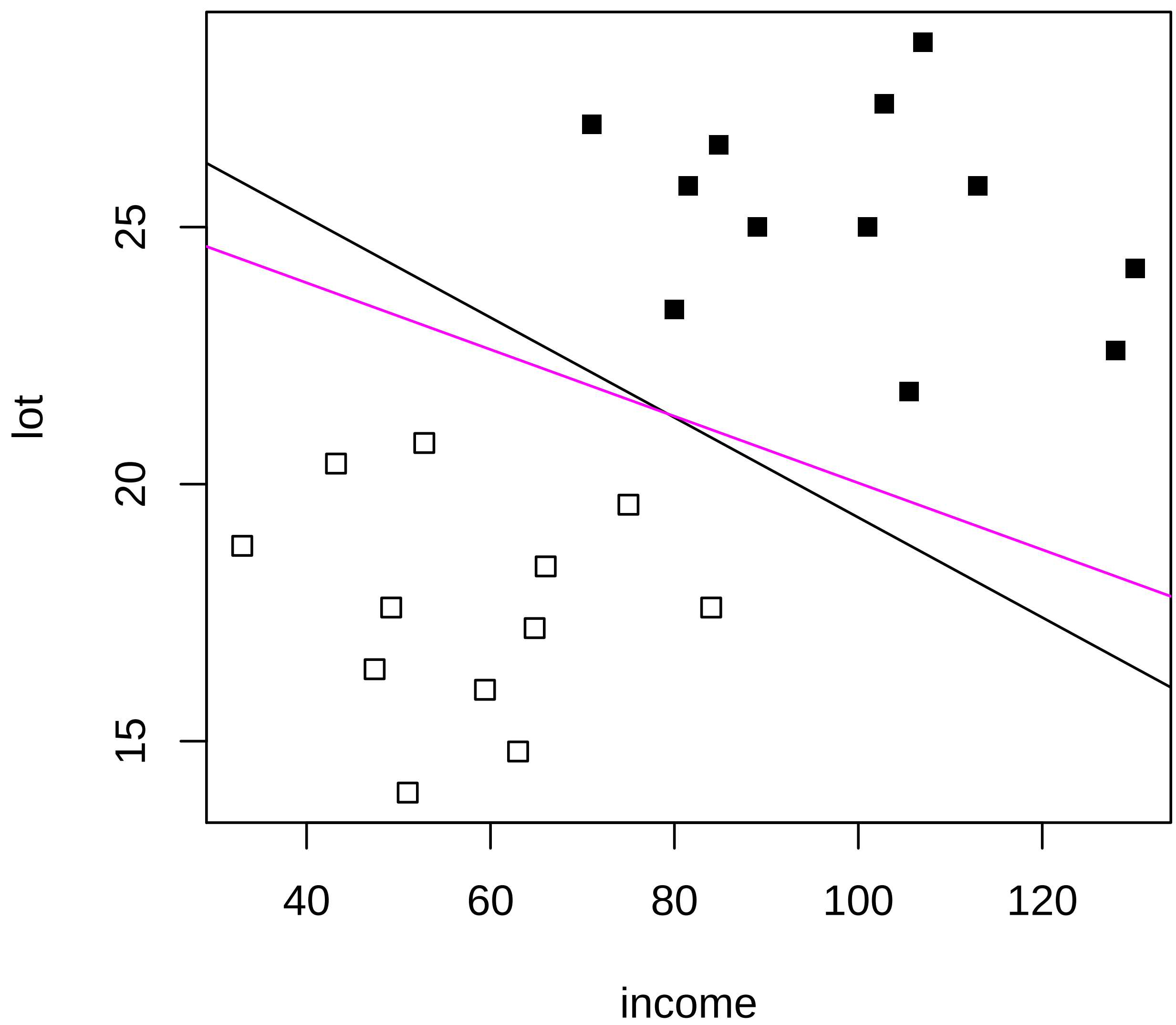
STAT 441: Lecture 20
Classification via regression II
Support vector machines

"Cutting edge stuff..."

