Stat 406: Algorithms for classification and prediction

Final review

Kevin Murphy

Wed 11 April, 2007[1]

[1] Slides last updated on April 10, 2007

Outline

- Linear regression
- Overfitting, model selection
- Ridge regression
- PCA
- EM for mixture models

Linear regression

Linear regression is the following conditional density model

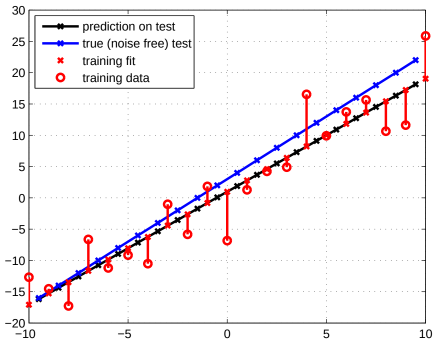
$$p(y_i|\mathbf{x}_i) = \mathcal{N}(y_i|\mathbf{w}^T\mathbf{x}_i, \sigma^2) \tag{1}$$

This can be written equivalently as

$$y_i = \mathbf{w}^T\mathbf{x}_i + \epsilon_i \tag{2}$$

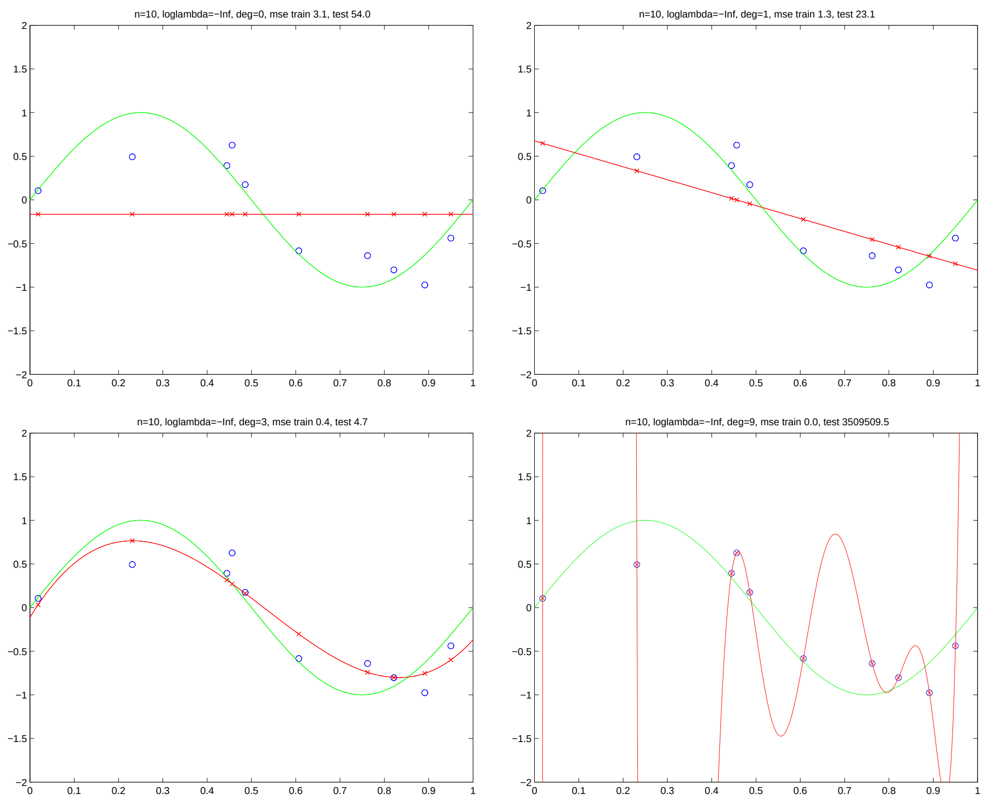
where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ e.g.

$$y_i = w_0 + w_1 x_i + \epsilon_i \tag{3}$$



Polynomial regression

$$p(y|x) = \mathcal{N}(y|\mathbf{w}^T \boldsymbol{\phi}(x), \sigma^2) \quad (4)$$
$$\boldsymbol{\phi}(x) = [1, x, x^2] \quad (5)$$



Linear least squares

The likelihood of the data is

$$p(\mathcal{D}|\mathbf{w}, \lambda_y) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mathbf{w}^T\mathbf{x}_i, \sigma^2) \tag{6}$$

Let $\ell = \log p(\mathbf{y}|X, \mathbf{w}, \sigma^2)$ be the log likelihood.

$$\frac{\partial \ell}{\partial \mathbf{w}} = 0 \Rightarrow \hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y} \tag{7}$$

