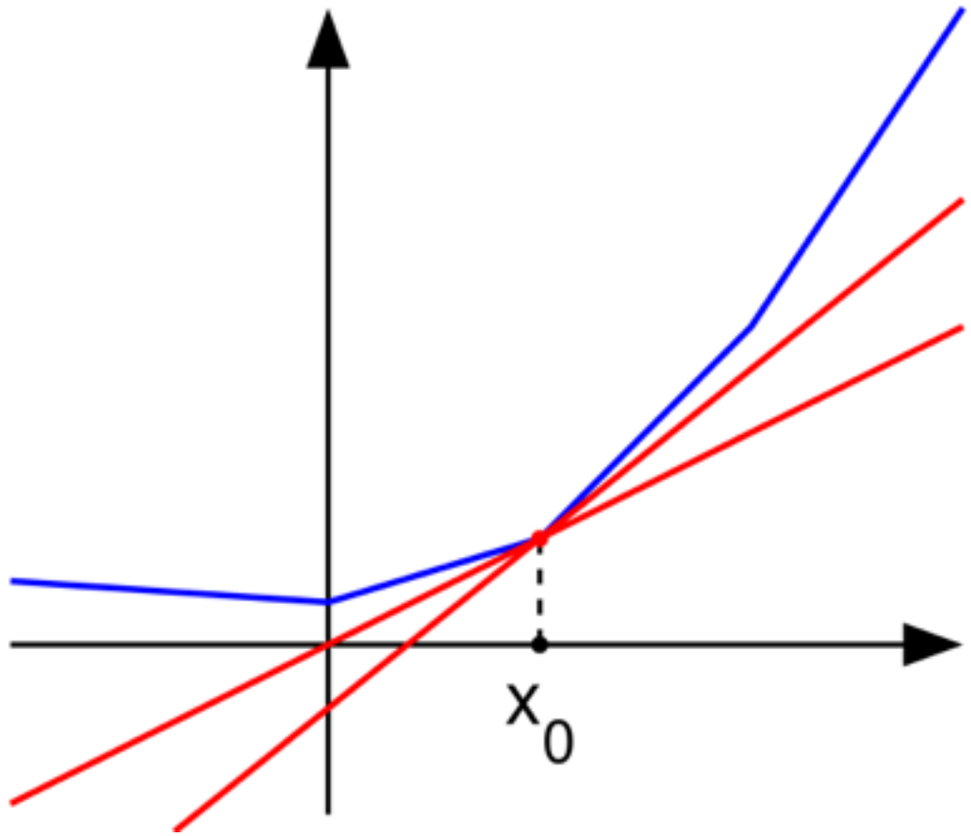
COMPUTER SCIENCE
INDIANA UNIVERSITY
School of Informatics and Computing
Bloomington

Multiclass classification and review

Reminders/Comments

- Assignment 2 due this Wednesday
- Poisson regression can require some careful parameter tuning; I suggest you look up some tricks
  - if you cannot get it to outperform linear regression, explain why
  - see if you can get it to outperform linear regression on some smaller sets of the data
- What happens if a function is not differentiable at a point?
  - Sub-gradient minimization



Discriminative versus generative

- Discriminative: learn p(y | x) directly
- Generative: learn p(x | y) and p(y) to compute p(y | x)
- Previous question: is this related to the distinction between MAP and ML?
  - only related in that both use Bayes rule
  - MAP and ML can be used to estimate p(y | x) or p(x | y) and p(y)

Generative: naive Bayes

- To sample (x,y) from generative model p(x,y) = p(x | y) p(y)
  - Sample y from p(y)
  - Then sample x from p(x | y)

- Why would you do this and/or why use the generative approach? Let's use the faces example again
  - Could sample faces from p(x | y) to see what your model produces
  - Could answer additional questions about the features, such as average eye size in population
  - Could depict average face, within a gender and across genders
  - Question: how would you do it within a gender?
  - Question: how would you do it across genders?

Multi-class and Multi-label

- Multi-class: have multiple classes, with each instance only in one class
  - e.g., a person can only have one blood type

- Multi-label: have multiple classes, where each instance can have multiple class labels
  - e.g., a newspaper article can be a sports article and medicine article

Exercise: problem representation for classification

- What if have many many classes (e.g., image classification)?
  - Is image classification multi-class or multi-label?
- Learn multiple logistic regression models
  - what are the issues with this approach for multi-class? (called one-vs-rest for multi-class)
  - what are the issues with this approach for multi-label? (called binary relevance for multi-label, different from one-vs-rest)
- What other techniques can we use for multi-class and multi-label?
  - instance-based approaches more naturally extend (k-NN)
  - for multiclass: ‘one vs one’ learn k (k-1)/2 binary classifiers
  - for multiclass: multinomial logistic regression

