(\boldsymbol{w}, x, y) = f_1(w_1, y, f_2(w_2, f_3(\ldots)))$$

Nodes are

Deep Learning:

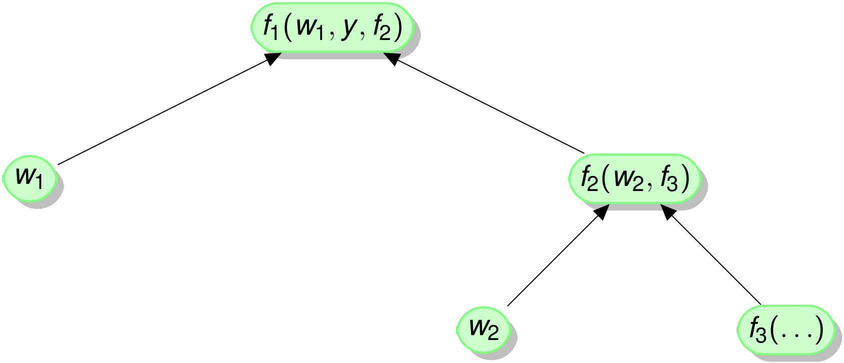
$$F(\boldsymbol{w}, x, y) = f_1(w_1, y, f_2(w_2, f_3(\ldots)))$$

Nodes are weights, data, and functions:

Deep Learning:

$$F(\boldsymbol{w}, x, y) = f_1(w_1, y, f_2(w_2, f_3(\ldots)))$$

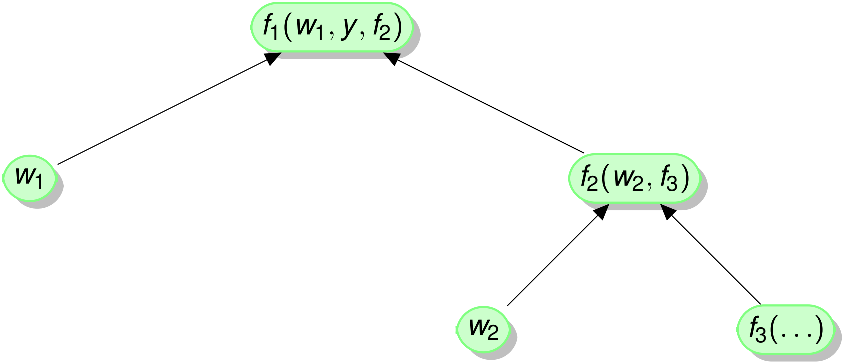
Nodes are weights, data, and functions:



Deep Learning:

$$F(\boldsymbol{w}, x, y) = f_1(w_1, y, f_2(w_2, f_3(\ldots)))$$

Nodes are weights, data, and functions:



Internal representation used by deep net packages.

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers
- Convolutions

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers
- Convolutions
- Rectified linear units (ReLU): $\max\{0, x\}$

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers
- Convolutions
- Rectified linear units (ReLU): $\max\{0, x\}$
- Maximum-/Average pooling

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers
- Convolutions
- Rectified linear units (ReLU): $\max\{0, x\}$
- Maximum-/Average pooling
- Soft-max layer

Deep Learning:

What are the individual functions/layers f_1, f_2 etc.?

- Fully connected layers
- Convolutions
- Rectified linear units (ReLU): $\max\{0, x\}$
- Maximum-/Average pooling
- Soft-max layer
- Dropout

Ensemble methods:

Ensemble methods:

- Train multiple classifiers on subsets of the data

Ensemble methods:

- Train multiple classifiers on subsets of the data
- Average the results

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression
- Binary SVM

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression
- Binary SVM
- Multiclass regression

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression
- Binary SVM
- Multiclass regression
- Multiclass SVM

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression
- Binary SVM
- Multiclass regression
- Multiclass SVM
- Deep Learning

Structured Prediction:

$$\min_{\boldsymbol{w}} \frac{C}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i\in\mathcal{D}} \epsilon \ln \sum_{\hat{\boldsymbol{y}}} \exp \frac{L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}) + F(\boldsymbol{w}, x^{(i)}, \hat{\boldsymbol{y}})}{\epsilon} - F(\boldsymbol{w}, x^{(i)}, \boldsymbol{y}^{(i)})$$

Attention:

- Scoring function $F(\boldsymbol{w}, x, y)$
- Loss function (log-loss, hinge-loss)
- Taskloss L

How to get to

- Binary Logistic regression
- Binary SVM
- Multiclass regression
- Multiclass SVM
- Deep Learning
- Structured prediction

Prediction: Inference (how to find the highest scoring configuration):

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming
- Integer linear program

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming
- Integer linear program
- Linear programming relaxation

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming
- Integer linear program
- Linear programming relaxation
- Message passing

Prediction: Inference (how to find the highest scoring configuration):

$$y^* = \arg\max_{\hat{y}} F(\boldsymbol{w}, x, y)$$

Structured Prediction

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_r f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming
- Integer linear program
- Linear programming relaxation
- Message passing
- Graph-cut

Learning theory:

Why does learning on the training set generalize?

Learning theory:

Why does learning on the training set generalize?

Why does independently solving the homework help in the midterm?

We are sure you'll all make it!