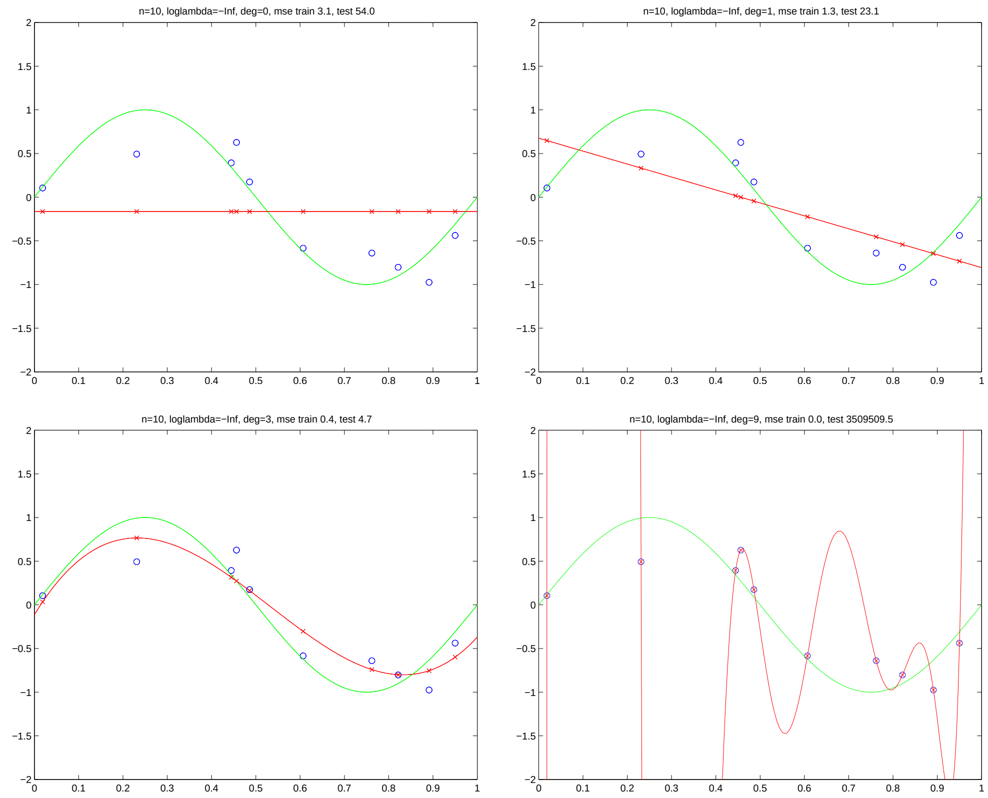
$$\frac{\partial \ell}{\partial \sigma^2} = 0 \Rightarrow \hat{\sigma}^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\mathbf{w})^2 \tag{8}$$

Outline

- Linear regression $\surd$
- Overfitting, model selection
- Ridge regression
- PCA
- EM for mixture models

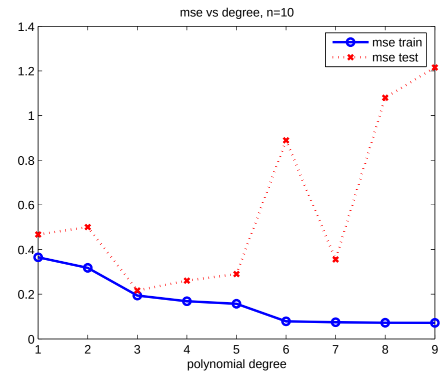
OVERFITTING

A 9 degree polynomial can perfecly interpolate 10 data points i.e., get 0 training error. Yet it may not generalize well.



Training vs test error

Plot of RMSE vs degree



Can use cross validation to do model selection.

Outline

- Linear regression $\surd$
- Overfitting, model selection $\surd$
- Ridge regression
- PCA
- EM for mixture models

Ridge regression: motivation

Parameters of overly complex models can get large; penalize magnitude to enforce smooth functions.

$deg = 0$	$deg = 1$	$deg = 3$	$deg = 9$
-0.165	-0.165	-0.165	-0.165
	-0.443	2.500	14171.273
		-7.301	-196385.669
		4.468	1148124.938
			-3681962.824
			7152057.596
			-8677072.717
			6448974.666
			-2691799.620
			483980.554

Ridge regression (weight decay, L2 regularization)

Gaussian Prior on weights

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, {\lambda_w}^{-1} I_d) \tag{9}$$

Posterior

$$\begin{aligned} -\log p(\mathbf{w}|D) &\propto -\log \mathcal{N}(\mathbf{w}|0, {\lambda_w}^{-1} I_p)\mathcal{N}(\mathbf{y}|X\mathbf{w}, {\lambda_y}^{-1} I_N) && (10) \\ &\propto \lambda_w ||\mathbf{w}||^2 + \lambda_y ||\mathbf{y} - X\mathbf{w}||^2 && (11) \end{aligned}$$