Extending binary to multiclass

- Need to enforce that only one class is predicted

- Multinomial distribution is probability of n successes in k Bernoulli trials

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ \\ 0 & \text{otherwise,} \end{cases}$$

- Multinomial classification corresponds to learning several logistics regression (Bernoulli) models, and constraining that their probabilities sum to 1

- Corresponding transfer uses sigmoid, with normalization —> this is called the softmax transfer

Multinomial logistic regression

- y = [0 1 0 0] means that instance is in class 2 out of 4 classes
- Let k be the number of classes, n = 1 for 1 success

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{y_1! \dots y_k!} p(y_1 = 1|\mathbf{x})^{y_1} \dots, p(y_k|\mathbf{x})^{y_k}$$

- The transfer (inverse of link) is the softmax transfer

$$\begin{aligned}\text{softmax}(\mathbf{x}^\top \mathbf{W}) &= \left[\frac{\exp(\mathbf{x}^\top \mathbf{w}_1)}{\sum_{j=1}^k \exp(\mathbf{x}^\top \mathbf{w}_j)}, \dots, \frac{\exp(\mathbf{x}^\top \mathbf{w}_k)}{\sum_{j=1}^k \exp(\mathbf{x}^\top \mathbf{w}_j)} \right] \\ &= \left[\frac{\exp(\mathbf{x}^\top \mathbf{w}_1)}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})}, \dots, \frac{\exp(\mathbf{x}^\top \mathbf{w}_k)}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})} \right]\end{aligned}$$

Relation to logistic regression

- For k= 2, y = [0 1] or y = [1 0]

$$\begin{aligned}\text{softmax}(\mathbf{x}^\top \mathbf{W}) &= \left[\frac{\exp(\mathbf{x}^\top \mathbf{w}_1)}{\sum_{j=1}^k \exp(\mathbf{x}^\top \mathbf{w}_j)}, \cdots, \frac{\exp(\mathbf{x}^\top \mathbf{w}_k)}{\sum_{j=1}^k \exp(\mathbf{x}^\top \mathbf{w}_j)}\right] \\ &= \left[\frac{\exp(\mathbf{x}^\top \mathbf{w}_1)}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})}, \cdots, \frac{\exp(\mathbf{x}^\top \mathbf{w}_k)}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})}\right]\end{aligned}$$

$$\mathbf{W} = [\mathbf{w},\ \mathbf{0}] \qquad \begin{aligned} p(y=1|\mathbf{x}) &= \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})} \\ &= \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\exp(\mathbf{x}^\top \mathbf{W}) + \exp(\mathbf{x}^\top \mathbf{0})} \\ &= \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\exp(\mathbf{x}^\top \mathbf{W}) + 1} \\ &= \sigma(\mathbf{x}^\top \mathbf{w}). \end{aligned}$$

Relation to logistic regression…

- In general, setting w_k to zero is a convention
  - could choose any of the classes, e.g., could learn p(y = 0 | x)

- Normalization enforces constraint on w_k (or on one of the classes), so setting w_k = 0 causes other w_i to pivot around it and learn the correctly normalized probabilities

  - **Exercise**: show that setting w_k = 0 is also required to ensure that the softmax transfer is invertible

$$\mathbf{W} = [\mathbf{w},\ \mathbf{0}] \qquad p(y=1|\mathbf{x}) = \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W})}$$

$$= \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\exp(\mathbf{x}^\top \mathbf{W}) + \exp(\mathbf{x}^\top \mathbf{0})}$$

$$= \frac{\exp(\mathbf{x}^\top \mathbf{w})}{\exp(\mathbf{x}^\top \mathbf{W}) + 1}$$

$$= \sigma(\mathbf{x}^\top \mathbf{w}).$$

Main take-away

- For multinomial logistic regression, we have
  - p(y | x) is a multinomial distribution
  - the corresponding (invertible) transfer is the softmax transfer with w_k = 0 to ensure invertibility
  - the corresponding potential or log-normalizer is <1, exp(xW)>

- Using our the minimization obtained for our generalized linear models, we can plug in the transfer and potential to get

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k} : \mathbf{W}_{:k} = \mathbf{0}} \log \left(\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W}) \right) - \mathbf{x}^\top \mathbf{W} \mathbf{y}$$

with gradient

$$\nabla \left(\log \left(\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W}) \right) - \mathbf{x}^\top \mathbf{W} \mathbf{y} \right) = \frac{\exp(\mathbf{x}^\top \mathbf{W})^\top \mathbf{x}^\top}{\mathbf{1}^\top \exp(\mathbf{x}^\top \mathbf{W}} - \mathbf{x} \mathbf{y}^\top .$$

Out-of-sample prediction

- For a new sample, use softmax(xW) = [p(y=1 l x), p(y = 2l x), …, p(y = k l x)]

