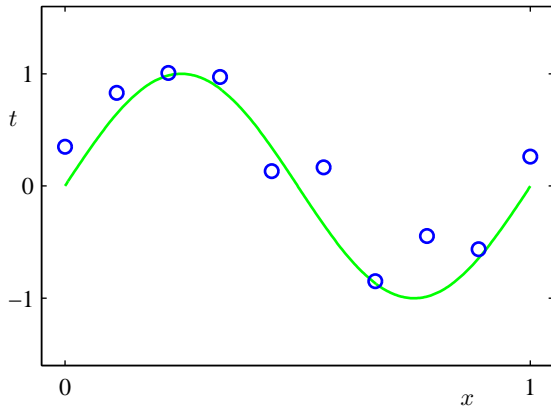


Basic Linear Regression

Christian Osendorfer

A noisy real-valued function

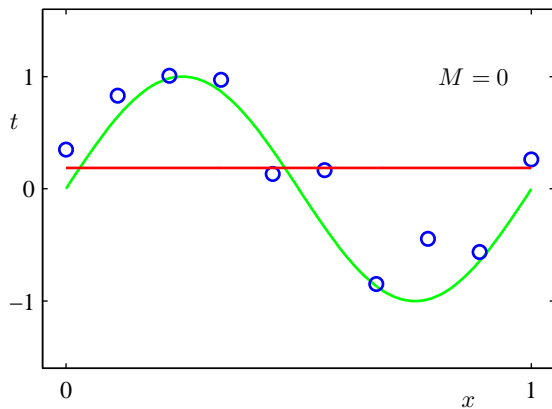


$$\text{inputs: } \mathbf{X} = (x_1, \dots, x_N)^T \quad (1)$$

$$\text{targets: } \mathbf{z} = (z_1, \dots, z_N)^T, \quad z_i = h(x_i) + \epsilon = \sin(2\pi x_i) + \epsilon \quad (2)$$

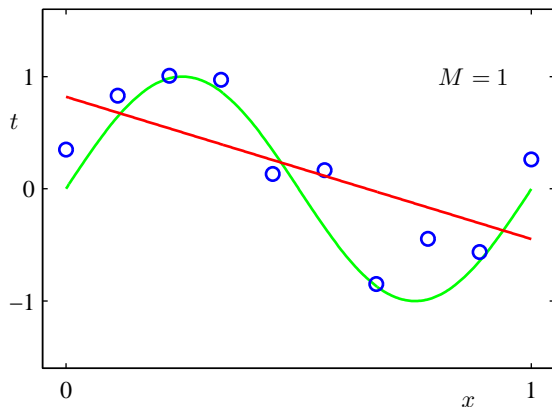
Most figures are from C. Bishop: Pattern Recognition and Machine Learning

Model: 0th order polynomial



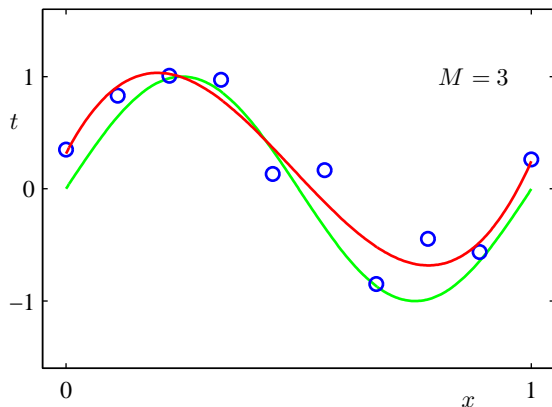
$$y(x, \mathbf{w}) = w_0$$

Model: 1st order polynomial



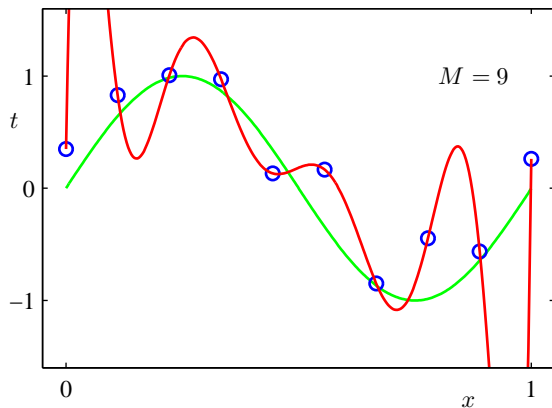
$$y(x, \mathbf{w}) = w_0 + w_1 x$$

Model: 3rd order polynomial



$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + w_3x^3$$

Model: 9th order polynomial



$$y(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j$$

Problem Definition

We have input vectors \mathbf{x}_i and associated output values z_i . We want to describe the underlying functional relation.