MAP estimate

$$\begin{aligned} \hat{\mathbf{w}}_{ridge} &= \arg\min_{\mathbf{w}} ||\mathbf{y} - X\mathbf{w}||^2 + \lambda ||\mathbf{w}||^2 && (12) \\ &= (X^T X + \lambda I) X^T \mathbf{y} && (13) \end{aligned}$$

where $\lambda = \frac{\lambda_w}{\lambda_y}$

Connection with SVD

Let $X = UDV^T$, where $U^TU = V^TV = I$, $VV^T = I$. For least squares,

$$\hat{\mathbf{w}}_{ls} = VD^{-1}U^T\mathbf{y} \tag{14}$$

$$\hat{\mathbf{y}} = X\hat{\mathbf{w}}_{ls} = \sum_{j=1}^{d} \mathbf{u}_j\mathbf{u}_j^T\mathbf{y} \tag{15}$$

For ridge,

$$\hat{\mathbf{w}}_{ridge} = V(D^2 + \lambda I)^{-1}DU^T\mathbf{y} \tag{16}$$

$$\hat{\mathbf{y}} = X\hat{\mathbf{w}}_{ridge} = \sum_{j=1}^{d} \mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y} \tag{17}$$

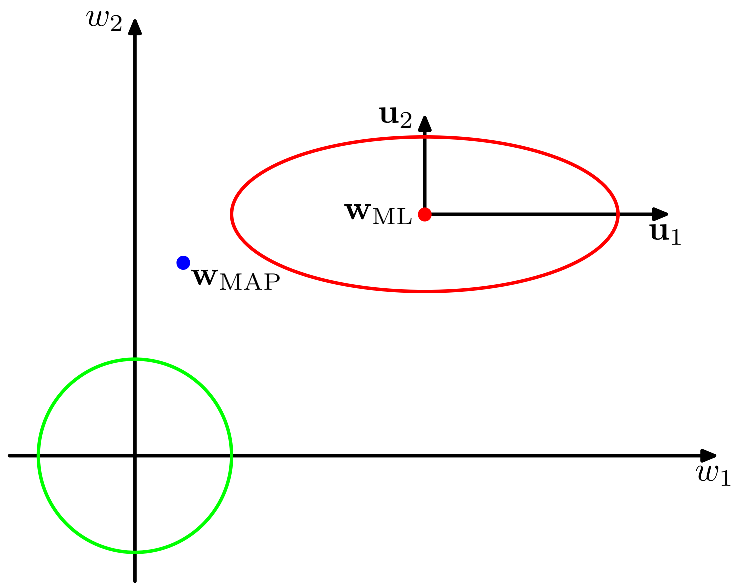
We shrink parameters w_j to 0 more if they have small d_j^2.

Connection with PCA

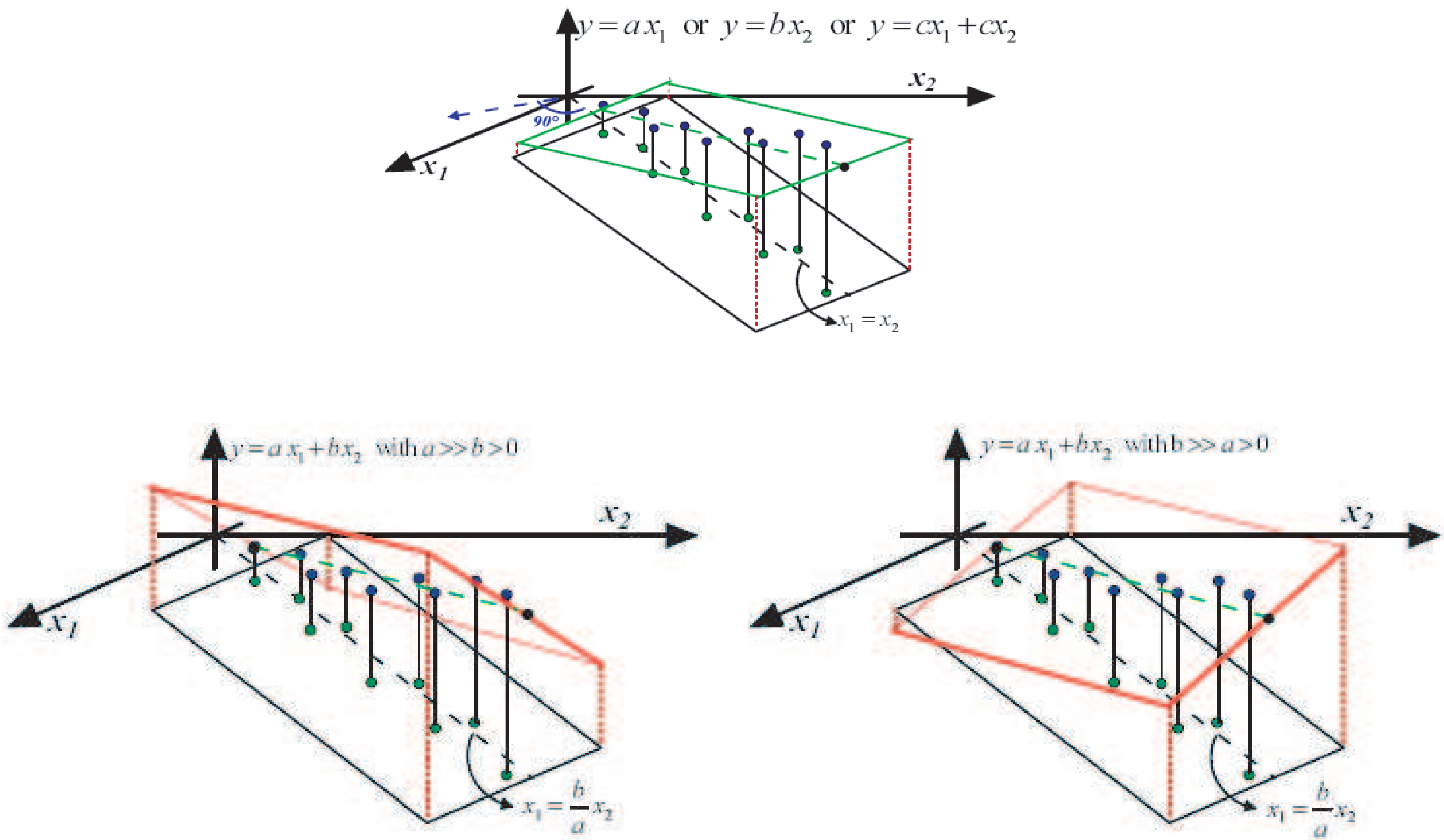
If $X = UDV^T$, then the eigen decomposition of the sample covariance matrix is

