SB2b Statistical Machine Learning
Hilary Term 2017

Seth Flaxman

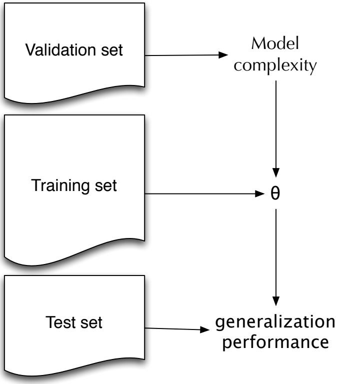
New website:
http://www.stats.ox.ac.uk/~flaxman/course_ml.html

Announcements:
First problem sheet available! (due: Friday 27 Jan. at 5pm for Part B)
First problem class for MSc students: Thursday (26 Jan) at 3pm in LG.01

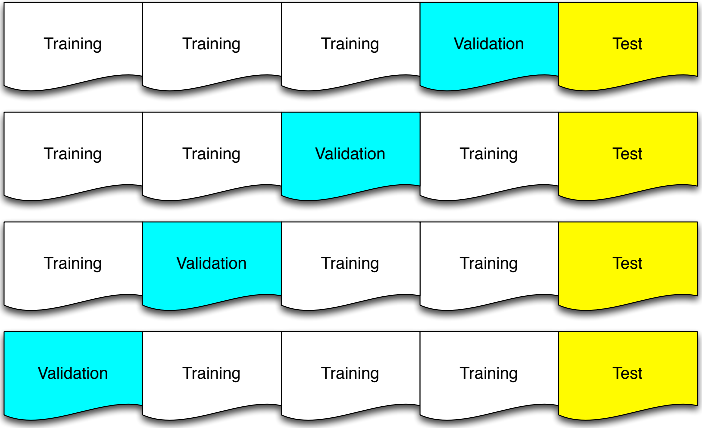
Validation and Cross-Validation

Validation

- For each combination of tuning parameters γ:
  - Train our model on the training set, fit parameters $\theta = \theta(\gamma)$, obtaining decision function $f_{\theta(\gamma)}$.
  - Evaluate $R^{\text{val}}\left(f_{\theta(\gamma)}\right)$ average loss on a validation set.
- Pick γ^* with best performance on validation set.
- Using γ^*, train on both training and validation set to obtain the optimal θ^*.
- $R^{\text{val}}(f_{\theta(\gamma^*)})$ is now a biased estimate of $R(f_{\theta(\gamma^*)})$ and can be overly optimistic!
- Evaluate model with γ^*, θ^* on test set, reporting generalization performance.



Cross-Validation



Logistic regression

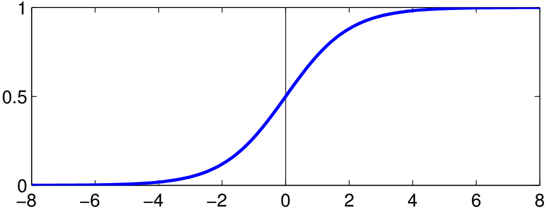
- One of the most popular methods for classification
- Linear model on the probabilities
- Dates back to work on population growth curves by Verhulst [1838, 1845, 1847]
- Statistical use for classification dates to Cox [1960s]
- Independently discovered as the perceptron in machine learning [Rosenblatt 1957]
- Main example of "discriminative" as opposed to "generative" learning
- Naïve approach to classification: linear regression

Logistic regression

- Statistical perspective: consider $\mathcal{Y} = \{0, 1\}$. Generalised linear model with Bernoulli likelihood and logit link:

$$Y|X = x, a, b \sim \text{Bernoulli}\left(s(a + b^\top x)\right)$$

$$s(a + b^\top x) = \frac{1}{1+\exp(-(a+b^\top x))}.$$



- ML perspective: a **discriminative classifier**. Consider binary classification with $\mathcal{Y} = \{-1, +1\}$. Logistic regression uses a parametric model on the conditional $Y|X$, not the joint distribution of (X, Y):

$$p(Y = y|X = x; a, b) = \frac{1}{1 + \exp(-y(a + b^\top x))}.$$

Linearity of log-odds and logistic function

- $a + b^\top x$ models the **log-odds ratio**:

$$\log \frac{p(Y=+1|X=x;a,b)}{p(Y=-1|X=x;a,b)} = a + b^\top x.$$

- Solve explicitly for conditional class probabilities:

$$p(Y=+1|X=x;a,b) = \frac{1}{1+\exp(-(a+b^\top x))} =: s(a+b^\top x)$$
$$p(Y=-1|X=x;a,b) = \frac{1}{1+\exp(+(a+b^\top x))} = s(-a-b^\top x)$$

Fitting the parameters of the hyperplane

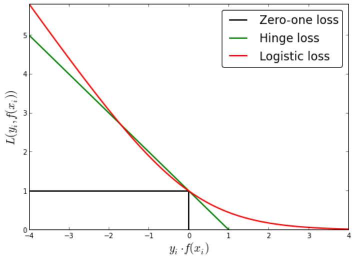
How to learn a and b?

