JOHNS HOPKINS UNIVERSITY

EN.600.475 Machine Learning

Logistic Regression

Raman Arora
Lecture 10
February 27, 2017

- Stochastic Gradient Descent
- Regularized Logistic Regression

Slides credit: Greg Shakhnarovich 1

Review

Review: classification theory

- Loss of choice: 0/1 loss $L_{0/1}(\widehat{y}, y) = 0$ if $\widehat{y} = y$, 1 otherwise
- Goal in learning $h : \mathcal{X} \to \{1, \ldots, C\}$: minimize risk $R(h) \;=\; E_{\mathbf{x},y}\left[L(h(\mathbf{x}), y)\right]$
- Optimal classifier:

$$h(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\, p\left(y = c \,|\, \mathbf{x}\right).$$

- Log-odds criterion:

$$h(\mathbf{x}) \;= c^* \quad \Leftrightarrow \quad \log \frac{p\left(y = c^* \,|\, \mathbf{x}\right)}{p\left(y = c \,|\, \mathbf{x}\right)} \geq 0 \quad \forall c$$

Review: optimal prediction in supervised learning

- We now have identified an optimal predictor for both core supervised learning tasks
- Can write them down based on (alas, unknown) $p(\mathbf{x}, y)$
- Regression: optimal regressor

$$\widehat{y} \;=\; E\left[y|\mathbf{x}\right]$$

- Classification: optimal classifier

$$\widehat{y} \;=\; \operatorname*{argmax}_{c} p\left(y = c\,|\,\mathbf{x}\right)$$

- In both cases, even the optimal classifiers suffer from error due to inherent uncertainty in $p(\mathbf{x}, y)$

Review: logistic regression

- Directly model log-odds as a function of $\mathbf{x}$

$$\log\frac{p\left(y = 1\,|\,\mathbf{x}\right)}{p\left(y = 0\,|\,\mathbf{x}\right)} \;=\; f(\boldsymbol{\phi}(\mathbf{x}); \mathbf{w}) \;=\; 0.$$

- After some algebra:

$$p\left(y = 1\,|\,\mathbf{x}\right) \;=\; \frac{1}{1 + \exp\left(-f(\boldsymbol{\phi}(\mathbf{x}); \mathbf{w})\right)}$$

- For linear $\boldsymbol{\phi}$, and $f(\boldsymbol{\phi}(\mathbf{x})) \;=\; \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x})$ we get

$$p\left(y = 1\,|\,\mathbf{x}\right) = 1/\left(1 + \exp(-w_0 - \mathbf{w} \cdot \mathbf{x})\right)$$

The logistic model

- We can model the (unknown) decision boundary directly:

$$\log \frac{p\,(y=1\,|\,\mathbf{x})}{p\,(y=0\,|\,\mathbf{x})} \;=\; w_0 + \mathbf{w}\cdot\mathbf{x} \;=\; 0.$$

- Since $p\,(y=1\,|\,\mathbf{x}) = 1 - p\,(y=0\,|\,\mathbf{x})$, we have (after exponentiating):

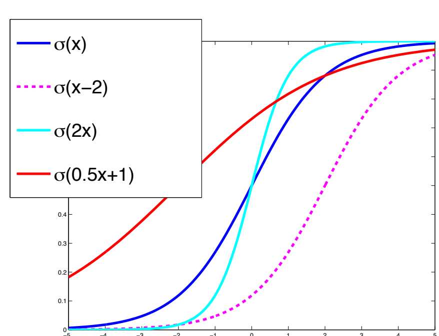
$$\begin{aligned} \frac{p\,(y=1\,|\,\mathbf{x})}{1-p\,(y=1\,|\,\mathbf{x})} \;&=\; \exp(w_0 + \mathbf{w}\cdot\mathbf{x}) \;=\; 1 \\ \Rightarrow \frac{1}{p\,(y=1\,|\,\mathbf{x})} \;&=\; 1+\exp(-w_0 - \mathbf{w}\cdot\mathbf{x}) \;=\; 2 \\ \Rightarrow p\,(y=1\,|\,\mathbf{x}) \;&=\; \frac{1}{1+\exp(-w_0 - \mathbf{w}\cdot\mathbf{x})} \;=\; \frac{1}{2}. \end{aligned}$$