What about the following simple model? (Looks familiar?)

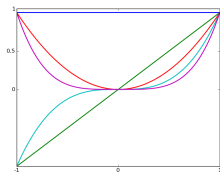
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (3)$$

where

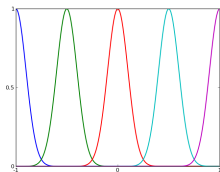
- ϕ **basis function** – many choices, can be nonlinear
- w_0 **bias** – equivalent to defining $\phi_0 \equiv 1$

It is **linear** in \mathbf{w} ! Nothing new if you know Taylor expansion, Fourier transform, wavelets...

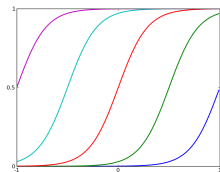
Typical Basis Functions



polynomials



Gaussians



“sigmoids”
(=S-shaped curves)

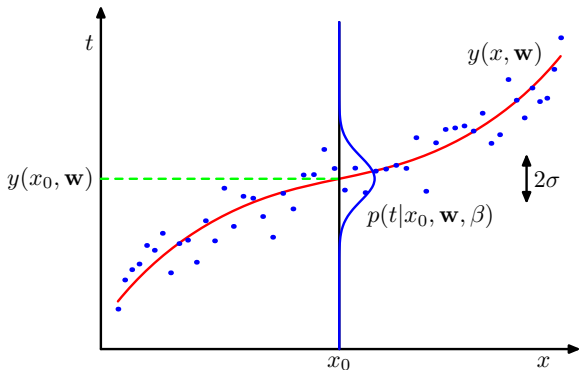


Illustration of a Gaussian conditional distribution for t (we use: z) given x . σ is fixed (i.e. in particular, it does not depend on x).

We measure random variable z as

$$z = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad [\epsilon: \text{Gaussian, zero mean}] \quad (4)$$

The dataset \mathcal{D} consists of observations $\mathbf{z} = (z_1, z_2, \dots, z_N)$ and corresponding input vectors $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$.

Then the **likelihood function** is (conditional independence on the fly!)

$$p(\mathbf{z}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^N \mathcal{N}(z_n | \mathbf{w}^T \phi(\mathbf{x}_n), \sigma^2) \quad (5)$$

As usual, it's easier to deal with logarithmized probabilities (\mathbf{X} is left out for brevity):

$$\ln p(\mathbf{z}|\mathbf{w}, \sigma^2) = -\frac{N}{2} \ln \sigma^2 - \underbrace{\frac{N}{2} \ln(2\pi)}_{const.} - \frac{1}{\sigma^2} \underbrace{\frac{1}{2} \sum_{n=1}^N (z_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2}_{E_{\mathcal{D}}(\mathbf{w})} \quad (6)$$

Maximum Likelihood Solution

Gradient of log likelihood w.r.t. \mathbf{w} (why do we need this?):

$$\nabla_{\mathbf{w}} \ln p(\mathbf{z}|\mathbf{w}, \sigma^2) \propto \sum^n (z_n - \mathbf{w}^T \phi(\mathbf{x}_n)) \phi(\mathbf{x}_n) \quad (7)$$

Let's go back to eq. 6 and have a look at $E_{\mathcal{D}}(\mathbf{w})$ (minus sign is missing, so we solve for the minimum here!)

$$E_{\mathcal{D}}(\mathbf{w}) = \frac{1}{2}(\mathbf{z} - \Phi\mathbf{w})^T(\mathbf{z} - \Phi\mathbf{w}) \quad (8)$$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & & \vdots \\ \vdots & \vdots & \ddots & \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} = \text{design matrix of } \phi.$$

Using standard rules for **matrix calculus** (see the handout by Sam Roweis linked on the course website!), we get:

$$\nabla_{\mathbf{w}} \ln p(\mathbf{z}|\mathbf{w}, \sigma^2) \propto \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w} - \mathbf{\Phi}^T \mathbf{z}$$

Apart from the sign, this is identical to eq. 7 (verify!). But in this form, it is immediately clear how to find the maximum likelihood solution! Solve

$$\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w} = \mathbf{\Phi}^T \mathbf{z}$$

Normal equations of (ordinary) least squares problem.

$$\mathbf{w}_{\text{ML}} = \underbrace{(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T}_{=\mathbf{\Phi}^\dagger} \mathbf{z} \quad (9)$$

$\mathbf{\Phi}^\dagger$ is called **Moore-Penrose pseudo-inverse** of $\mathbf{\Phi}$ (because for an *invertible* square matrix, $\mathbf{\Phi}^\dagger = \mathbf{\Phi}^{-1}$).