- Example: softmax(xW) = [0.1 0.2 0.6 0.1] suggests we should pick class y = 3, since it has the highest probability

Class feedback / Pop quiz

- Any anonymous feedback is highly welcome

- A couple of questions to track how you're doing

Representation learning

- Generalized linear models enabled many p(y l x) distributions
  - Underneath, still learning a linear representation for E[y l x], which may not have enough representation capacity

- Approach we discussed earlier: augment current features x using polynomials

- There are many strategies to augmenting x
  - fixed representations, like polynomials, wavelets
  - learned representations, like neural networks and matrix factorization

Polynomial representations

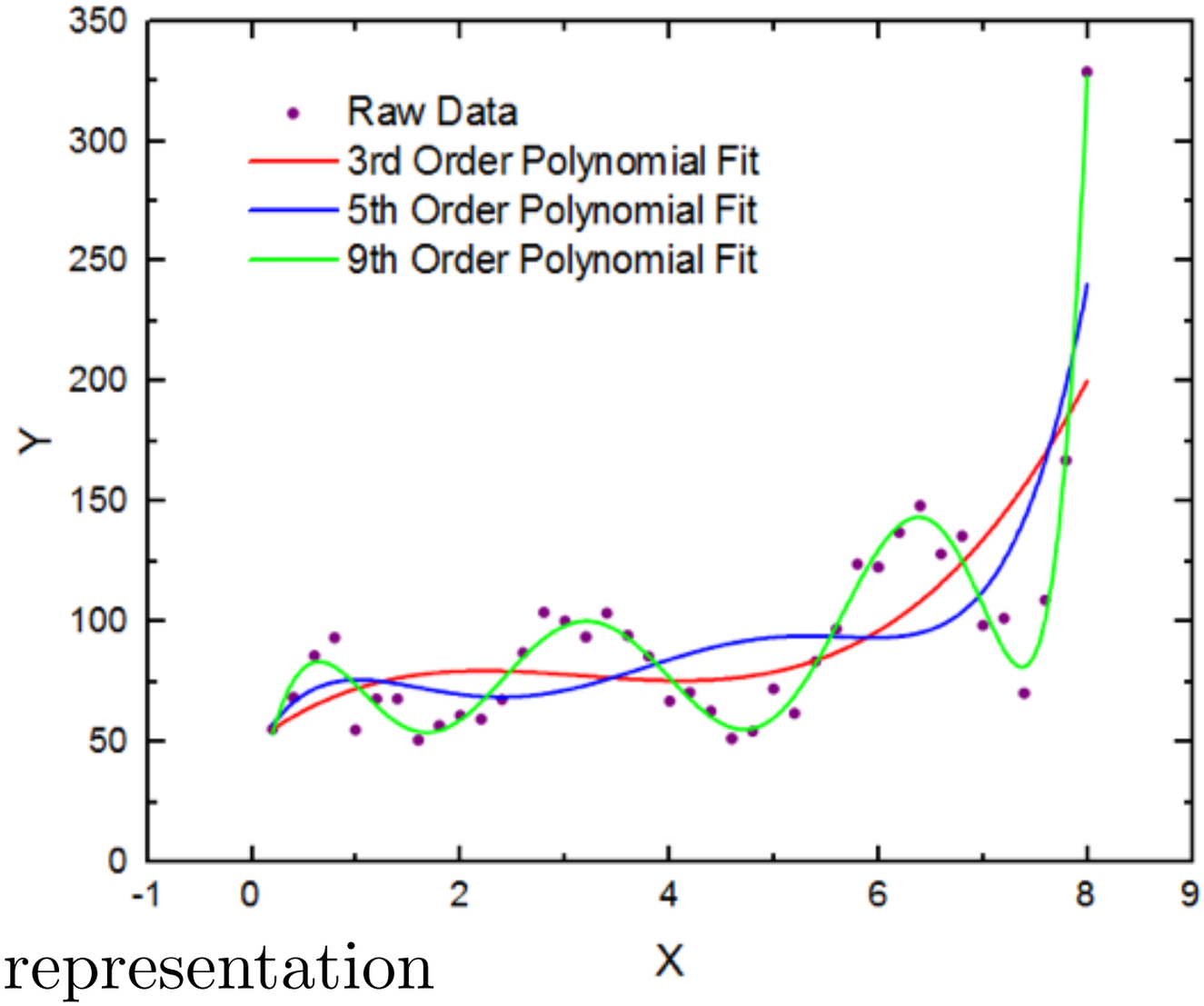
- Using Taylor series, many functions (any function we care about mostly) can be represented as a (high-order) polynomial