The logistic function

$$p\,(y=1\,|\,\mathbf{x}) \;=\; \frac{1}{1+\exp(-w_0 - \mathbf{w}\cdot\mathbf{x})}$$

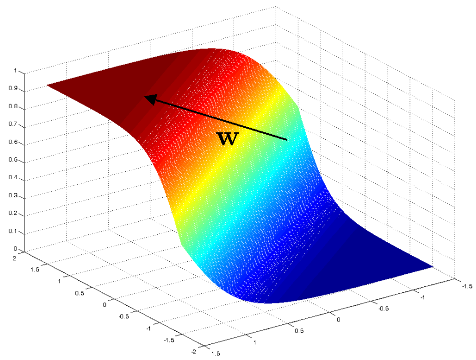
- The logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$:
  For any x, $0 \le \sigma(x) \le 1$;
  Monotonic, $\sigma(-\infty) = 0$, $\sigma(+\infty) = 1$

- $\sigma(0) = 1/2$. To shift the crossing to an arbitrary z: $\sigma(x-z)$.
- To change the "slope": $\sigma(ax)$.



Logistic function in $\mathbb{R}^d$

- What if $\mathbf{x} \in \mathbb{R}^d = [x_1 \ldots x_d]$?
- $\sigma(w_0 + \mathbf{w} \cdot \mathbf{x})$ is a scalar function of a scalar variable $w_0 + \mathbf{w} \cdot \mathbf{x}$.

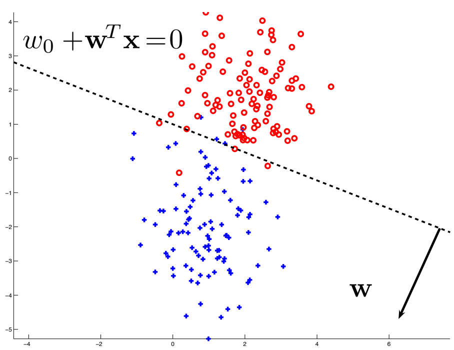


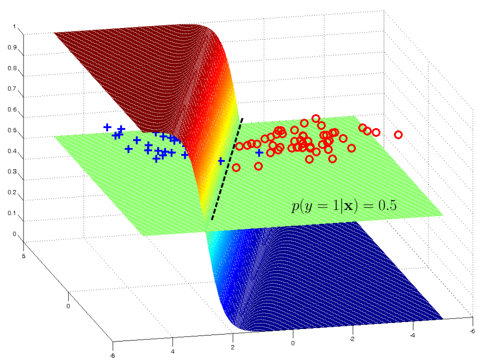
- the direction of $\mathbf{w}$ determines orientation;
- w_0 determines the location;
- $\|\mathbf{w}\|$ determines the slope.

Logistic regression: decision boundary

$$p(y = 1 \mid \mathbf{x}) = \sigma(w_0 + \mathbf{w} \cdot \mathbf{x}) = 1/2 \Leftrightarrow w_0 + \mathbf{w} \cdot \mathbf{x} = 0$$

- With linear logistic model we get a linear decision boundary.





Likelihood under the logistic model

- Regression: observe values, measure residuals under the model.
- Logistic regression: observe labels, measure their probability under the model.

$$\begin{aligned} p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right) &= \begin{cases} \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i) & \text{if } y_i = 1, \\ 1 - \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i) & \text{if } y_i = 0 \end{cases} \\ &= \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i)^{y_i}\left(1 - \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right)^{1-y_i}. \end{aligned}$$

- The log-likelihood of $\mathbf{w}$:

$$\begin{aligned} \log p(Y|X;\mathbf{w}) &= \sum_{i=1}^{N} \log p\left(y_i \,|\, \mathbf{x}_i;\, \mathbf{w}\right) \\ &= \sum_{i=1}^{N} y_i \log \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i) + (1 - y_i)\log\left(1 - \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right) \end{aligned}$$

The maximum likelihood solution

$$\log p(Y|X;\mathbf{w}) = \sum_{i=1}^{N} y_i \log \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i) + (1 - y_i)\log\left(1 - \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right)$$