Second derivative: $\mathbf{\Phi}^T \mathbf{\Phi}$ is (semi) positive definite.

Computational aspect

Computing the MLE solution \mathbf{w}_{ML} using the normal equations is not such a great idea. Why?

- ▶ If Φ is not full rank, $(\Phi^T \Phi)^{-1}$ does not exist (why? how can rank deficiency happen?)
- ▶ Even if Φ is full rank, it can be *ill-conditioned* (???), (i.e. $\kappa(\Phi)$ is large), $\Phi^T \Phi$ will be even worse ($\kappa(\Phi^T \Phi) = \kappa(\Phi)^2$).

Applied numerical computing :)

Singular Value Decomposition (SVD)

Any real $N \times D$ matrix \mathbf{X} can be decomposed as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

with $\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$, $\mathbf{V}\mathbf{V}^T = \mathbf{I}_D$ and $\mathbf{\Sigma}$ is a $N \times D$ diagonal matrix containing the $\min(N, D)$ *singular values* $\sigma_i \geq 0$.

There is a connection to the eigenvalue decomposition of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$. See the link on the website for a nice explanation of the SVD.

For now we assume

$$\Phi = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Using this in $\Phi^T \Phi \mathbf{w} = \Phi^T \mathbf{z}$, we get

$$\mathbf{V} \Sigma \Sigma^T \mathbf{w} = \mathbf{V} \Sigma \mathbf{U}^T \mathbf{z}$$

(Why?) Solving for \mathbf{w} results in

$$\mathbf{w} = \mathbf{V} \hat{\Sigma} \mathbf{U}^T \mathbf{z}$$

Why?? What is $\hat{\Sigma}$? Are any entries on the diagonal of this matrix problematic?

Out of all solutions that minimize $\|\Phi \mathbf{w} - \mathbf{z}\|_2^2$ this one has minimum $\|\mathbf{w}\|_2^2$. Ask in person, if interested, but don't how to do proof.

Online Learning

The algorithm for learning maximum likelihood estimates for \mathbf{W} assumes that all data points are available at once (*offline* learning, *batch* learning).

What if large data sets are involved so that batch processing of all points at once is infeasible? What if data points arrive over time (sequentially), and possibly should be discarded as soon as possible?

Online Learning

Online Learning

Sequential learning using the *Robbins Monro* algorithm (cf. eq. 7). Note that the problem is convex!

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \alpha_t \frac{\partial}{\partial \mathbf{w}_{t-1}} \log p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{w}_{t-1}, \beta)$$

If $\{\alpha_1, \alpha_2, \dots\}$ satisfy the following conditions, then the sequence \mathbf{w}_t converges to the optimum.

$$\lim_{t \rightarrow \infty} \alpha_t = 0 \tag{10}$$

$$\sum_{t=1}^{\infty} \alpha_t = \infty \tag{11}$$

$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty \tag{12}$$

General name of this optimization procedure? Interpretation of constraints?

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

Regularization

MLE often suffers from overfitting \hookrightarrow use **regularization**:

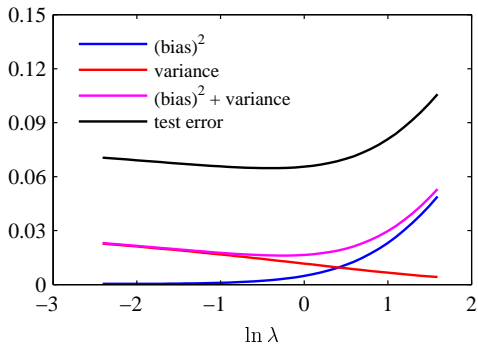
$$E_{\mathcal{D}} = \frac{1}{2} \sum^n (z_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2 + \frac{\lambda}{2} \sum^M |w_j|^q \quad (13)$$

\Rightarrow this is like a Lagrange term specifying an additional constraint:

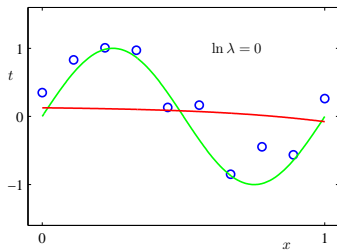
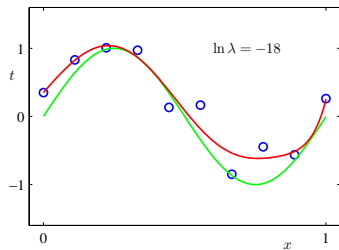
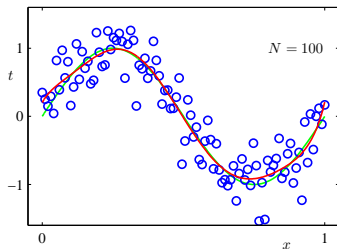
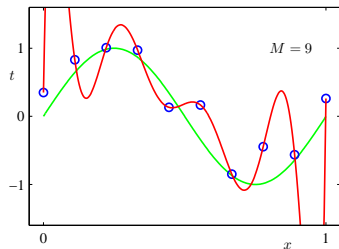
$$\sum^M |w_j|^q \leq \epsilon$$

\hookrightarrow most often, use quadratic regularizer (ℓ_2 , *ridge regression*): $q = 2$, i.e.

$$\sum^M |w_j|^q = \mathbf{w}^T \mathbf{w}$$



Regularization and Model Complexity



MAP Solution

Recap: For the coin flip experiment, we introduced prior information to prevent **overfitting**. By analogy:

	train data	likelihood	prior	posterior
coin:	$\mathcal{D} = \mathbf{X}$	$p(\mathcal{D} \theta)$	$p(\theta a, b)$	$p(\theta \mathcal{D})$
regr.:	$\mathcal{D} = \{\mathbf{X}, \mathbf{z}\}$	$p(\mathbf{z} \mathbf{X}, \mathbf{w}, \beta)$	$p(\mathbf{w} \cdot)$	$p(\mathbf{w} \mathbf{X}, \mathbf{z}, \cdot)$

Note: As mentioned earlier, \mathbf{X} is usually dropped for clarity.

Prior: How to find a good one?

- ▶ recall (from Eq. 5) that our likelihood function $p(\mathbf{z}|\mathbf{w}, \beta)$ is a Gaussian,
- ▶ treat precision $\beta = 1/\sigma^2$ as a known parameter (for now),
- ▶ know that the **conjugate prior** for a Gaussian with known variance is also a Gaussian.

MAP Solution

Hence introduce the following prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0) \quad [\mathbf{m}_0: \text{mean}, \mathbf{S}_0: \text{covariance}] \quad (14)$$

Often we don't know much about the prior distribution anyway. For a suitably designed model with independent parameters \mathbf{w} , the following prior is usually reasonable:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0 = 0, \mathbf{S}_0 = \alpha^{-1}\mathbf{I}) \quad (15)$$

This results in the posterior:

$$p(\mathbf{w}|\mathbf{z}, \alpha, \beta) \propto p(\mathbf{z}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha) \quad (16)$$

Applying the *negative* log yields again an error function E_{MAP} to *minimize*:

$$\ln p(\mathbf{w}|\mathbf{z}, \alpha, \beta) = \frac{\beta}{2} \sum^n (z_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const.} \quad (17)$$

⇒ Interpret the second term of this function! What values for \mathbf{w} does it *prefer*?

Posterior Distribution

We can actually find a closed expression for the posterior!

Posterior parameter distribution

$$p(\mathbf{w}|\mathbf{z}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \quad (18)$$

with

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{z} \right) \quad (19)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \Phi^T \Phi \quad (20)$$

Properties of the posterior:

- ▶ Since we again have a Gaussian, the maximum posterior solution equals the mode: $\mathbf{w}_{\text{MAP}} = \mathbf{m}_N$.
- ▶ In the limit of an infinitely broad prior, $\mathbf{S}_0^{-1} \rightarrow 0$, therefore $\mathbf{w}_{\text{MAP}} \rightarrow \mathbf{w}_{\text{ML}} = \Phi^\dagger \mathbf{z}$ (Eq. 9).
- ▶ For $N = 0$, i.e. no data points, we get the prior back.