$$X^T X = V D^2 V \tag{18}$$

Hence small d_j (large shrinkage) corresponds to small variance directions; large d_j (small srhinkage) corresponds to large variance.



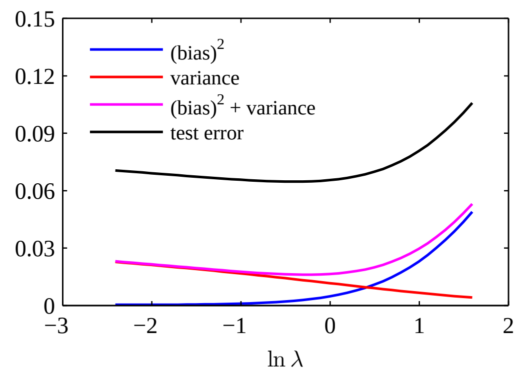
Regularize the low variance direction more



Bias-variance tradeoff

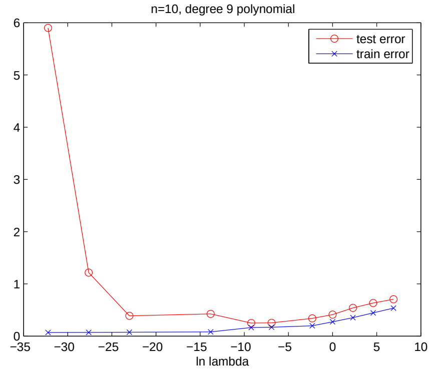
Ridge is a biased estimator. But it is much lower variance. So it is much better overall, since

$$MSE = \text{variance} + \text{bias}^2 \tag{19}$$



Picking the regularization constant

Use cross validation



Spline model

Suppose we assume the function is piecewise constant, having height w_j in interval I_j:

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{d} w_j I(\mathbf{x} \in I_j) \tag{20}$$

This is called a (zero-order) **spline model**. The intervals can be defined by a series of **knots**, $I_j = (k_j, k_{j+1}]$, at fixed locations. Then we get a sparse design matrix, where $X_{ij} = 1$ if x_i is in interval j and 0 otherwise.

We may more parameters than data points. Solution: We can impose a smoothness prior on the neighboring w_j'.

Generalized ridge

$$p(\mathbf{w}) \sim \mathcal{N}_\lambda(\boldsymbol{\mu} = 0, \Lambda = \lambda D^T D) \tag{21}$$

where D is the following $(n-1) \times n$ difference matrix:

$$D = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \tag{22}$$

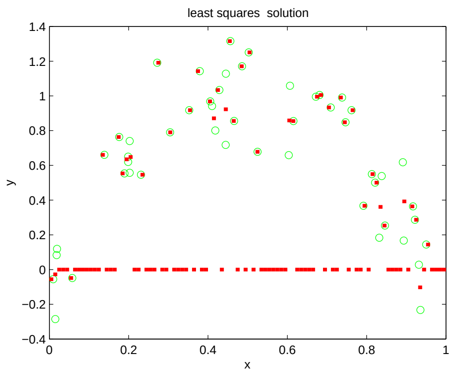
The term in the exponent gives

$$\mathbf{w}^T (D^T D)\mathbf{w} = ||D\mathbf{w}||^2 = \tfrac{1}{2} \sum_{i=1}^{n-1} (w_{i+1} - w_i)^2 \tag{23}$$