- Setting the derivatives to zero, we get

$$\frac{\partial}{\partial w_0}\log p(Y|X;\mathbf{w}) = \sum_{i=1}^{N}\left(y_i - \sigma\left(w_0 + \mathbf{w}\cdot\mathbf{x}_i\right)\right) = 0;$$

$$\frac{\partial}{\partial w_j}\log p(Y|X;\mathbf{w}) = \sum_{i=1}^{N}\left(y_i - \sigma\left(w_0 + \mathbf{w}\cdot\mathbf{x}_i\right)\right) x_{ij} = 0.$$

- We can treat $y_i - p\left(y_i \,|\, \mathbf{x}_i\right) = y_i - \sigma(w_0 + \mathbf{w}\cdot\mathbf{x}_i)$ as the *prediction error* of the model on $\mathbf{x}_i, y_i$.
- As with linear regression: prediction errors are uncorrelated with any linear function of the data.

Gradient ascent

- We can cycle through the examples, accumulating the gradient, and then applying the accumulated value to form an update

$$\begin{aligned}\mathbf{w}_{new} \;:=\; & \mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} \log p(X; \mathbf{w}) \\ = \; & \mathbf{w} + \eta \sum_{i=1}^{N} \left(y_i - \sigma(w_0 + \mathbf{w} \cdot \mathbf{x}_i)\right) \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}\end{aligned}$$

- Remember: need to choose η rather carefully:
  - Too small $\Rightarrow$ slow convergence;
  - Too large: $\Rightarrow$ overshoot and osillation.

Logistic regression

Newton-Raphson

- The *Newton-Raphson* algorithm: approximate the local shape of $\log p$ as a quadratic function.

$$\mathbf{w}_{new} \;:=\; \mathbf{w} + \mathbf{H}^{-1} \frac{\partial}{\partial \mathbf{w}} \log p(X; \mathbf{w}),$$

where $\mathbf{H}$ is the *Hessian* matrix of second derivatives:

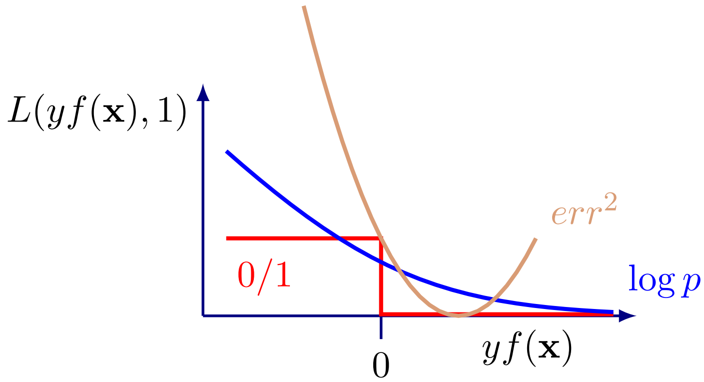
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \log p}{\partial w_0^2} & \frac{\partial^2 \log p}{\partial w_0 w_1} & \cdots & \frac{\partial^2 \log p}{\partial w_0 w_d} \\ \frac{\partial^2 \log p}{\partial w_0 w_1} & \frac{\partial^2 \log p}{\partial w_1^2} & \cdots & \frac{\partial^2 \log p}{\partial w_1 w_d} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 \log p}{\partial w_d w_0} & \frac{\partial^2 \log p}{\partial w_d w_1} & \cdots & \frac{\partial^2 \log p}{\partial w_d^2} \end{bmatrix}$$

Surrogate loss

- Recall that we really want to minimize 0/1 loss
- Instead, we are minimizing the log-loss:

$$\operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i;\, \mathbf{w}) \;=\; \operatorname*{argmin}_{\mathbf{w}} -\sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i\,;\mathbf{w})$$

- This is a *surrogate* loss; we work with it since it is not computationally feasible to optimize the 0/1 loss directly.