- Consider maximizing the **conditional log likelihood** for $\mathcal{Y} = \{-1, +1\}$:

$$\ell(a, b) = \sum_{i=1}^{n} \log p(Y = y_i | X = x_i) = \sum_{i=1}^{n} \log s(y_i(a + b^\top x_i)).$$

- Equivalent to minimizing the empirical risk associated with the **log loss**:

$$\hat{R}_{\text{log}}(f_{a,b}) = \frac{1}{n} \sum_{i=1}^{n} - \log s(y_i(a + b^\top x_i)) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(a + b^\top x_i)))$$



Logistic Regression

- Not possible to find optimal a, b analytically.
- For simplicity, absorb a as an entry in b by appending '1' into x vector.
- Objective function:

$$\hat{R}_{\text{log}} = \frac{1}{n}\sum_{i=1}^{n} -\log s(y_i x_i^\top b)$$

Logistic Function

$$\begin{aligned} s(-z) &= 1 - s(z) \\ \nabla_z s(z) &= s(z)s(-z) \\ \nabla_z \log s(z) &= s(-z) \\ \nabla_z^2 \log s(z) &= -s(z)s(-z) \end{aligned}$$

- Differentiate wrt b:

$$\begin{aligned} \nabla_b \hat{R}_{\text{log}} &= \frac{1}{n}\sum_{i=1}^{n} -s(-y_i x_i^\top b) y_i x_i \\ \nabla_b^2 \hat{R}_{\text{log}} &= \frac{1}{n}\sum_{i=1}^{n} s(y_i x_i^\top b)s(-y_i x_i^\top b) x_i x_i^\top \succeq 0. \end{aligned}$$

Logistic Regression

- Second derivative is positive-definite: objective function is **convex** and there is **a single unique global minimum**.
- Many different algorithms can find optimal b, e.g.:
  - Gradient descent:
  $$b^{\text{new}} = b + \epsilon \frac{1}{n} \sum_{i=1}^{n} s(-y_i x_i^\top b) y_i x_i$$
  - Stochastic gradient descent:
  $$b^{\text{new}} = b + \epsilon_t \frac{1}{|I(t)|} \sum_{i \in I(t)} s(-y_i x_i^\top b) y_i x_i$$
  where $I(t)$ is a subset of the data at iteration t, and $\epsilon_t \to 0$ slowly ($\sum_t \epsilon_t = \infty, \sum_t \epsilon_t^2 < \infty$).
  - Newton-Raphson:
  $$b^{\text{new}} = b - (\nabla_b^2 \hat{R}_{\text{log}})^{-1} \nabla_b \hat{R}_{\text{log}}$$
  This is also called **iterative reweighted least squares**.
  - Conjugate gradient, LBFGS and other methods from numerical analysis.

Linearly separable data

Assume that the data is linearly separable, i.e. there is a scalar α and a vector β such that $y_i(\alpha + \beta^\top x_i) > 0$, $i = 1, \ldots, n$. Let $c > 0$. The empirical risk for $a = c\alpha$, $b = c\beta$ is

$$\hat{R}_{\mathsf{log}}(f_{a,b}) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-cy_i(\alpha + \beta^\top x_i)))$$

which can be made arbitrarily close to zero as $c \to \infty$, i.e. soft classification rule becomes $\pm\infty$ (overconfidence) $\to$ overfitting.

Multi-class logistic regression

The **multi-class/multinomial** logistic regression uses the **softmax** function to model the conditional class probabilities $p\left(Y=k|X=x;\theta\right)$, for K classes $k=1,\ldots,K$, i.e.,

$$p\left(Y=k|X=x;\theta\right)=\frac{\exp\left(w_k^\top x+b_k\right)}{\sum_{\ell=1}^{K}\exp\left(w_\ell^\top x+b_\ell\right)}.$$

Parameters are $\theta=(b,W)$ where $W=(w_{kj})$ is a $K\times p$ matrix of weights and $b\in\mathbb{R}^K$ is a vector of bias terms.