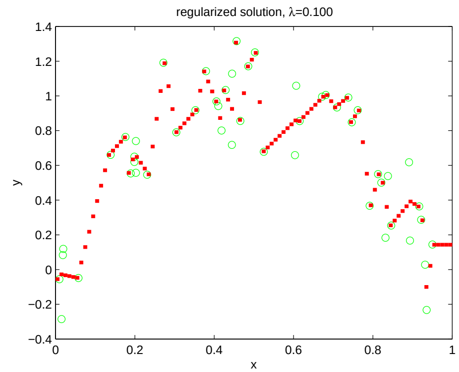
MAP estimate

$$\begin{aligned} J(\mathbf{w}) &= -\log \mathcal{N}_\lambda(\mathbf{y}||\mathbf{w}, I_n) - \log \mathcal{N}_\lambda(\mathbf{w}|0, \sqrt{\lambda} D^T D) & (24) \\ &= \frac{1}{2}||\mathbf{y} - \mathbf{w}||^2 + \frac{\lambda}{2}||D\mathbf{w}||^2 + const & (25) \end{aligned}$$

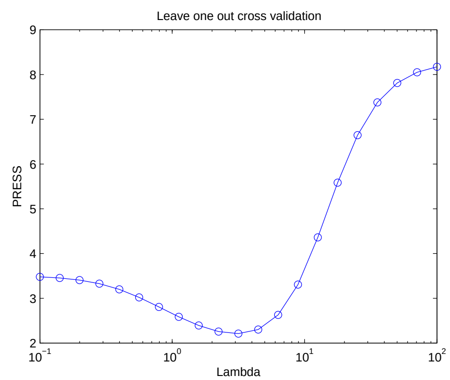
Regularized splines



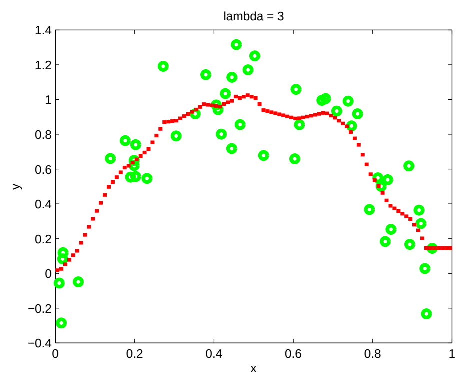
(a)



(b)



(c)



(d)

Outline

- Linear regression $\surd$
- Overfitting, model selection $\surd$
- Ridge regression $\surd$
- PCA
- EM for mixture models

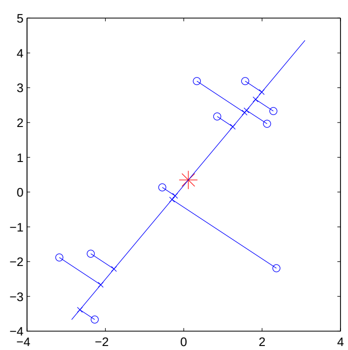
Principal components analysis

Find low dimensional space (pc basis) $\mathbf{w}$, and coordinates (principal components) $\mathbf{z}$ in that space, that best represents data points $\mathbf{x}$ in a least squares sense:

$$J(\mathbf{w}_1, \mathbf{z}_1) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - z_{1i}\mathbf{w}_1)^2 \tag{26}$$

subject to $\mathbf{w}_1^T \mathbf{w}_1 = 1$, $\mathbf{w}_1 \in \mathbb{R}^d$, $\mathbf{z}_1 \in \mathbb{R}^n$.

$$\mathbf{Z} = \mathbf{X}\mathbf{W}, \quad \hat{\mathbf{X}} = \mathbf{Z}\mathbf{W}^T \tag{27}$$



First PC = principal evec of cov

$$\frac{\partial}{\partial z_{1i}} J(\mathbf{w}_1, z_{1i}) = 0 \Rightarrow z_{1i} = \mathbf{w}_1^T \mathbf{x}_i \tag{28}$$

Plugging in

$$\frac{\partial}{\partial w_{1i}} J(\mathbf{w}_1) = 0 \Rightarrow \tag{29}$$

$$\hat{C}\mathbf{w}_1 = \lambda_1 \mathbf{w}_1 \tag{30}$$

$$\hat{C} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \tag{31}$$

Variance of projected data is

$$\mathbf{w}_1^T \hat{C} \mathbf{w}_1 = \lambda_1 \tag{32}$$

SECOND PC = 2ND LARGEST EVEC OF COV

Pick direction of maximum variance subject to $\mathbf{w}_1^T \mathbf{w}_2 = 0$ and $\mathbf{w}_2^T \mathbf{w}_2 = 1$. We find

$$\hat{C}\mathbf{w}_2 = \lambda_2 \mathbf{w}_2 \tag{33}$$

Compuation of PCA

4 methods

- Eig of $X^T X$, $O(d^3)$ time
- Eig of XX^T, $O(n^3)$ time
- SVD of X, $O(nd^2)$ time
- SVD of X^T, $O(dn^2)$ time