Generalized additive models

- As with regression we can extend this framework to arbitrary features (basis functions):

$$p\,(y = 1\,|\,\mathbf{x}) \;=\; \sigma\,(w_0 + \phi_1(\mathbf{x}) + \ldots + \phi_m(\mathbf{x}))\,.$$

- Example: quadratic logistic regression in 2D

$$p\,(y = 1\,|\,\mathbf{x}) \;=\; \sigma\left(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2\right).$$

Generalized additive models

- As with regression we can extend this framework to arbitrary features (basis functions):

$$p\,(y = 1\,|\,\mathbf{x}) \;=\; \sigma\left(w_0 + \phi_1(\mathbf{x}) + \ldots + \phi_m(\mathbf{x})\right).$$

- Example: quadratic logistic regression in 2D

$$p\,(y = 1\,|\,\mathbf{x}) \;=\; \sigma\left(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2\right).$$

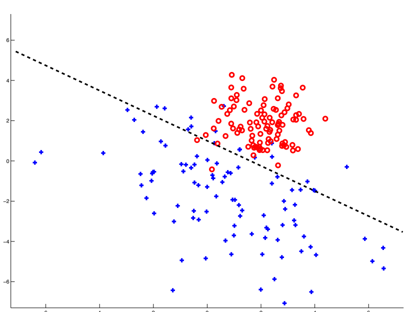
  - Decision boundary of this classifier:

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 = 0,$$

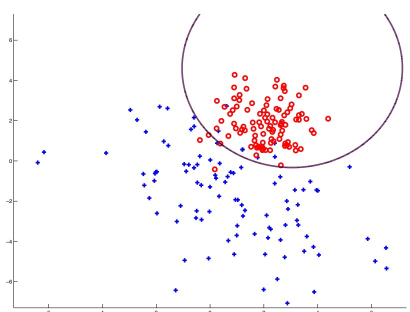
    i.e. it's a quadratic decision boundary.

Logistic regression: 2D example

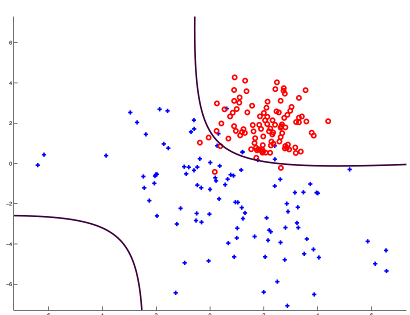
Linear



Quadratic



We can also include $x_1 x_2$:



Roadmap

- Last lecture:
  - Linear classifiers, and a couple of surrogate loss functions
  - Learning algorithm (gradient descent)
- Today:
  - Another learning algorithm: stochastic gradient descent
  - Regularized logistic regression (+ another view of regularization)
  - Non-linear predictors: decision trees

Review: gradient descent

$$\frac{\partial}{\partial \mathbf{w}} \log p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right) \;=\; \left[y_i - \sigma(\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}_i))\right] \boldsymbol{\phi}(\mathbf{x}_i)$$

- Initialize $\mathbf{w}^{(t)} = \mathbf{0}$
- Updates until convergence:

$$\mathbf{w}^{(t+1)} \;:=\; \mathbf{w}^{(t)} \;+\; \eta \sum_{i=1}^{N} \frac{\partial}{\partial \mathbf{w}} \log p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right)$$

- Cost of a single update: computing gradient on all N examples (an *epoch*)

Stochastic gradient descent: intuition

- Computing gradient on all N examples is expensive and may be wasteful
- Many data points provide similar information
- Idea: present examples one at a time, and pretend that the gradient on the entire set is the same as gradient on one example
- Formally: estimate gradient of the loss L

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial \mathbf{w}}L(y_i,\mathbf{x}_i;\mathbf{w}) \approx \frac{\partial}{\partial \mathbf{w}}L(y_t,\mathbf{x}_t;\mathbf{w})$$

Stochastic gradient descent

- An incremental algorithm:
  - Present examples $(\mathbf{x}_i, y_i)$ one at a time,
  - Modify $\mathbf{w}$ slightly to increase the log-probability of observed y_i:

$$\mathbf{w} \;:=\; \mathbf{w} + \eta\frac{\partial}{\partial \mathbf{w}}\log p\,(y_i \,|\, \mathbf{x}_i;\mathbf{w})$$

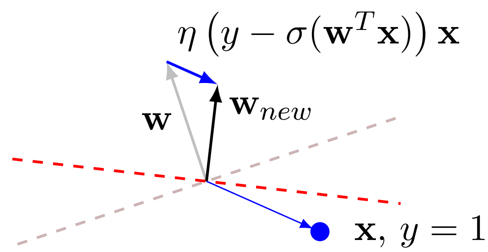
  where the *learning rate* η determines how "slightly".
- Epoch (full pass through data) contains N updates instead of one
- Good practice: shuffle the data