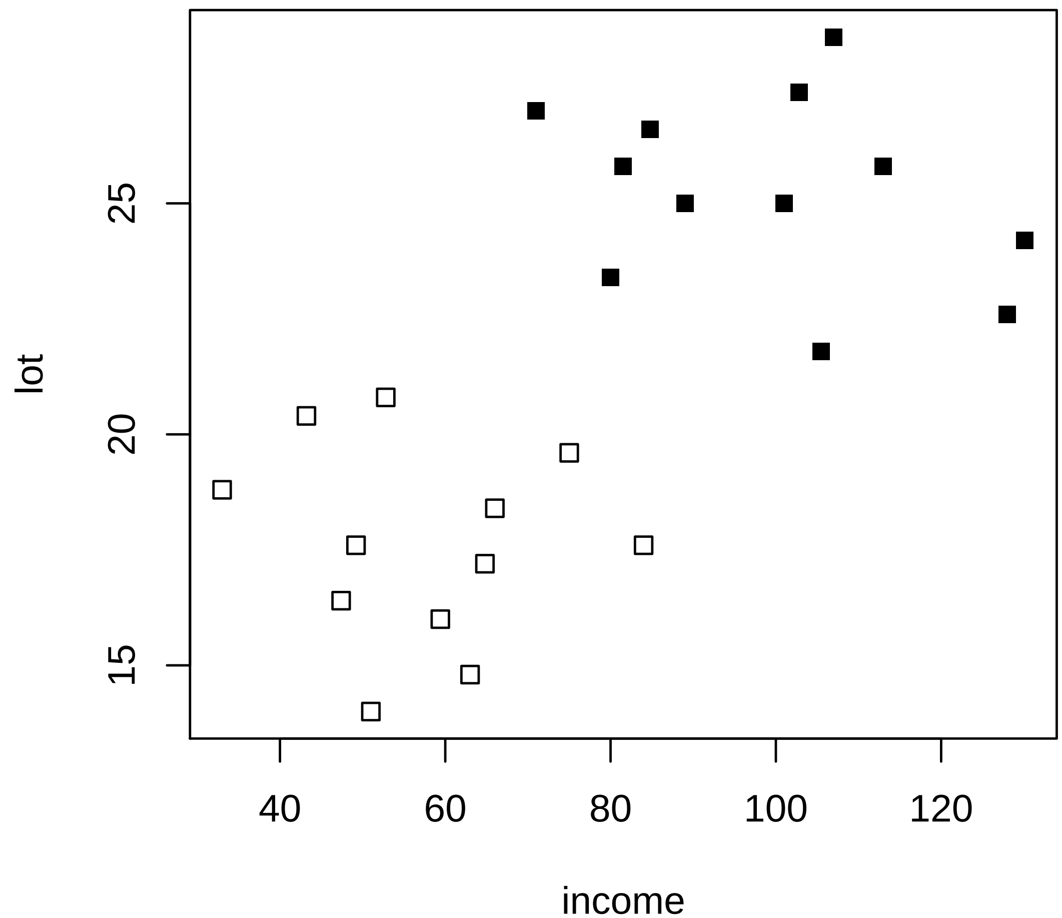
Logistic regression and LDA as regression again



What happened

```
> plot(moow[,1:2],pch=15*moow[,3])
> lm(2*riding-1~income+lot,data=moow)
(Intercept)       income          lot
   -4.32669      0.01447      0.14881
> abline(4.32669/0.14881,-0.01447/0.14881)
> glm(riding~income+lot,family=binomial,data=moow)
...
Warning messages:
1: algorithm did not converge in: glm.fit(x = X, y = Y, weights = weights, start
2: fitted probabilities numerically 0 or 1 occurred in: glm.fit(x = X, y = Y, we
```

Separated data



Did you notice that these are not original mowers data?
The classes are *separated*.

Then maximize margin!

New approach: *maximize margin*. Let $y_i = \pm 1$.

Find maximal B such that $y_i(x_i^T \beta + \gamma) \geqslant B$ and $\|\beta\| = 1$

Substitute $\tilde{\beta} = \dfrac{\beta}{B}$, $\tilde{\gamma} = \dfrac{\gamma}{B}$, to see equivalent:

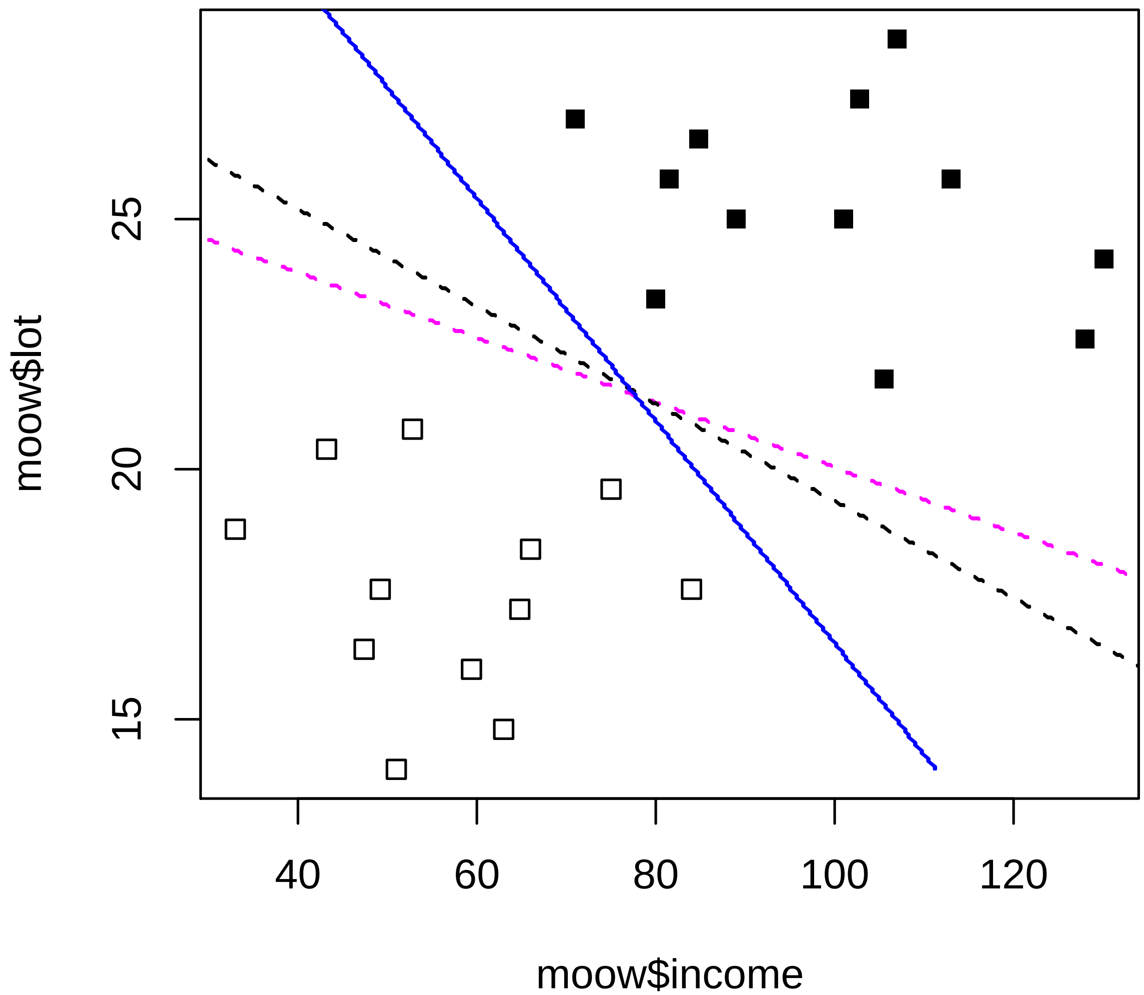
Find minimal $\|\tilde{\beta}\|$ such that $y_i(x_i^T \tilde{\beta} + \tilde{\gamma}) \geqslant 1$.

Solved via so-called *quadratic programming*.

```
> plot(moow[,1:2],pch=15*moow[,3])
> abline(15.7692/0.4072,-0.0905/0.4072)
> eqscplot(moow[,1],moow[,2],pch=15*moow[,3])
> abline(15.74692/0.4072,-0.0905/0.4072)
```

Doesn't look like it...

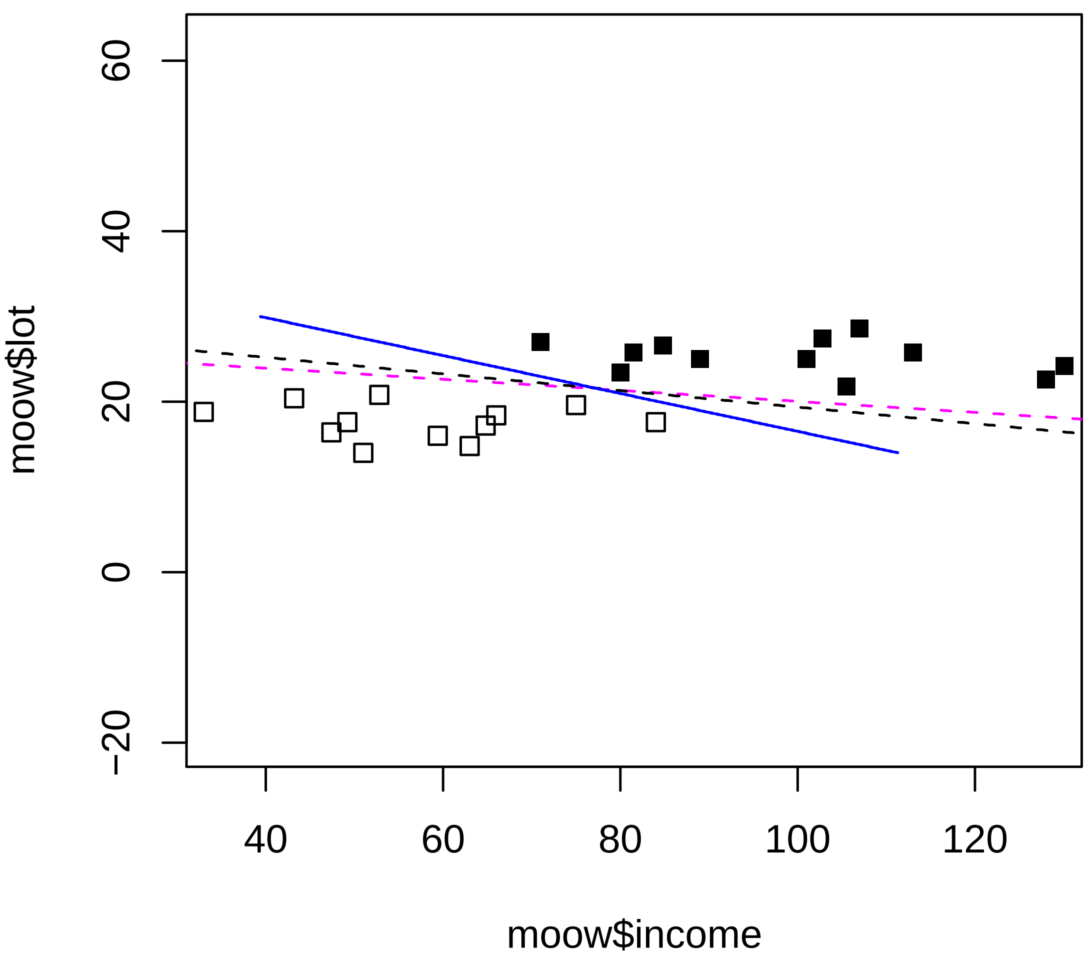
(broken black: LDA; broken magenta: logistic regression)