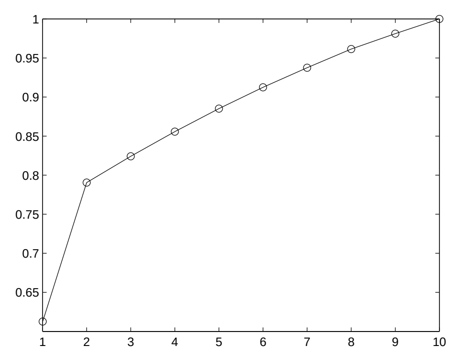
Choosing the number of dimensions

Residual MSE

$$J = \sum_{j=K+1}^{d} \lambda_j \tag{34}$$

Make scree plot

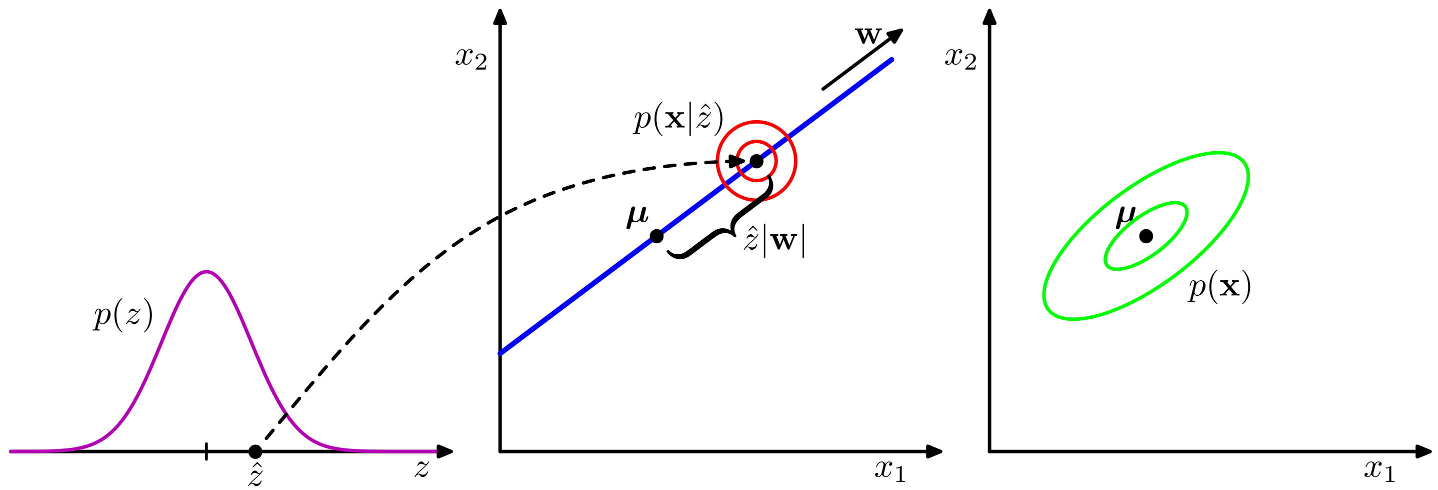
$$\sum_{j=1}^{k} \lambda_j / (\sum_{j'=1}^{K} \lambda_{j'}) \tag{35}$$



Probabilistic PCA

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \sigma^2\mathbf{I}_d) \tag{36}$$

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k) \tag{37}$$



MLE for PPCA

Marginal distribution on observed data

$$
\begin{aligned}
E\left[\mathbf{x}\right] &= E\left[\mathbf{W}\mathbf{z}+\boldsymbol{\mu}+\boldsymbol{\epsilon}\right]=\boldsymbol{\mu} && (38)\\
\mathrm{Cov}[\mathbf{x}] &= E\left[(\mathbf{W}\mathbf{z}+\boldsymbol{\epsilon})(\mathbf{W}\mathbf{z}+\boldsymbol{\epsilon})^T\right]=E\left[\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T\right]+E\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\right] && (39)\\
&= \mathbf{W}\mathbf{W}^T+\sigma^2 I \stackrel{\text{def}}{=} \mathbf{C} && (40)
\end{aligned}
$$

Log likelihood

$$
\log p(\mathbf{X}|\boldsymbol{\mu},\mathbf{W},\sigma^2) = -\frac{n}{2}\ln|\mathbf{C}| - \tfrac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}) \qquad (41)
$$

MLE for PPCA

MLE mean

$$\boldsymbol{\mu} = \overline{\mathbf{x}} \tag{42}$$

MLE weight matrix

$$\hat{\mathbf{W}} = \mathbf{U}_K(\boldsymbol{\Lambda}_K - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{R} \tag{43}$$

where $\mathbf{U}_K$ is the $d \times K$ matrix whose columns are the first K eigenvectors of $\mathbf{S}$, $\boldsymbol{\Lambda}_K$ is the corresponding diagonal matrix of eigenvalues, amd $\mathbf{R}$ is an arbitrary $K \times K$ orthogonal matrix.
MLE variance

$$\hat{\sigma}^2 = \frac{1}{d-K}\sum_{j=K+1}^{d}\lambda_j \tag{44}$$

which is the average variance associated with the discarded dimensions.

PPCA: why bother with probabilities?