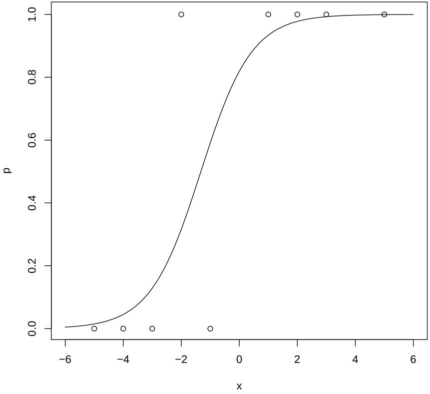
Logistic Regression: Summary

- Makes less modelling assumptions than generative classifiers: often resulting in better prediction accuracy.
- Diverging optimal parameters for linearly separable data: need to **regularise** / pull them towards zero.
- A simple example of a generalised linear model (GLM), for which there is a well established statistical theory:
  - Assessment of fit via deviance and plots,
  - Well founded approaches to removing insignificant features (drop-in deviance test, Wald test).

Overfitting and Regularization

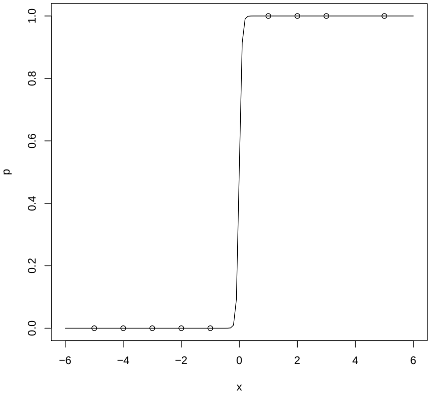
Overfitting in Logistic Regression

```
dx <- c(-5,-4,-3,-2,-1,1,2,3,5)
d <- data.frame(dx)
x <- seq(-6,6,.1)
y <- c(0,0,0,1,0,1,1,1,1)
lr <- glm(y ~ ., data=d,family=binomial)
p <- predict(lr,newdata=data.frame(dx=x),type="response")
plot(x,p,type="l")
points(dx,y)
```



Overfitting in Logistic Regression

```
dx <- c(-5,-4,-3,-2,-1,1,2,3,5)
d <- data.frame(dx)
x <- seq(-6,6,.1)
y <- c(0,0,0,0,0,1,1,1,1)
lr <- glm(y ~ ., data=d,family=binomial)
p <- predict(lr,newdata=data.frame(dx=x),type="response")
plot(x,p,type="l")
points(dx,y)
```



Overfitting in Linear Regression

http://www.stats.ox.ac.uk/~flaxman/sml17/overfitting.html

Regularization

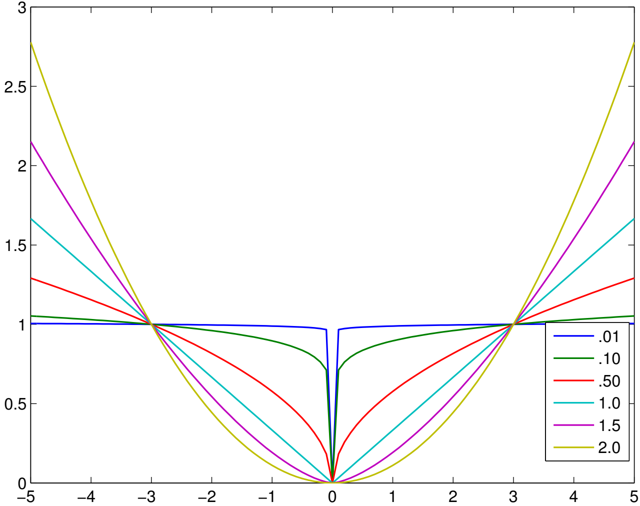
- Flexible models for high-dimensional problems require many parameters.
- With many parameters, learners can easily overfit to the noise in the training data.
- **Regularization**: Limit flexibility of model to prevent overfitting.
- Deep connections with Bayesian perspective: prior distributions provide regularization
- Typically: add term **penalizing** large values of parameters θ.

$$R^{\text{emp}}(\theta) + \lambda \|\theta\|_\rho^\rho = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(a + b^\top x_i))) + \lambda \|b\|_\rho^\rho$$

  where $\rho \in [1, 2]$, and $\|z\|_\rho = (\sum_{j=1}^{p} |z_j|^\rho)^{1/\rho}$ is the L_ρ norm of b (also of interest when $\rho \in [0, 1)$, but is no longer a norm).
- Also known as **shrinkage** methods—parameters are shrunk towards 0.
- Typical cases are $\rho = 2$ (Euclidean norm, **ridge regression**) and $\rho = 1$ (**LASSO**). When $\rho \leq 1$ it is called a **sparsity** inducing regularization.
- λ is a **tuning parameter** (or **hyperparameter**) and controls the amount of regularization, and resulting complexity of the model.