Stochastic gradient descent

- Linear model (assume $w_0 = 0$)

$$\begin{aligned}\mathbf{w}_{new} \; := \; & \mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} \log p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right) \\ = \; & \mathbf{w} + \eta \left(y_i - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \mathbf{x}\end{aligned}$$

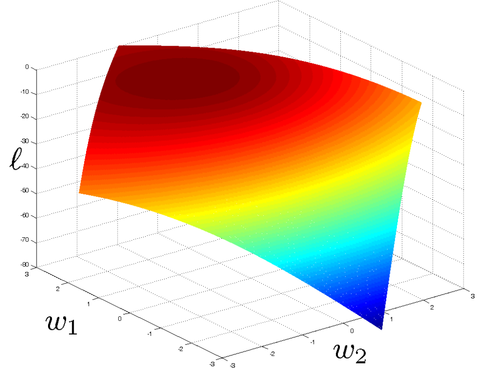
- Contribution of one example:

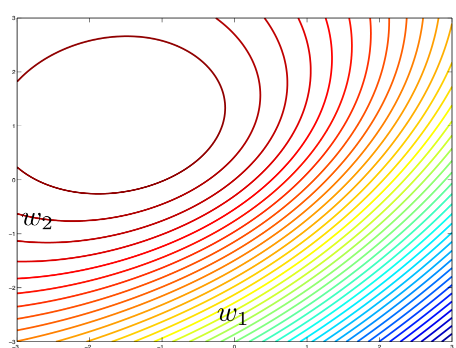


Visualizing the log-likelihood surface

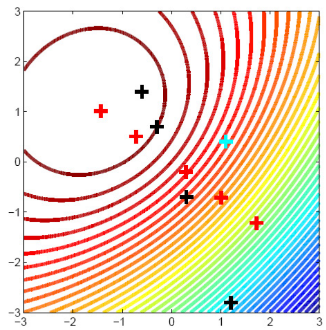
- We will look at a 2D example, and assume $w_0 = 0$, i.e. our model will be $\hat{p}(y = 1|\mathbf{x}) = \sigma(w_1 x_1 + w_2 x_2)$.

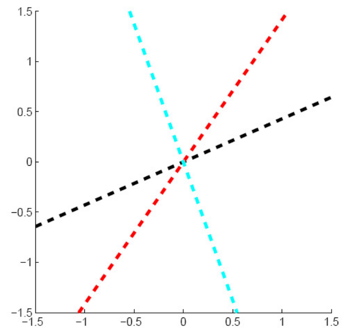
$\log p$ as a function of $\mathbf{w}$ Contour plot: high/low



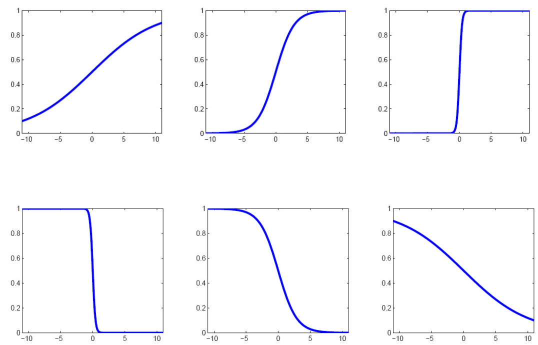


Mapping from boundaries to $\mathbf{w}$





- A line $\alpha\mathbf{w}$ in the parameter space $\Leftrightarrow$ identical decision boundaries of the form $\alpha\mathbf{w}\cdot\mathbf{x} = 0$.
- The sign of α determines the direction.
- Think about the effect of w_0



Overfitting with logistic regression

- We can get the same decision boundary with an infinite number of settings for $\mathbf{w}$.
- When the data are *separable* by $w_0 + \alpha\mathbf{w}\cdot\mathbf{x} = 0$, what's the best choice for α?

$$p(y = 1 \,|\, \mathbf{x}) \;=\; \sigma(w_0 + \alpha\mathbf{w}\cdot\mathbf{x}).$$

- With $\alpha \to \infty$, we have $p(y_i|\mathbf{x}; w_0, \alpha\mathbf{w}) \to 1$.
- With $\alpha = \infty$ there is a continuum of w_0 that reach perfect separation.
- When the data are not separable, similar effect is present but more subtle.