...because it's not scaled!

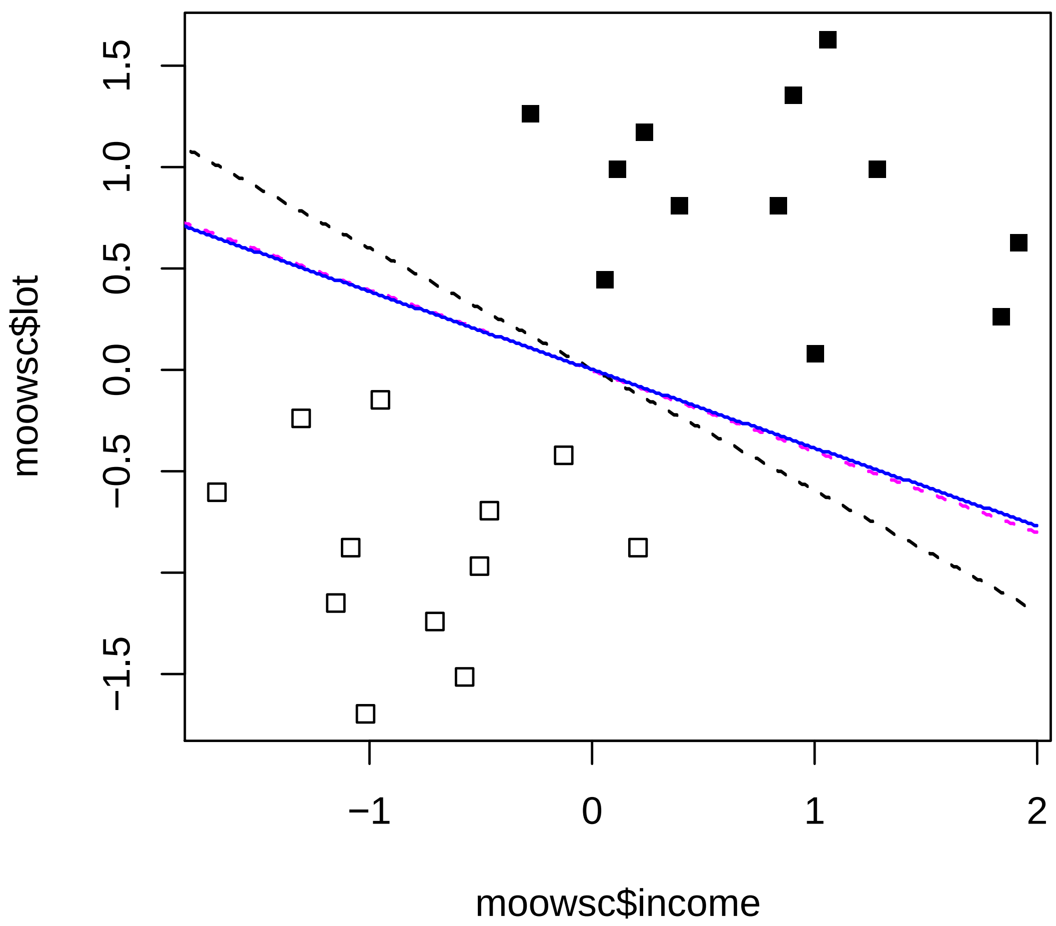
Using `eqscplot` from `library(MASS)`,

```
> eqscplot(moow$income,moow$lot,pch=15*moow[,3])
```



Three linear classifiers for scaled data

Maximum margin overplots logistic regression here - but this is rather a coincidence. There are similarities between the two, but in general they are different.