- Defines a proper density model $p(\mathbf{x})$
- Can be used inside a mixture distribution or a generative classifier
- Can be compared to other density models $p(\mathbf{x})$
- Provides a likelihood function for a Bayesian analysis

Outline

- Linear regression $\surd$
- Overfitting, model selection $\surd$
- Ridge regression $\surd$
- PCA $\surd$
- EM for mixture models

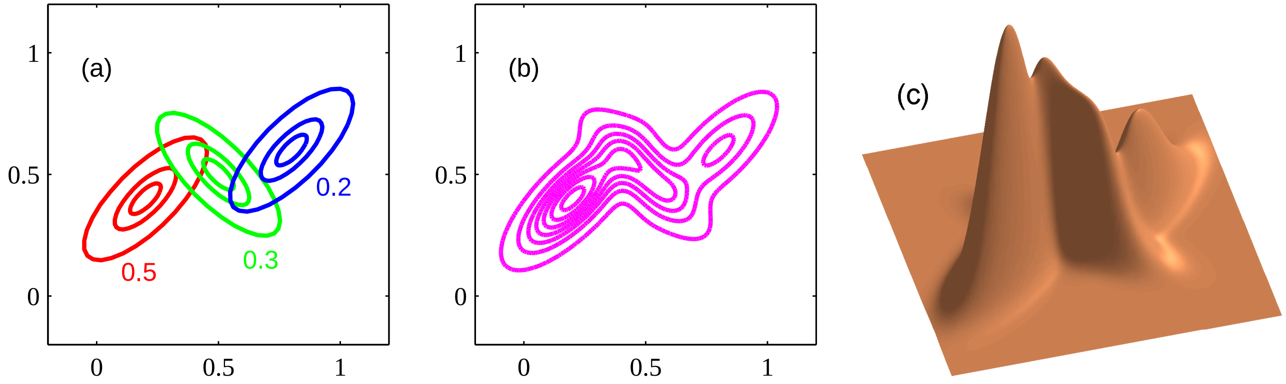
Joint probability model

$$p(x|z=k,\theta) = \mathcal{N}(x|\mu_k,\Sigma_k) \tag{45}$$

$$p(z=k|\theta) = \pi_k \tag{46}$$

Observed data probability model is a mixture

$$p(x|\theta) = \sum_{k=1}^{K} p(z=k)p(x|z=k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k) \tag{47}$$



MLE for fully observed data problem

complete data log likelihood is given by

$$\ell_c(\theta) = \log p(x_{1:N}, z_{1:N}|\theta) \tag{48}$$
$$= \log \prod_n p(z_n|\pi) p(x_n|z_n, \theta) \tag{49}$$
$$= \log \prod_n \prod_k [\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)]^{I(z_n=k)} \tag{50}$$
$$= \sum_n \sum_k I(z_n = k)[\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)] \tag{51}$$

Hence we can find the optimal μ_k, Σ_k separately for each k (empirical mean/ covariance), and then find the optimal π_k by counting.

EM intuition

- If we knew the values of the latent variables z_n, then optimizing the (complete data) likelihood wrt θ would be easy: we would simply esimate μ_k and Σ_k applying the standard closed-form formula to all the data assigned to cluster k.
- Since we don't know the z_n, let's estimate them, and use their **filled in** values as substitutes for the real values. More precisely, we will optimize the *expected* complete data log likelihood instead of the actual complete data log likelihood.
- Since the estimate of z_n depends on θ, we iterate until convergence.