Posterior Distribution for a Simple Prior

If we look at our simplified case:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0 = 0, \mathbf{S}_0 = \alpha^{-1}\mathbf{I}) \quad (21)$$

the posterior parameters simplify to:

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{z} \quad (22)$$

$$\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} \quad (23)$$

A simple example

Bayesian regression for the target values

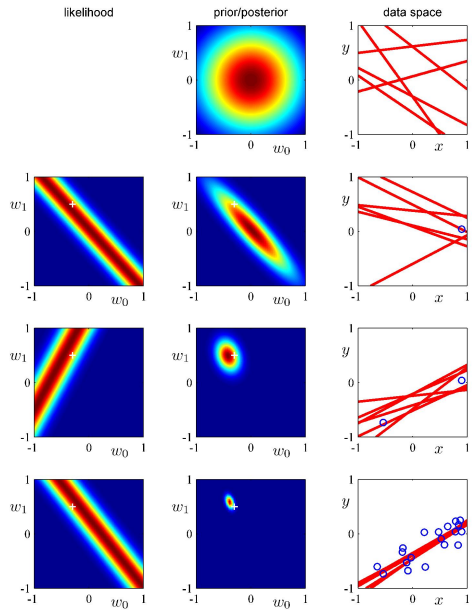
$$z_n = -0.3 + 0.5x_n + \epsilon$$

where ϵ is a Gaussian noise term ($\sigma = 0.2$).

To model this, we set $\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}$ and thus

$$y(x, \mathbf{w}) = w_0 + w_1x$$

Sequential Estimation: The demo (by Martin Felder) shows how the posterior's breadth gets smaller as more and more points z are taken into account, and how its mode converges to the optimum (=correct) values of the weights (white cross).



Predictive Distribution

Usually, we want to know output z for new values of \mathbf{x} – the model parameters \mathbf{w} are just a means to achieve this. To predict z , evaluate

$$p(z|\mathbf{x}, \mathbf{z}, \alpha, \beta) = \int \underbrace{p(z|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood (5)}} \underbrace{p(\mathbf{w}|\mathbf{z}, \alpha, \beta)}_{\text{posterior (18)}} d\mathbf{w} \quad (24)$$

(coin flip analogy: $p(x|\mathcal{D}, a, b) = \int_0^1 p(x|\theta)p(\theta|\mathcal{D}, a, b) d\theta$)

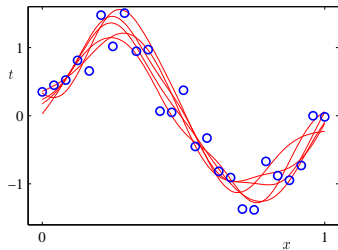
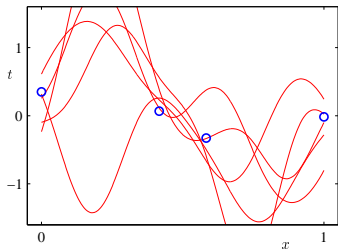
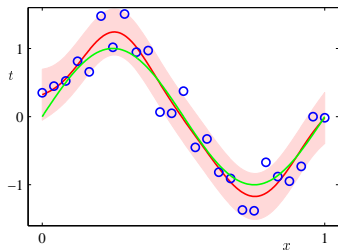
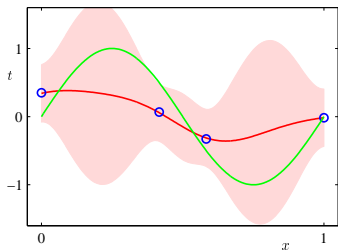
Predictive distribution

$$p(z|\mathbf{x}, \mathbf{z}, \alpha, \beta) = \mathcal{N}(z|\mathbf{m}_N^T\phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \quad (25)$$

with variance

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}). \quad (26)$$

Example (using 9 Gaussian basis functions)



Green: Underlying function, Blue: Observations, Dark-Red: Mode

Details for slide 12, example matrix calculus.

$$(\mathbf{z} - \Phi\mathbf{w})^T(\mathbf{z} - \Phi\mathbf{w}) = \mathbf{z}^T\mathbf{z} - 2\mathbf{w}^T\Phi^T\mathbf{z} + \mathbf{w}^T\Phi^T\Phi\mathbf{w}$$

Looking for the derivative with respect to \mathbf{w} , formulae (6e) and (5b) from the Roweis handout are applied:

$$2\Phi^T\Phi\mathbf{w} - 2\Phi^T\mathbf{z}$$

$\Phi^T\Phi$ is symmetric!