$$\mathbf{x} \rightarrow$$

$$\text{2nd-order polynomial}(\mathbf{x}) =$$

$$w_6 x_1^2 + w_5 x_2^2 + w_4 x_1 x_2$$

$$+ w_2 x_2 + w_1 x_1 + w_0$$



$$\mathbf{x}^\top \mathbf{w} = g(E[y|\mathbf{x}])$$

transformed to more powerful representation

$$\text{polynomial}(\mathbf{x})^\top \mathbf{w} = g(E[y|\mathbf{x}])$$

Radial basis function network

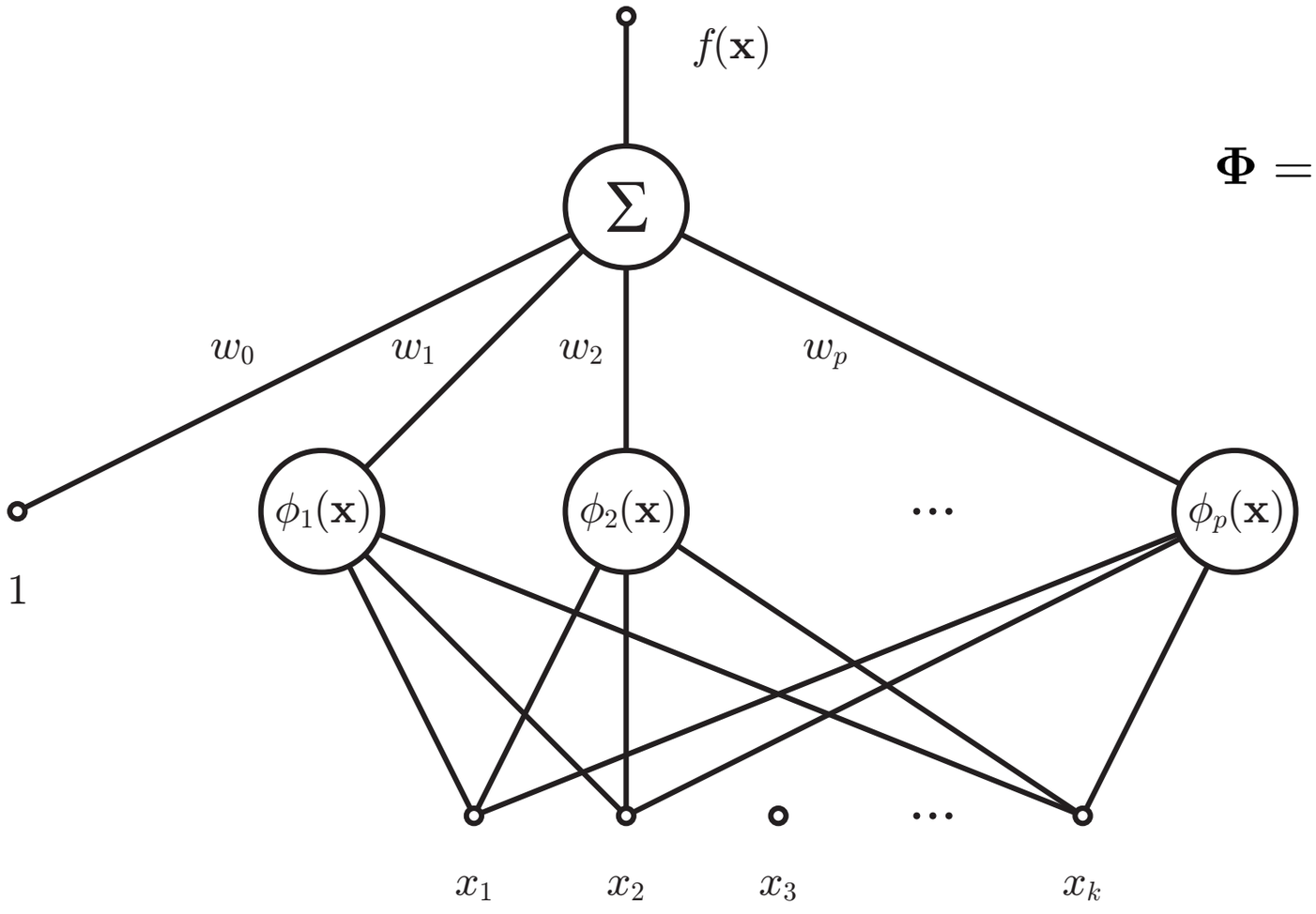


Figure 7.1: Radial basis function network.

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_p(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & & \\ \vdots & & \ddots & \\ & & & \\ \phi_0(\mathbf{x}_n) & & & \phi_p(\mathbf{x}_n) \end{bmatrix}$$

e.g., $\phi_j(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2\sigma_j^2}}$,

$$\begin{aligned} f(\mathbf{x}) &= w_0 + \sum_{j=1}^{p} w_j \phi_j(\mathbf{x}) \\ &= \sum_{j=0}^{p} w_j \phi_j(\mathbf{x}) \end{aligned}$$