EM algorithm

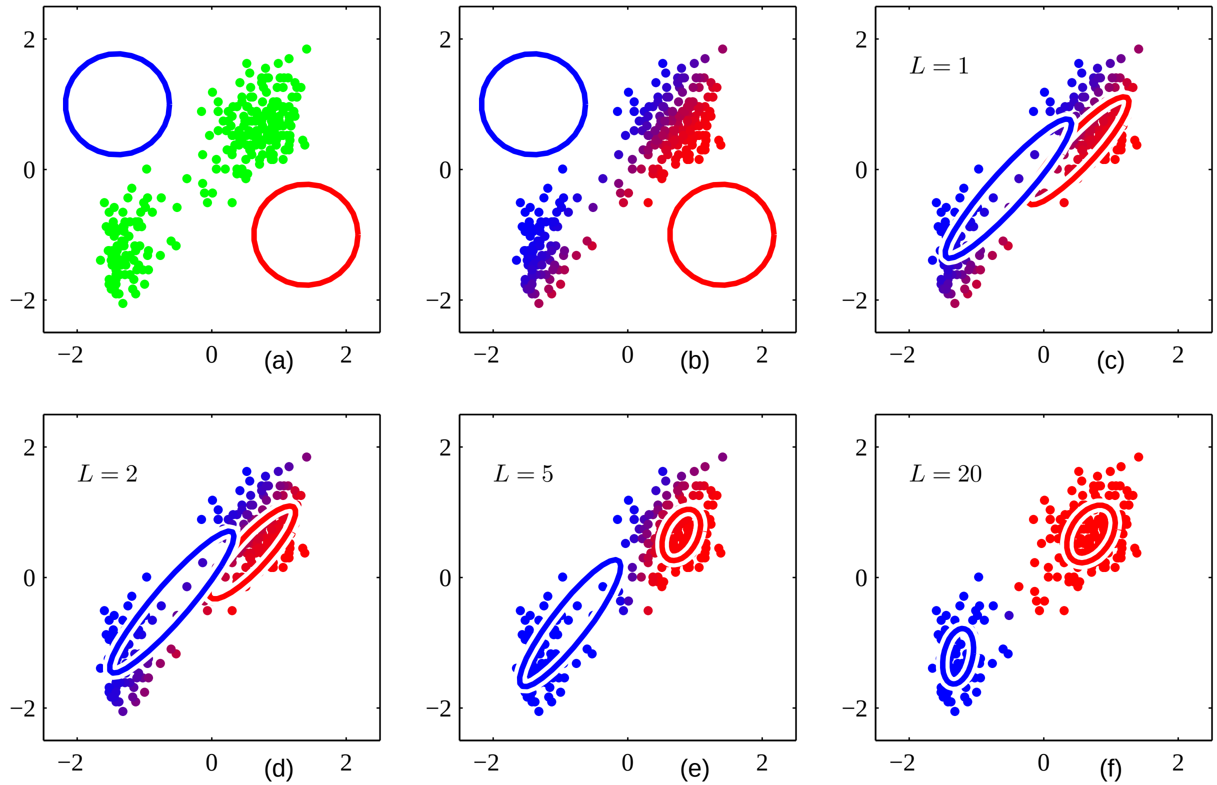
1. Initialize θ.
2. Repeat until $\ell(\theta)$ stops changing
   (a) E step: compute $p(z_n|x_n, \theta^{old})$ for each case n.
   (b) M step: compute
   $$\theta^{new} = \arg\max_{\theta} Q(\theta, \theta^{old}) \tag{52}$$
   where **auxiliary function** Q is the expected complete data log likelihood.
   (c) Compute the log likelihood
   $$\ell(\theta) = \log \sum_{n} \sum_{z_n} p(z_n, x_n|\theta) \tag{53}$$

Q function for GMMs

Expected complete data log likelihood:

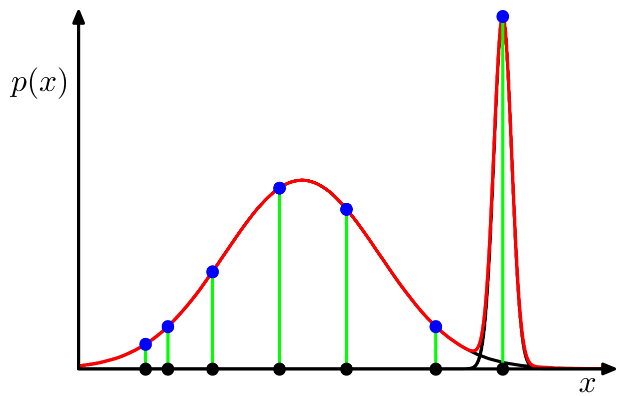
$$
\begin{aligned}
Q(\theta, \theta^{old}) &= E \sum_n \log p(x_n, z_n|\theta) && (54)\\
&= E \sum_n \sum_k I(z_n = k) \log[\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)] && (55)\\
&= \sum_n p(z_n|x_n, \theta^{old}) \sum_k I(z_n = k) \log[\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)] && (56)\\
&= \sum_n \sum_k r_{nk} \log[\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)] && (57)\\
&= \sum_n \sum_k r_{nk} \log \pi_k + \sum_n \sum_k r_{nk} \log \mathcal{N}(x_n|\mu_k, \Sigma_k)] && (58)\\
&= J(\pi) + J(\mu, \Sigma) && (59)
\end{aligned}
$$

EM for GMM demo



Need for regularization (MAP estimation)

Some mixture components may have few data points assigned to them. This can cause various problems. e.g., the likelihood can blow up by letting $\sigma_j \to 0$.



K MEANS

Special case of EM for GMMs where

- $\Sigma_k = \sigma^2 I$ is fixed
- We do a hard assignment during the E step:

$$
\begin{aligned}
z_n^* &= \arg\max_k p(k|x_n, \theta) && (60)\\
&= \arg\max_k \exp(-\tfrac{1}{2}||x_n - \mu_k||^2) && (61)\\
&= \arg\min_k ||x_n - \mu_k||^2 && (62)
\end{aligned}
$$

EM for mixtures of Bernoullis

For clustering binary data, we can use

$$p(x|z=k,\theta) = \prod_{i=1}^{K} Be(x_i|\theta_{ki}) = \prod_{i=1}^{K} x_i^{\theta_{ki}}(1-x_i)^{1-\theta_{ki}} \qquad (63)$$

We find $\boldsymbol{\mu}_k$ is a weighted average of all the bit vectors $\mathbf{x}_i$ assigned to cluster k.