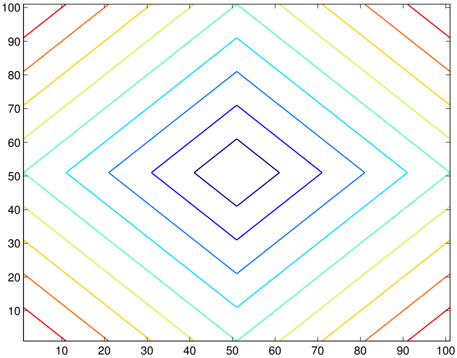
Bach, Sparse Methods in Machine learning.

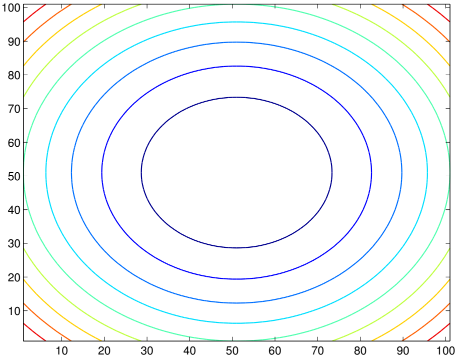
Regularization



L_ρ regularization profile for different values of ρ.

Regularization





L_1 and L_2 norm contours.

L_2 regularization

- Dates to Tikhonov [1943] (more general framework)
- Rediscovered as ridge regression [Hoerl and Kennard 1970]
- Derivation of ridge regression estimator [on board]

Sparsity Inducing Regularization

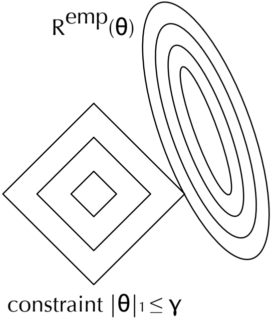
- Consider constrained optimization problem

$$\min_{\theta} R^{\text{emp}}(\theta) \text{ s.t. } \|\theta\|_1 < \gamma$$

- Lagrange multiplier $\lambda > 0$ to enforce constraint,

$$\min_{\theta} R^{\text{emp}}(\theta) + \lambda(\|\theta\|_1 - \gamma)$$

  - At the optimal value of λ, the parameter θ is the one minimizing the regularized empirical risk objective.
  - Conversely, given λ, there is a value of γ such that the corresponding optimal Lagrange multiplier is λ.
- Generally: L_1 regularization leads to optimal solutions with many zeros, i.e. the regression function depends only on the (small) number of features with non-zero parameters.



Illustration

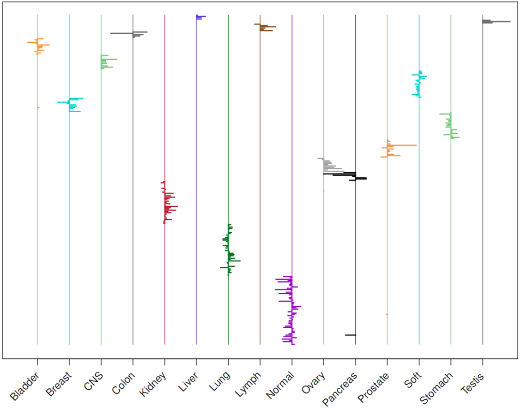
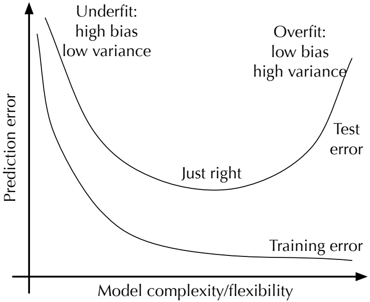


Figure 1.1 *15-class gene expression cancer data: estimated nonzero feature weights from a lasso-regularized multinomial classifier. Shown are the 254 genes (out of 4718) with at least one nonzero weight among the 15 classes. The genes (unlabelled) run from top to bottom. Line segments pointing to the right indicate positive weights, and to the left, negative weights. We see that only a handful of genes are needed to characterize each class.*

Source: *Statistical Learning with Sparsity: the Lasso and Generalizations* by Hastie, Tibshirani, and Wainwright

Bias/variance tradeoff

- Why does regularization prevent overfitting?
- Regularization introduces bias with the goal of reducing variance:



- http://www.stats.ox.ac.uk/~flaxman/sml17/regularization.html

Useful Resources and Pointers

[links below are clickable]

- Compendium of common loss functions for classification and regression
- Statistical Learning with Sparsity: The Lasso and Generalizations by Hastie, Tibshirani, and Wainwright
- glmnet (R package) tutorial
- Matrix Cookbook