Noisy Data

Noisy data $\Rightarrow$ small margin.

Solution: ignore the noisy points.
Maximum Margin Classifiers

Recall that our maximum margin classifiers are models of the form

$$\hat{f}(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b),$$

where the normal vector $\mathbf{w}$ and the offset term $b$ of the decision surface are computed via the primal optimization problem,

$$\min \phi(\mathbf{w}, b) = \min \frac{1}{2} \mathbf{w} \cdot \mathbf{w},$$

subject to the constraints,

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1 \geq 0,$$

with $i = 1, \ldots, l$ given the training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \in \mathbb{R}^n \times \{+1, -1\}$. 
Softmargin Classifiers

If we allow points to lie on the “wrong” side of their supporting hyperplanes we need to keep track of the amount of error that this introduces ⇒ slack variables denoted with $\xi (x_i)$ (see Fig b above)

We change our training algorithm by taking the slack variables into account. We rewrite our constraints as

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) + \xi_i - 1 \geq 0,$$

with $\xi_i \geq 0$.

We also modify our objective function,

$$\min_{\mathbf{w}, \xi, b} \phi(\mathbf{w}, \xi, b) = \min_{\mathbf{w}, \xi, b} \left( \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{l} \xi_i \right),$$

Our new objective function looks just like the objective function for maximum margin classifiers except for the penalty term $C \sum_{i=1}^{l} \xi_i$. C is called the cost. In this way the optimization becomes a trade off between the size of the margin and the size of the error measured by the slack variables,

large $C \sim$ small margin
small $C \sim$ large margin
Softmargin Classifiers

Putting this all together,

**Proposition:** [Soft-Margin Optimization] Given a training set

\[ D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\} \subseteq \mathbb{R}^n \times \{+1, -1\}, \]

we can compute a soft-margin decision surface \( w^* \cdot x = b^* \) with an optimization,

\[
\min_{w, \xi, b} \phi(w, \xi, b) = \min_{w, \xi, b} \left( \frac{1}{2} w \cdot w + C \sum_{i=1}^{l} \xi_i \right),
\]

subject to the constraints,

\[
y_i (w \cdot x_i - b) + \xi_i - 1 \geq 0, \\
\xi_i \geq 0,
\]

with \( i = 1, \ldots, l, \xi = (\xi_1, \ldots, \xi_l) \), and \( C > 0 \).

**Note:** The slack variables have no impact on the form of our model \( \hat{f}(x) = \text{sign}(w^* \cdot x - b^*). \)
The Dual

As before we start by constructing the Lagrangian,

\[ L(\alpha, \beta, w, \xi, b) = \frac{1}{2} w \cdot w + C \sum_{i=1}^{l} \xi_i \]

\[ - \sum_{i=1}^{l} \alpha_i (y_i (w \cdot x_i - b) + \xi_i - 1) \]

\[ - \sum_{i=1}^{l} \beta_i \xi_i \]

We have an additional set of Lagrangian multipliers for the additional constraints.

This gives us the Lagrangian optimization problem,

\[ \max_{\alpha, \beta} \min_{w, \xi, b} L(\alpha, \beta, w, \xi, b), \]

subject to the constraints,

\[ \alpha_i \geq 0, \]

\[ \beta_i \geq 0, \]

for \( i = 1, \ldots, l.\)
The Dual

Since the primal objective function is convex, this Lagrangian has a unique saddle point and therefore a solution \( \alpha^*, \beta^*, w^*, \xi^*, b^* \) has to satisfy the KKT conditions,

\[
\frac{\partial L}{\partial w}(\alpha, \beta, w^*, \xi, b) = 0,
\]

\[
\frac{\partial L}{\partial \xi_i}(\alpha, \beta, w, \xi^*_i, b) = 0,
\]

\[
\frac{\partial L}{\partial b}(\alpha, \beta, w, \xi, b^*) = 0,
\]

\[
\alpha^*_i (y_i (w^* \cdot \bar{x}_i - b^*) + \xi^*_i - 1) = 0,
\]