Yet another method that may need scaling...

Linear discriminant analysis doesn't
- only original variables involved

Logistic regression (dashed) doesn't
- if only original variables are involved

"Maximum margin method" may need (and usually does)

And - what if the classes overlap?

Modification

Find maximal B such that $y_i(\mathbf{x}_i^T \beta + \gamma) \geqslant B(1 - e_i)$ and $\|\beta\| = 1$, $e_i \geqslant 0$, $\sum_i e_i \leqslant C$.

Again, substitute $\tilde{\beta} = \frac{\beta}{B}$, $\tilde{\gamma} = \frac{\gamma}{B}$, to find the equivalent:

Find minimal $\|\tilde{\beta}\|$ such that $y_i(\mathbf{x}_i^T \tilde{\beta} + \tilde{\gamma}) \geqslant 1 - e_i$, and $e_i \geqslant 0$, $\sum_i e_i \leqslant C$.

Lagrange multiplier formulation: minimize

$$\frac{1}{2}\|\beta\|^2 + \lambda \sum_{i=1}^{n} e_i$$

subject to $e_i \geqslant 0$, $y_i(\mathbf{x}_i^T \beta + \gamma) \geqslant 1 - e_i$.

Equivalent formulation, with $f(\mathbf{x}) = \mathbf{x}_i^T \tilde{\beta} + \tilde{\gamma}$; minimize

$$\sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \frac{1}{2\lambda}\|\beta\|^2 \quad \text{where } u_+ = \max\{u, 0\}$$

Each C corresponds to some λ - additional requirement now: need to select tuning parameter C.

Nevertheless, can be overcome...

...and nonlinear version (adding transformed features) is called Support Vector Machine

How nonlinear

As with logistic regression, we can use not only x_i's, but also their functions, like x_i^2, $\log x_i$, ...

General form: $f(\mathbf{x}) = h(\mathbf{x})^T \beta + \gamma$

When working out the solution, people found out that it is not necessary to know h - but some related quantities for any pair of $\mathbf{x}$, $\mathbf{x}'$, which can be retrieved from a function called kernel: $K(\mathbf{x}, \mathbf{x}')$. (Here "kernel" means in general something else than "kernel" in kernel density estimation.)

Different kernels give different h (quite rich often, but seems not a problem).

Kernels

Linear kernel (`vanilladot`): $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$

Polynomial kernel (`polydot`): $K(\mathbf{x}, \mathbf{x}') = (\tau + \sigma \mathbf{x}^\top \mathbf{x}')^d$

Gaussian radial basis (`rbfdot`): $K(\mathbf{x}, \mathbf{x}') = e^{-\sigma \|\mathbf{x} - \mathbf{x}'\|^2}$

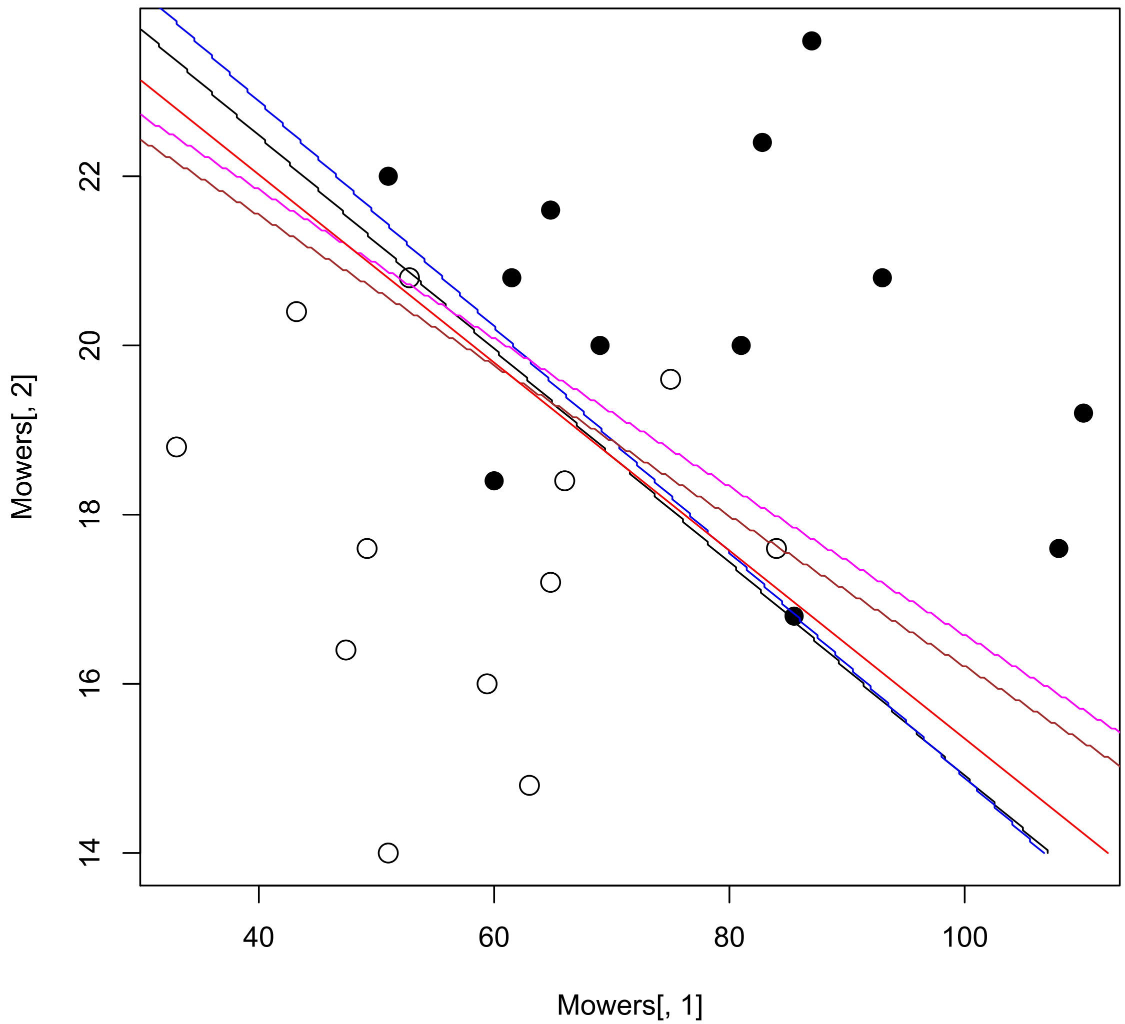
```
> library(kernlab)
> ksvm(factor(riding)~income+lot,data=Mowers,scaled=F,
+ kernel='vanilla',C=0.01)
> ksvm(riding~income+lot,type="C-svc",data=Mowers,scaled=F,
+ kernel='vanilla',C=0.1)
```

