LR + polynomial features = polynomial regression

↳ replace each X_i w/ $\Phi(x_i)$ w/ all terms of degree $0 \ldots p$

e.g. quadratic regression: $\Phi(x_i) = [X_{i1}^2 \quad X_{i1} x_{i2}^2 \quad x_{i1} \quad x_{i2} \quad 1]^T$

fictitious dim.

↳ now, perform linear regression. (or logistic regression)

aside: logistic regression w/ polynomial features = fit posterior probabilities like QDA

↳ fitting logistic function applied to quadratic fn.

- overfitting: as degree p gets higher, easier to overfit

↳ fit to noise rather than data

- polynomials of too high degree can't capture behavior of data outside of train data's domain

---

Weighted LS Regression:

last time: weight different sample points by importance

↳ each sample point X_i has weight w_i

↳ $$\Omega = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix}$$

greater $w \Rightarrow$ trying extra hard to minimize $\|\hat{y}_i - y_i\|_2^2$ for that particular i

full problem: $\min_{\vec{w}} (X\vec{w} - \vec{y})^T \Omega (X\vec{w} - \vec{y}) = \sum_{i=1}^{n} w_i (X_i \cdot \vec{w} - \vec{y}_i)^2$ like 127 HW5!

Set gradient to 0 to find $\vec{w}^*$: $X^T \Omega X \vec{w}^* = X^T \Omega y$

$$\Rightarrow \boxed{w^* = (X^T \Omega X)^{-1} X^T \Omega \vec{y}}$$

for weighted LS regression

Newton's Method: iterative optimization method for smooth functions $J(\vec{v})$

Smooth function = ∇^2 defined everywhere

idea: Start at some $\vec{v}$, want to get closer to min $J(\vec{v})$

1) approximate $J(\vec{v})$ locally by a quadratic function
2) Jump to quadratic's minimum
3) repeat until convergence

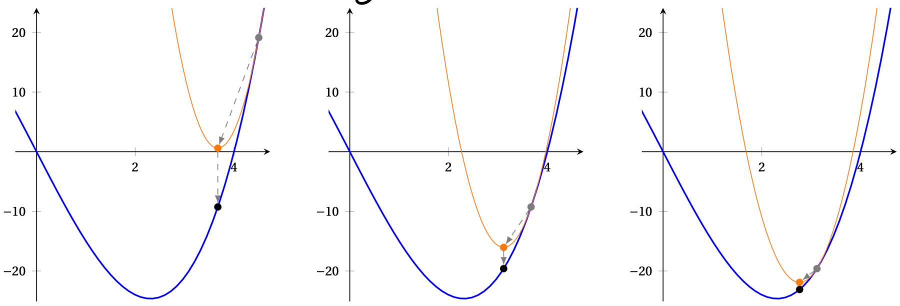

Figure 11.3: Example iterations of Newton's method

Flaws w/ Newton's Method:

- may not find true minimum
- may not find min. at all
  - ↳ e.g. quadratic approx becomes concave ⟶ go opposite dir. to local maximum

Taylor Series of J about $\vec{v}$: $\nabla J(\vec{w}) = \nabla J(\vec{v}) + (\nabla^2 J(w))(\vec{w}-\vec{v}) + O(\|\vec{w}-\vec{v}\|^2)$

Hessian of $J(w)$

find critical point: Set $\nabla J(\vec{w}) = 0$

↳ $\vec{w} = \vec{v} - (\nabla^2 J(w))^{-1} \nabla J(\vec{v})$ ⟵ $\vec{w}$ is critical point

↳ new vector

Newton's Method in Code

```
pick starting point w⃗
repeat until convergence:
    solve linear system (∇²J(w⃗))e⃗ = −∇J(w⃗) for e⃗
    w⃗ = w⃗ + e⃗
```

Warnings:
- doesn't know diff. between minima, maxima, or saddle points
- Starting point must be close enough to desired critical point to converge

Newton's Method	Gradient Descent
- Step size depends on shape of cost fn. ↳ e.g. quadratic $J(\vec{w})$ ⟶ 1 step	- Predetermined step size
- doesn't always go in dir. of steepest descent ↳ instead focuses on best direction	- always descends
- finding $\nabla^2 J$ is expensive	
- doesn't work for nonsmooth fns	

Logistic Regression Cont.

recall: $S(x) = \text{Sigmoid}$, $\frac{\partial S}{\partial x} = S(x)(1-S(x))$, $s_i = s(X_i \cdot \vec{w})$

$$\nabla_w J(\vec{w}) = -\sum_{i=1}^{n} (y_i - s_i) X_i = -X^T(\vec{y} - \vec{s})$$

for newton's method: $\nabla_w J = -X^T(\vec{y} - s)$

always PD

$$\nabla^2 J = X^T \text{diag}(s_i(1-s_i)) X$$

↳ $J(w)$ is convex

↳ unique minimum

↳ Newton's Method will find min.

Newton's method in code

```
w⃗ = 0
repeat until convergence:
    solve (XᵀΩX)e⃗ = Xᵀ(y⃗ − s⃗) for e⃗
    w⃗ = w⃗ + e⃗
```

different system each iteration

Observe: – misclassified points far from boundary have a lot of influence
↳ $y_i - s_i$ so large

LDA vs. Logistic Regression

LDA	Logistic Regression
– Produces Posterior that look like logistic regr.	
– Stable for well-separated classes ↳ reliable decision boundary	– not as stable
– can be used for multi-class classification	– designed for binary classification ↳ change to softmax regr. for multi-class
– more accurate when classes are nearly normal ↳ especially when # pts. is small	
	– always separates separable data
	– misclassified data have largest effect on decision boundary
	– robust on non-Gaussian distributions
	– better for binary classification

ROC Curves shows relationship between false pos. & false neg. rates

Suppose our classifier gives posterior probabilities and we can choose a threshold between the 2 classes

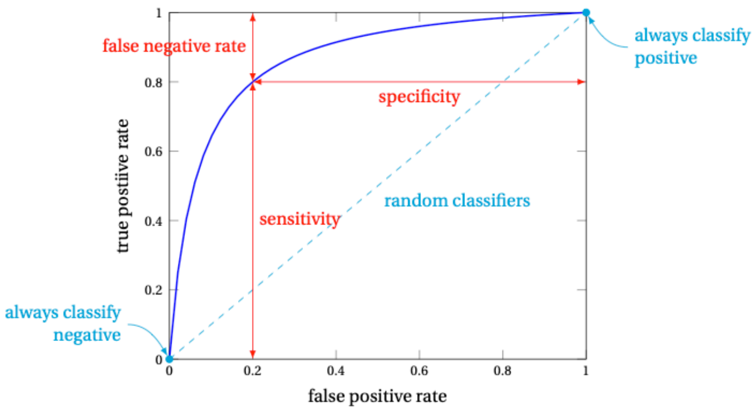

Figure 11.4: The ROC curve

Sensitivity: true positive rate