MAP estimation for logistic regression

- Intuition: we may have some belief about the value of $\mathbf{w}$ before seeing any data.
  - E.g., may prefer smaller values of $\|\mathbf{w}\|$ (ignore w_0) Recall our previous motivation for regularizing $\mathbf{w}$!
- A possible prior that captures that belief:

$$p(\mathbf{w}) \;=\; \mathcal{N}\left(\mathbf{w};\, \mathbf{0}, \sigma^2\mathbf{I}\right).$$

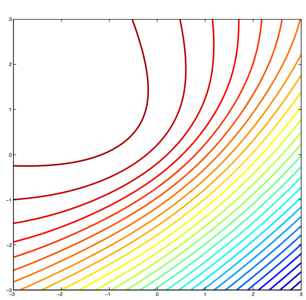
- Instead of $\log p(Y|X;\mathbf{w})$ the objective becomes log-posterior

$$\begin{aligned}\log p(Y|X,\mathbf{w};\sigma) \;&=\; \log p(Y|X,\mathbf{w}) + \log p(\mathbf{w};\sigma)\\ &=\; \sum_{i=1}^{N}\log p\left(y_i\,|\,\mathbf{x}_i,\mathbf{w}\right) - \frac{1}{2\sigma^2}\sum_{j=1}^{d} w_j^2 \;+ \text{const}(\mathbf{w}).\end{aligned}$$

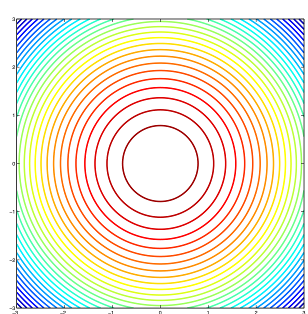
- Setting σ^2 affects the penalty on $\|\mathbf{w}\|$ (cf. λ)

Log posterior surface

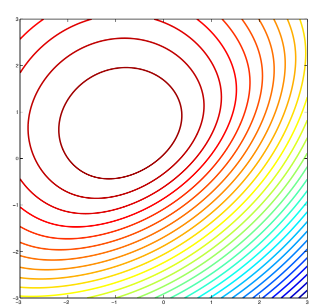
$\log p(Y|X,\mathbf{w})$ $\qquad$ $\log p(\mathbf{w};\sigma)$ $\qquad$ $\log p(Y|X,\mathbf{w};\sigma)$



$+$



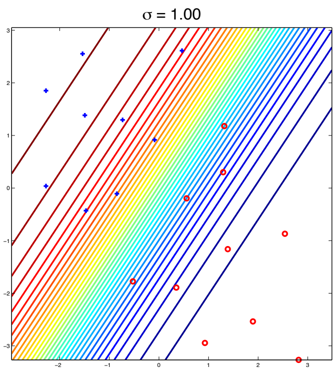
$=$



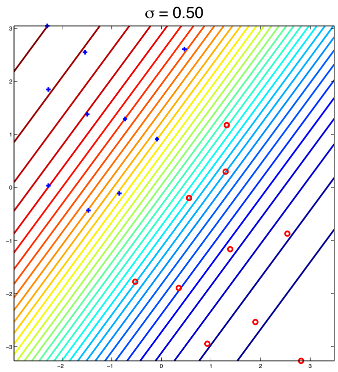
- This is our objective function, and we can find its peak by gradient ascent as before.
  - Need to modify the calculation of gradient and Hessian.

The effect of regularization: separable data

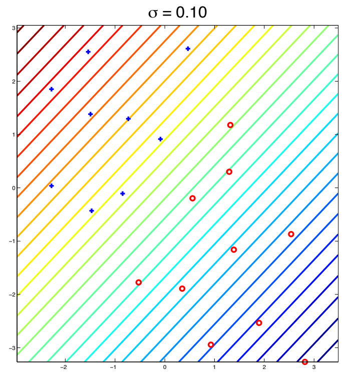
$$\log p(Y|X, \mathbf{w}; \sigma) \;=\; \sum_{i=1}^{N} \log p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right) \;-\; \frac{1}{2\sigma^2}\|\mathbf{w}\|^2$$