Uses λ/n instead of λ; also many, many other possibilities, also other methods (regression - default if response is numeric, and `type='C-svc'` is not specified)

Mowers again: linear kernel, various C

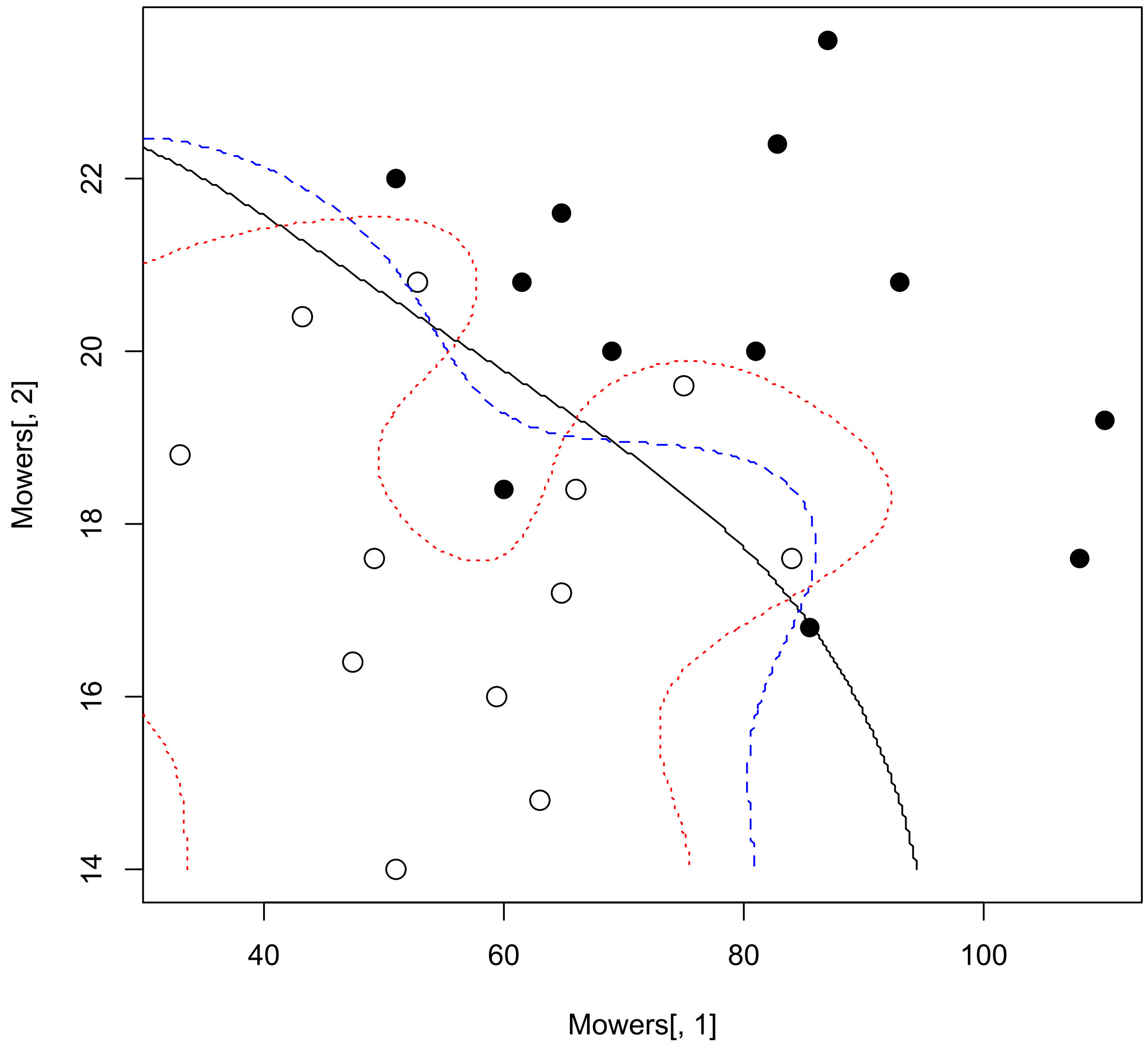
$C = 0.01, 0.1, 1, 10, 100.$

(See also the movie.)