\[
\beta^*_i \xi^*_i = 0,
\]

\[
y_i (w^* \cdot \bar{x}_i - b^*) + \xi^*_i - 1 \geq 0,
\]

\[
\alpha^*_i \geq 0,
\]

\[
\beta^*_i \geq 0,
\]

\[
\xi^*_i \geq 0,
\]

for \( i = 1, \ldots, l \).
The Dual

Now taking the partial derivatives in terms of the primal variables:

\[
\frac{\partial L}{\partial w}(\alpha, \beta, w^*, \xi, b) = w^* - \sum_{i=1}^{l} \alpha_i y_i \overline{x}_i = 0,
\]

\[
\frac{\partial L}{\partial b}(\alpha, \beta, w, \xi, b^*) = \sum_{i=1}^{l} \alpha_i y_i = 0,
\]

\[
\frac{\partial L}{\partial \xi_i}(\alpha, \beta, w, \xi_i^*, b) = C - \alpha_i - \beta_i = 0,
\]

Since both \( \alpha_i \geq 0 \) and \( \beta_i \geq 0 \) the last equation implies that

\[
C \geq \alpha_i \geq 0.
\]

Putting this all together we can derive the dual,

\[
\phi'(\overline{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \overline{x}_i \cdot \overline{x}_j.
\]
The Dual

**Proposition** [The Soft-Margin Lagrangian Dual] Given a soft-margin optimization in primal form (see the beginning of this set of slides) then the Lagrangian dual optimization for a soft-margin classifier is

$$
\max_{\bar{\alpha}} \phi'(\bar{\alpha}) = \max_{\bar{\alpha}} \left( \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j \right)
$$

subject to the constraints,

$$
\sum_{i=1}^{l} \alpha_i y_i = 0,
C \geq \alpha_i \geq 0,
$$

with $i = 1, \ldots, l$. Here, $C$ is the cost constant.

It is remarkable that this dual differs from the hard-margin case only in the range of values the Lagrangian multipliers can take on: Points in the margin $\alpha_i = C$, points on the supporting hyperplanes $C > \alpha_i > 0$, and points far away from the decision surface $\alpha_i = 0$. 
Soft-Margin Classifiers

\[ > \text{svm.model} \leftarrow \text{svm}(\text{Diagnosis} \sim ., \right. \\
\left. \quad \text{data=biomed.df,} \right. \\
\left. \quad \text{type="C-classification",} \right. \\
\left. \quad \text{cost=1.0,} \right. \\
\left. \quad \text{kernel="linear"}) \]
> svm.model <- svm(y~.,
    data=non.linear.df,
    type="C-classification",
    cost=1,
    kernel="polynomial",
    degree=2,
    coef0=0)
**Kernel-Perceptron**

```
let $D = \{(x_1, y_1), \ldots, (x_l, y_l)\}$
let $0 < \eta < 1$
$\alpha \leftarrow 0$
$b \leftarrow 0$
\(r \leftarrow \max\{|x| \mid (x, y) \in D\}$
repeat
  for $i = 1$ to $l$
    if sign($\sum_{j=1}^{l} \alpha_j y_j x_j \cdot x_i - b$) \(\neq y_i\) then
      $\alpha_i \leftarrow \alpha_i + 1$
      $b \leftarrow b - \eta y_i r^2$
    end if
  end for
until done
return $(\alpha, b)$
```

let $D = \{(x_1, y_1), \ldots, (x_l, y_l)\}$
let $\eta > 0$
$\alpha \leftarrow 0$
$b \leftarrow 0$
repeat
  for $i = 1$ to $l$
    if sign($\sum_{j=1}^{l} \alpha_j y_j k(x_j, x_i) - b$) \(\neq y_i\) then
      $\alpha_i \leftarrow \alpha_i + 1$
      $b \leftarrow b - \eta y_i$
    end if
  end for
until done
return $(\alpha, b)$

**Observations:**

- We extend our linear classifier to a non-linear perceptron.
- However, sub-optimal decision surface, algorithm stops as soon as a decision surface is found.