$\sigma^2 = 1$



$\sigma^2 = 0.5$

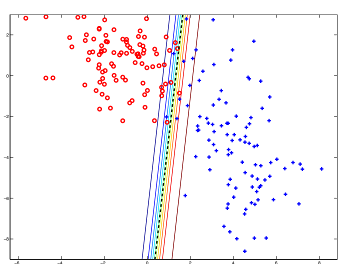


$\sigma^2 = 0.1$

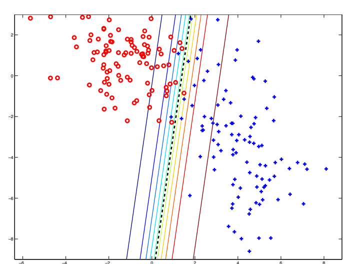
The effect of regularization

$$\log p(Y|X; \mathbf{w}, \sigma) \;=\; \sum_{i=1}^{N} \log p\left(y_i \,|\, \mathbf{x}_i; \mathbf{w}\right) \;-\; \frac{1}{2\sigma^2}\|\mathbf{w}\|^2$$

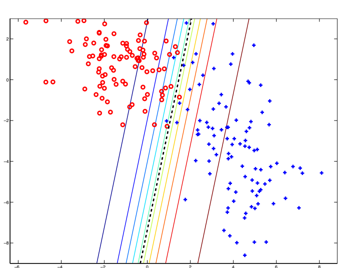
ML



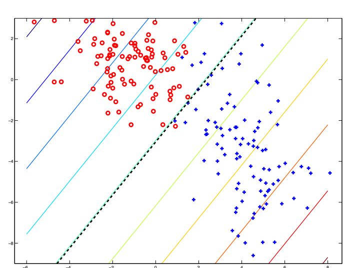
$\sigma^2 = 1$



$\sigma^2 = 0.1$



$\sigma^2 = 0.01$



Softmax

- Logistic regression computes a *score* $f(\mathbf{x}; \mathbf{w}) = \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x})$, which is converted to posterior

$$p\left(y = 1 \,|\, \mathbf{x}\right) \; = \; \frac{\exp f(\mathbf{x}; \mathbf{w})}{1 + \exp f(\mathbf{x}; \mathbf{w})}$$

  (verify that this is equivalent to the form we had before!)
- The *softmax* model: we now have C classes, and C scores $f_c(\mathbf{x}; \mathbf{W}) \; = \; \mathbf{w}_c \cdot \boldsymbol{\phi}(\mathbf{x})$
- To get posteriors from scores, exponentiate and normalize:

$$p\left(y = c \,|\, \mathbf{x}\right) \; = \; \frac{\exp\left(\mathbf{w}_c \cdot \boldsymbol{\phi}(\mathbf{x})\right)}{\sum_{k=1}^{C} \exp\left(\mathbf{w}_k \cdot \boldsymbol{\phi}(\mathbf{x})\right)}$$

  Note: decision on $\mathbf{x}$ depends on all $\mathbf{w}_c$ for $c = 1, \ldots, C$.
- For $C = 2$, this is identical to the logistic regression (homework)
- The boundaries between classes still linear in $\mathbf{w}$ and in $\boldsymbol{\phi}(\mathbf{x})$
- Note: for prediction, do not need to exp. and normalize!

Softmax parameterization

$$p\left(y = c \,|\, \mathbf{x}\right) \; = \; \frac{e^{\mathbf{w}_c \cdot \boldsymbol{\phi}(\mathbf{x}) - a}}{\sum_{k=1}^{C} e^{\mathbf{w}_k \cdot \boldsymbol{\phi}(\mathbf{x}) - a}}$$

- The posteriors are invariant to shifting scores
- A common problem: overflow in $exp(\mathbf{w}_c \cdot \boldsymbol{\phi}(\mathbf{x}))$
- Solution: subtract $a = \max_c \mathbf{w}_c \cdot \boldsymbol{\phi}(\mathbf{x})$
- Then, max score is 0, and the rest are negative; underflow is OK (some may turn to zero)
- Examples: scores $= [1000, 995, 10, 10, 1]$
  Naïve exponentiation: $\approx \; [\infty, \infty, 2.2e4, 2.2e4, 2.7]$
  After shifting dynamic range: $\approx \; [1, 0.007, 0, 0, 0]$