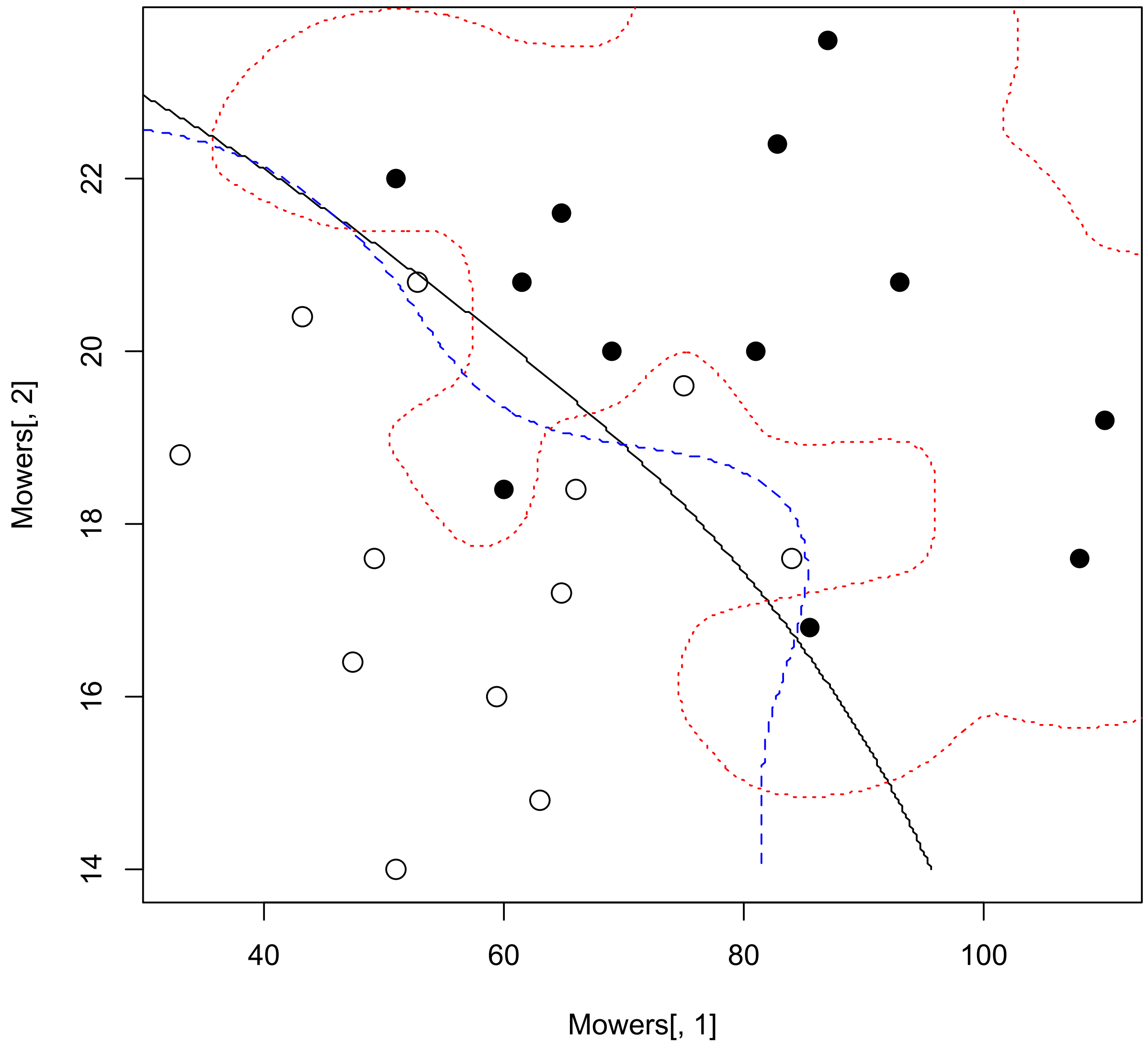
Mowers: Gaussian kernel, various C

$C = 0.001, 1, 100.$



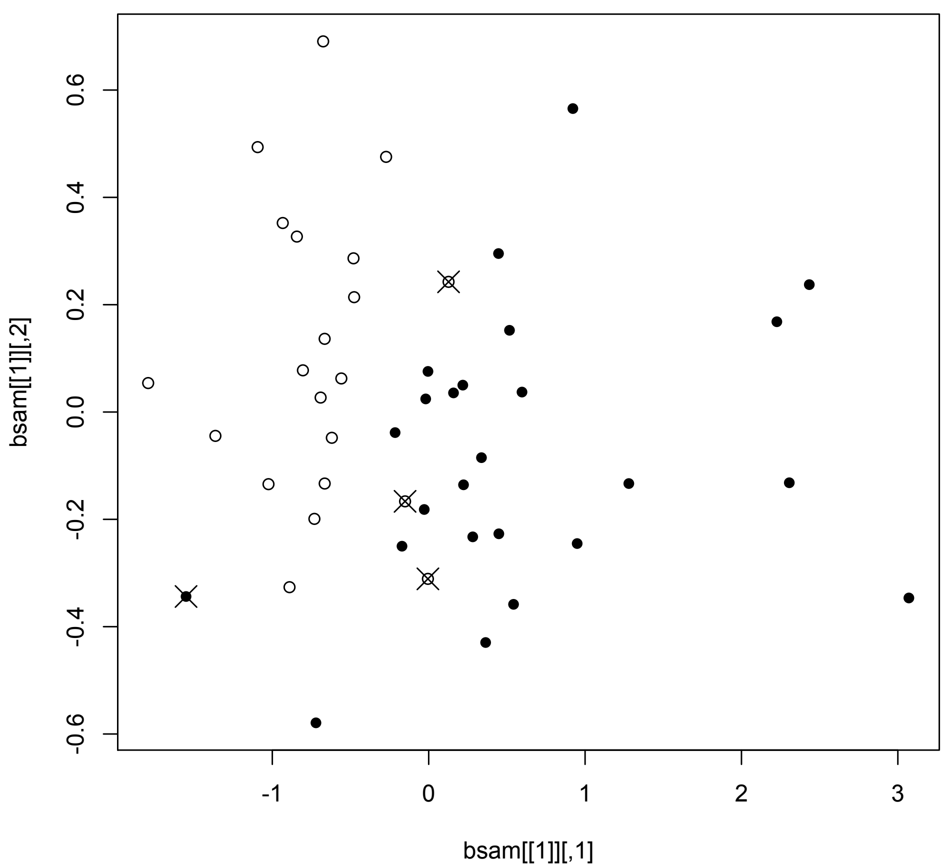
Mowers: Gaussian kernel, $C = 1$, various σ

$\sigma = 0.1, 1, 10$. A lot of room to play...



Bank data, default (Gaussian kernel, $C = 1$)

```
> bsam=sammon(dist(Bank[,1:4]))
> plot(bsam[[1]],pch=15*Bank$k+1)
> bansvm=ksvm(factor(k)~.,data=Bank)
Using automatic sigma estimation (sigest) for RBF or laplace kernel
> wrong=(Bank$k!=(predict(bansvm)))
> points(bsam[[1]][wrong,],pch=4,cex=2)
```



Summary

Very flexible and promising, but still in development.

A lot of tuning parameters, although this is being addressed.

Some similarity to logistic regression: the objective function

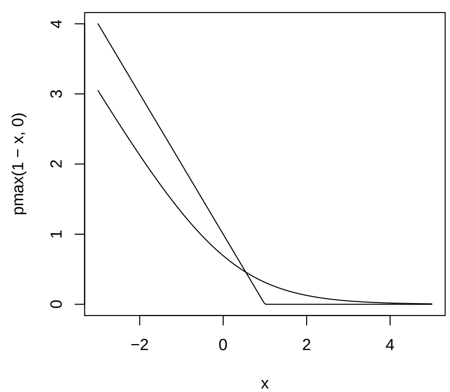
$$\sum_{i=1}^{n}(1-y_i f(x_i))_+ + \frac{1}{2\lambda}\|\beta\|^2$$

is in general

$$\sum_{i=1}^{n} L(y_i, f(x_i)) + \frac{1}{2\lambda}\|\beta\|^2$$

$L(y, f(x)) = (1 - y f(x))_+$ - support vector machine

$L(y, f(x)) = \log(1 + e^{-y f(x)})$ - logistic regression



Rather for two classes; for more than two somewhat unsettled.